Exploring Candidate Retrieval and Ranking for Entity Linking

エンティティ・リンキングのための候補検索とランキング方法に関する研究

TOHOKU
UNIVERSITY

Shuangshuang Zhou

Graduate School of Information Sciences

Tohoku University

A thesis submitted for the degree of *Doctor of Information Sciences*

January 2017

Acknowledgments

Without the support of many people, I could not finish this thesis. I would like to thank for their advice and support.

First of all, I would like to express my utmost gratitude to my advisor Prof. Kentaro Inui for his support and guidance during the past four years. He guided my research, provided the excellent facilities, and supported me in various aspects of living and studying in Japan. Thanks to him for giving me an opportunity to study abroad and always encouraging me with warm words.

I also would like to appreciate associate Prof. Naoaki Okazai who provided me insightful advice and comments on my research, especially, when my search fell into bottleneck. He brought neural network into my research task, which added novelty and improvement to my work. He also provided me patient and careful supervision on writing papers.

I am indebted to my seniors in my lab to support me. Firstly, I express my appreciation to my seniors, assistant Prof. Naoya Inoue, Dr. Canasai Kruengkrai. I am a newcomer to natural language processing field when I came to this lab. They advised me in various aspects, such as, learning basic knowledge of natural language processing, the way of using servers, programming skills, etc. Moreover, I am also grateful to research assistant Prof. Ran Tian, researcher Koji Matsuda for insightful comments on my research.

I would like to thank technical assistants in our lab, Yamaki-san, Mayumi-san, Kanno-san. As a foreigner student, they assisted me not only on issues in school but also my life in Japan. Thanks to them for often taking to delicious restaurants, which helped me to relax.

Finally, I would like to express my deep appreciation to my parents for their love and encouragement. Special thanks to my fiance, Dr. Yitao Ma, without his support and assistance, I would not get through all the problems and difficulties.

Abstract

Named entities that refer to real object in the world are essential components in natural language processing (NLP), such as *information retrieval*, *information extraction*, *question answering*, *knowledge base population*, etc. Since resolving named entity mentions has two main problems:'*variety*' and '*ambiguity*', it is very hard for computers to understand the meaning of texts while human can understand it according to the context. The problem of variety means that a named entity has multiple name variations (alias) while different entities could share the same surface. The problem of ambiguity means that many different named entities share the same name surface. The many-to-many mappings between mentions and name entities widely exist in text.

Entity Linking (EL), also known as *Named Entity Disambiguation* (NED), is the task of identifying and linking mentions in different documents to their corresponding entries in a large-scale knowledge base, which can solve the problem of variety and ambiguity. EL is beneficial for many NLP tasks including information retrieval, question answering, searching digital libraries, coreference resolution, named entity recognition, etc. Furthermore, grounding written language with respect to background about entities and events is important for constructing general or domain-specific knowledge base and ontologies. Hence, EL is useful for knowledge base population as well. Generally, there are two important components of EL systems:

candidate retrieval and candidate ranking.

In candidate retrieval phase, the aim is to find a candidate list for each mention. The retrieved candidate lists have two properties, high-recall and small candidate numbers. However, previous work did not deeply explore the research of candidate retrieval. In this thesis, we applied comparable study on the performance of candidate retrieval by several conventional methods including search-based method and alias-based method. For alias-based method, we integrated various alias resources and constructed a alias dictionary. We find that it is more effective than other methods based on string/character similarity (search-based method).

Moreover, we added two modules for improving recall and decreasing candidate entities: mention extension and pruning. The experimental results verify that mention extension module can improve recall while pruning module can eliminate noise candidates. Furthermore, we thoroughly analyzed mentions which can not be reached to correct entities as well, which could benefit improving the performance of candidate retrieval in the future.

In candidate ranking phase, the motivation is to rank candidate entities according to the information in the context of mentions and the information of candidate entities. State-of-the-art systems are based on a global resolution method and mostly adopt link-based features that leverage relationships of co-occurring entities in the knowledge. We find that linguistic features can also significantly solving the problem of ambiguity.

Thus, in this thesis, we explored effective linguistic features extracted from the context, which could be the fundamental part of the combination of global resolution method and effective features. Moreover, we studied and compared the effects of linguistic features in a comprehensive way.

Furthermore, since the effectiveness of embedding models of representing words, entities and paragraphs has been verified in recent search, we take this advantage by exploring several embedding models that encode context information of Wikipedia entities. The experimental results demonstrate that those embedding models improved the performance of candidate ranking. We also reported the performance of feature based on each embedding model in detail.

After studying on candidate retrieval and candidate ranking of EL, we evaluated the EL

system by adding the NIL determination component. The accuracy of our system on Japanese Wikification corpus significantly outperforms the previous work. Finally, we analyzed unsolved mentions of our system and discussed the possible improvement solution in the future work.

List of Publications

Journal Papers (Refereed)

1. Shuangshuang Zhou, Canasai Kruengkrai, Naoaki Okazaki, Kentaro Inui. Exploring Linguistic Features for Named Entity Disambiguation. International Journal of Computational Linguistics and Applications, Vol.5 No.2, pp. 47-60, July-December 2014.

2. Shuangshuang Zhou, Naoaki Okazaki, Koji Matsuda, Ran Tian, Kentaro Inui. Supervised Approaches for Japanese Wikification. Journal of Information Processing. It will appear in March, 2017. .

International Conference

1. Shuangshuang Zhou, Koji Matsuda, Ran Tian, Naoaki Okazaki, Kentaro Inui. A Pipeline Japanese Entity Linking System with Embedding Features. In Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation (PACLIC 30), October 2016.

Other Publications and Presentations(Not refereed)

1. Shuangshuang Zhou, Naoaki Okazaki, Kentaro Inui. Exploring the Challenges of Entity Linking in Knowledge-based Question Answering. the 10th YANS-NLP Symposium, September 2015.

2. Shuangshuang Zhou, Canasai Kruengkrai, Kentaro Inui. Exploring Linguistic Features for Cross-document Named Entity Disambiguation. In Proceedings of the 21th Annual Meeting of the Association for Natural Language Processing, pp.373-376, March 2015.

3. Shuangshuang Zhou, Canasai Kruengkrai, Naoaki Okazaki, Kentaro Inui. Using a Broad Range of Linguistic Features in Entity Discovery and Linking. In Proceedings of Text Analysis Conference 2014 (TAC 2014), November 2014.

Contents

CHAPTER 1

Introduction

1.1 Background

In natural language processing (NLP), *named entities* are important components. Grishman and Sundheim [17] coined the term "Named Entity" for the Sixth Message Understanding Conference (MUC-6) that is related with Information Extraction (IE) tasks. A named entity is a unique and real object in the world, such as "Apple Inc.", "Steve Jobs" or any objects that can be named. Named entities are a kind of essential information unit in a number of NLP tasks, such as *Information Retrieval*, *Information Extraction*, *Question Answering*, *Knowledge Base Population*, etc. Figure 1.1 illustrates the role of named entities in those above NLP tasks.

Named Entity Recognition and Classification (NERC) is defined to recognize named entities including person, organization, location names and numeric expressions including time, date, money and percent expressions in text[17]. NERC is typically broken down into two main phases: mention detection and classification (also called entity typing). However, the type information of entities from the output of NERC is insufficient for complex Natural Language Processing tasks, such as automatic knowledge base population, and question answering, etc.

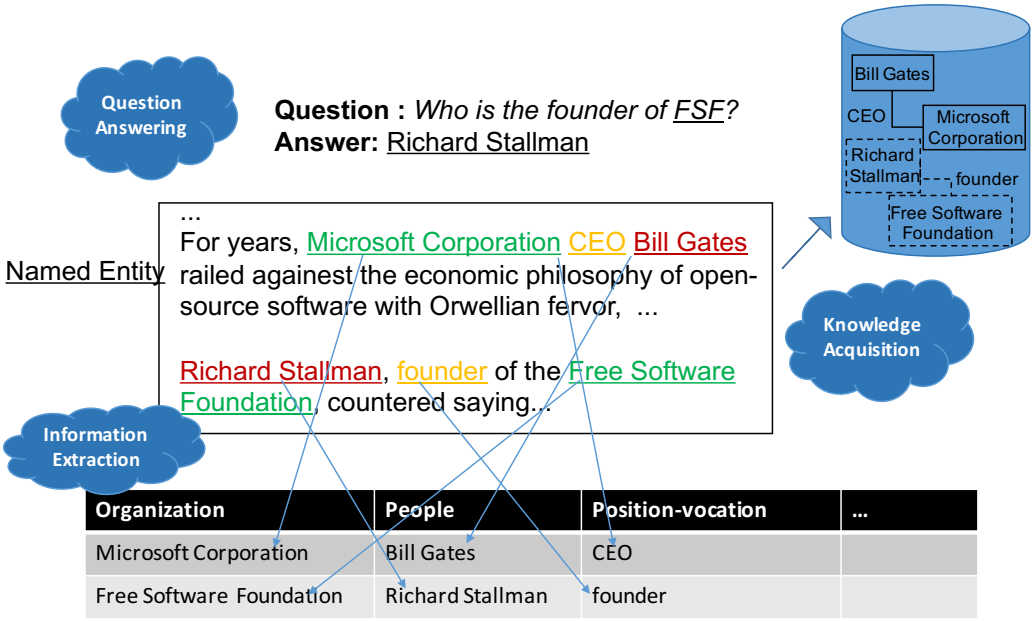

Figure 1.1: Example of the role of named entities in several NLP tasks.

To facilitate those complex NLP applications, the definitions and other detailed information about name entities are necessary. We can acquire a full range of information of named entities by retrieving knowledge base(KB), in which named entities are usually compiled and stored as entries. Therefore, we need a technology that can automatically resolve named entity mentions to entries in a KB.

On the other hand, since the ambiguity of human language, it is very hard for computers to understand the meaning of texts while human can easily understand it according to the context. For example,

> The **IOC** is facing the elements of instability from the market of China from the beginning of this new century. The Olympics at major Asian nations can never ignore this kind of political aspects.

In this snippet of text, 'IOC' may refer to amount of named entities, such as "International Olympic Committee", "International Ornithological Congress", etc. For human readers, it requires little effort to recognize 'IOC' with "International Olympic Committee" due to A clue word, 'Olympics', appears in the text. But for computers, it is not easy. Therefore, it is neces-

sary to develop a technology that can automatically distinguish among different named entities with high ambiguity.

Entity Linking (EL), is the task of identifying whether a mention refers to a certain entity and linking mentions in documents to their corresponding entries in a large-scale knowledge base. When the referent KB is Wikipedia[1], the task is called as *Wikification*. The detail of EL is introduced in Chapter 2.

EL is useful in many NLP tasks including information retrieval [4], question answering [34], searching digital libraries [22], coreference resolution [11, 21], named entity recognition [11]. Furthermore, grounding written language with respect to background about real-life entities and events is significant for building general or domain-specific knowledge base and ontologies. Hence, entity linking is beneficial for knowledge base population [55, 10] as well.

1.2 Key Challenges

There are two main problems of *variety* (synonymy) and *ambiguity* in natural language involving named entities.

1.2.1 Variety

Due to various ways of writing, named entities have many different surfaces in texts. For example,

> **IBM** gave investors a sign that **Big Blue** may finally be turning things around. Now it has to prove it can continue to drive momentum.

In this news article[2], "'Big Blue" and 'IBM' both refer to the named entity "International Brotherhood of Magicians".

The problem of variety exists in Japanese as well, For example,

[1] https://en.wikipedia.org

[2] https://www.bloomberg.com/news/articles/2016-07-19/ibm-investors-cheer-first-signs-of-success-for-big-blue-strategy

1. 長かったアメリカ大統領選挙が、まさかの結果とともに終わりました[3]。

   The long **USA** presidential election ended with a fruitless result.

2. 日産自動車広報によると、日産において日、米、欧で販売される車両[4]...

   According to the public information of Nissan Motor Co., Ltd. , Nissan vehicle to be sold in Japan, **USA** and Europe in Nissan...

The named entity “アメリカ合衆国(United States of America)” has various expressions, such as “アメリカ”, ‘米(USA)’, etc. A named entity may have various type of alias surfaces, such as acronym, spelling variations, abbreviation, metaphorical names, nick names, etc. It is difficult to match those various alias with their corresponding entity in the KB.

Entity linking focuses on matching a named mention to its possible corresponding named entity by both conventional and advanced approaches. Conventional approaches are usually based on string or character comparisons but some named mentions can not reach to the correct entires. Therefore, the advanced approach is to utilize a huge number of name alias that are extracted from resources including Wikipedia, Geonames, etc.

1.2.2 Ambiguity

The problem of ambiguity is the other significant problem of named entities mentions. As it is introduced in Section 1.1, ‘ambiguity’ makes computers confused when named mentions share the same surface. Here is an example,

1. 稲作（いなさく）とは、イネ（稲）を栽培することである。主に米を得る[5]...

   The rice (rice), is to cultivate rice (rice). In order to obtain mainly **Rice**...

2. 日産自動車広報によると、日産において日、米、欧で販売される車両[6]...

[3] https://socialnews.rakuten.co.jp/link/1065620
[4] https://ja.wikipedia.org/wiki/Œ
[5] https://ja.wikipedia.org/wiki/œ
[6] https://ja.wikipedia.org/wiki/Œ

According to the public information of Nissan Motor Co., Ltd. , Nissan vehicle to be sold in Japan, **USA** and Europe in Nissan...

The named entity mention '米' refers to the named entity "米(こめ) Rice" in the first instance while it refers to "アメリカ合衆国(United States of America)" in the second instance. Without additional disambiguation processing, computers may regard this two '米' mentions as the same named entity.

When we draw attention to the problem of variety and ambiguity, it is important to note a further problem caused by those two problems simultaneously occurring as well. We notice that the relation between mentions and entities is not many-to-one but many-to-many mapping. Figure1.2 shows the many-to-many correspondence between mentions and named entities.

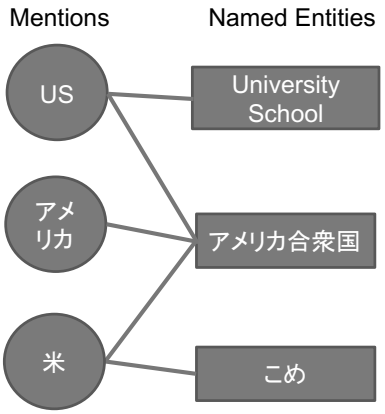

Figure 1.2: Many-to-many mapping between mentions and named entities.

1.3 Research Issues

In this thesis, we tackle the problem of variety and ambiguity of named entity mentions in entity linking. Thus, we address the following two main issues to solve those problems:

- **Candidate Retrieval** In candidate retrieval process, candidate entities are searched for the given named mention in text. Candidate retrieval search as many as possible candidate entities for named mentions.

- **Candidate Ranking** In candidate ranking phase, it is a fundamental way to disambiguating candidate entities by leveraging the context in the text. The correct entity is identified based on clues in the context of the given named mention.

1.4 Contributions of this Thesis

This thesis makes following contributions.

- **Exploring the research of candidate retrieval** In candidate retrieval phase, we aim to achieve a high-recall list of candidate entities without increasing the number of unrelated entities. We explore conventional methods for candidate retrieval. Especially, inspired by crosswikis[54], we investigate a method of building an alias dictionary combining with various resources (e.g. Wikipedia anchor texts, disambiguation pages, redirect pages, bold texts, etc.). Furthermore, we present two significant sub components for finding the gold named entity: *mention extension* that may increase the recall and *pruning* that may eliminate noisy candidates. We verify that the two sub components are effective for candidate retrieval.

  Moreover, previous studies are lack of comparable study on the performance of candidate retrieval and thorough analysis to mentions unreached to correct entities. In order to improve the accuracy of entity linking, we compare several conventional candidate retrieval methods in this thesis. In addition, we analyze mentions that can not reach to the correct entities by alias-based methods.

- **Studying linguistic features for candidate ranking** State-of-the-art systems [28, 49] are based on a global resolution method and mostly adopt link-based features that leverage relationships of co-occurring entities in the knowledge. However, when there is seldom co-occurring entities in context, linguistic information could affect disambiguation significantly [5, 10].

  Linguistic features can locally measure the coherence between mentions and entities in

context. Therefore, we study the effects of multiple linguistic features in a comprehensive way in this paper. Especially, we compare the effectiveness of each linguistic feature, e.g., document similarity, document topics, and POS features. Furthermore, we find those features effective in Japanese EL as well.

- **Exploring the effectiveness of embedding features in candidate ranking** Moreover, in order to better represent the context and compute the similarity between the context of mentions and Wikipedia articles, the previous work on English EL explores embedding models in candidate ranking [56, 4]. Embedding models provide useful representations of linguistic units such as words [42], entities [56], paragraphs [36], *etc*. This dense and low-dimensional representation is effective to compute the semantic similarities.

  Therefore, we learn several embedding models to encode context information of entities in Wikipedia articles, including word embedding, entity embedding, paragraph embedding. Distinguished with the previous work, our new representation of entities leverages their context across the Wikipedia articles. We find that the embeddings are useful for disambiguating mentions in texts by evaluating them on both English and Japanese corpus.

- **Constructing a pipeline entity linking system on Japanese** The research on Japanese entity linking has received less attention. These previous studies are not comparable with the ones for English EL. First, the domain of the previous work on Japanese is limited. For example, Furukawa et al. [13] focused on linking mentions in academic fields, and linked technical terms to English Wikipedia. Some studies only focus on linking geopolitical entities in local news articles [48, 29]as well. Second, a few studies address Wikification in generic domains, but they use the English Wikipedia as a target from Japanese mentions [26, 46]. This setting has two problems: translating Japanese mentions into English and the insufficient coverage of English Wikipedia for Japanese entities.

  In this thesis, we build a pipeline EL system containing candidate retrieval, candidate

ranking and NIL determination modules. The system can significantly outperform previous work on Japanese EL. In addition, we comprehensively analyze errors of our system on Japanese Wikification corpus.

1.5 Thesis Organization

This thesis is organized as follow.

- **Chapter 2: Preliminaries and Literature Review of Entity Linking** In this chapter, we first overview the related studies of sense disambiguation. We further introduce the general framework of entity linking, variant entity linking tasks and referent KBs for entity linking. We make a comprehensive survey on approaches of candidate ranking. In particular, we review features of supervised EL in previous work. Finally, we introduce traditional English EL dataset and a novel Japanese Wikification corpus.

- **Chapter 3: Exploring Candidate Retrieval for Entity Linking** In this chapter, we study on conventional candidate retrieval approaches, such as string matching approach and approaches based on information retrieving. Moreover, we discuss how to extract name alias from various resources. We construct an alias dictionary contains numerous many-to-many mappings between aliases and name entities. Additionally, we evaluate the above conventional approaches. We confirm that applying fuzzy matching on alias dictionary can provide high-recall candidate sets containing seldom noisy candidates.

- **Chapter 4: Exploring Candidate Ranking for Entity Linking** In this chapter, we utilize a supervised model to rank candidate entities. We study and apply various prevalent linguistic features for English entity linking. Furthermore, we reuse several effective features for Japanese entity linking and confirm their strength via experimental evaluation. Ultimately, we evaluate the overall performance of pipeline system on both English and Japanese copora.

- **Chapter 5: Embedding Features for Candidate Ranking** In this chapter, we describe

three embedding models for constructing low-dimensional vectors for the context of mentions and description text of entities: word, entity and paragraph vectors. We first illustrate the preprocessing and procedures of learning embedding models on a large-scale unstructured corpus, Wikipedia. What is more, we verity the effectiveness of embedding features on Japanese Wikification corpus.

- **Chapter 6: Conclusions** In this chapter, we summarize our work, contributions. Some future perspectives are discussed as well.

CHAPTER 2

Preliminaries and Literature Review of Entity Linking

We introduce the preliminaries and notation for the main contributions of this thesis in this chapter. We first introduce some relevant tasks of entity linking, the definition and history of entity linking and Wikification in Section 2.1. We then introduce and formally define the key components of entity linking task in Section 2.2. Section 2.3 concludes a general overview of related work and summarize recent approaches. Finally, we introduce performance measures and data set for evaluating entity linking system in Section 2.4.

2.1 From Sense Disambiguation to Entity Linking

Sense ambiguity is one of the well known tasks in NLP. It is easy for humans to disambiguate words and names while it is difficult for computers. The definition of sense ambiguity problem is widely considered to solve the ambiguity of common nouns, adjectives, and verbs. The problem of named entity recognition and classification became an essential task of information extraction[17] in the 1990s. Recently, more and more attention was payed to the ambiguity problem of named entities. In this chapter we present an overview of word sense disambigua-

tion and entity linking tasks.

2.1.1 Word Sense Disambiguation

Word sense disambiguation (WSD) is viewed as an "intermediate task"[59], which is necessary to accomplish many natural language processing tasks. It is essential for lots of language understanding applications, such as machine translation, information retrieval, speech processing and text processing.

The WSD task attracted the attention of researchers for many years and is still an open problem. WSD addresses the process of identifying which sense (meaning) of a word is used in a sentence, when the word has multiple meanings. Entity linking (EL) has some similarities with WSD because they are both solving with meaning based on the context. However, there are still several differences, which provide the following new challenges;

- In EL, the challenge is to link the named entity textual mention to a list of entries in knowledge base instead of a word to a vocabulary list.
- In WSD, all synonyms of a word exist in the dictionary. However, there is just one entry for each named entity in knowledge bases.
- While WSD cores a word as a single token, a named entity may be referred to by a single token or phrases (e.g. "the Big Apple").
- WSD involves with the part-of-speech of a word, such as noun, verb, etc. However EL is related with the entity type, such as, a person, organization, or location name.

2.1.2 Cross Document Coreference Resolution

Cross document coreference resolution (CDCR) focuses on clustering coreferences in a collections of documents. In coreference resolution, mentions are locally solved by using their context in a single document while CDCR use the document of a mention as context for disambiguation.

Begga and Baldwin [3] proposed vector space model (VSM) where context of a mention is represented as term vectors from sentences where the mention exists. They adopted an incremental clustering approach where items are compared with existing clusters and are either added to a similar cluster or create their own.

Gooi and Allan [15] generated a Person-X corpus containing 25K person name mentions from the NYT. They utilized VSM to cluster entity mentions, not coreference chains, and "snippet", which is a 55-token window surrounding the mention that may cross sentence boundaries. They comparably experimented with different clustering approaches and found that hierarchical agglomerative clustering (HAC) performs better than incremental clustering.

Singh et al. [52] carried out a large-scale clustering method that represents CDCR as an undirected graphical model. New cluster assignments are proposed and the parallel tasks are structured using a hierarchy to optimize assignment efficiency during the algorithm. Experiments on the Person-X corpus [15] show that their method reaches the same accuracy as pairwise clustering in 10% of the runtime. On a large scale corpus containing hyper-links to Wikipedia (Singh et al. [53]), their method scores 73.7% F B3.

2.1.3 Entity Linking (EL)

Entity linking is similar to the widely-studied task of Word Sense Disambiguation. Both tasks address problems of variety and ambiguity in natural language while EL focuses on named entities. The tasks differ in terms of candidate retrieval and NIL determination.

The problem of EL is to identify and connect textual mentions to an entry in referent KB that contains information about this mention. Therefore, EL, also known as Named Entity Disambiguation (NED), is to identify and link named entities (e.g. Steve Job), events (e.g. The Olympic Games), concepts (e.g. apple), etc., to the knowledge base. If Wikipedia is used as the referent KB, the task is defined as *Wikification*[41].

In 2008, the National Institute of Standards and Technology (NIST) of US initiated the Text Analysis Conference (TAC) to support research within the Natural Language Processing community by providing large-scale evaluation of NLP methodologies across various tracks.

Among them, the Knowledge Base Population (KBP) track defined the entity linking (EL) task.

In the entity linking task, each query (mention) may refer to a named entity in the referent knowledge base or refer to an entity that does not exist in the reference KB [39]. They also build standard resources for the evaluation of entity linking techniques by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. All the linguistic resources, including data, annotations, system assessment, tools, and specifications, have been created by LDC.

The entity linking task can be divided into the two subtasks of disambiguating named entity mentions. and identifying which mentions have no link in the knowledge base (NIL classification and clustering). This thesis focus on the task of the first part.

2.2 General Framework of Entity Linking

Firstly, we introduced some terms relevant to the entity linking task. A *mention* is a span of text in a document, such as news articles, web forums, etc. Mentions play referring roles in the task and may refer to proper nouns (e.g. International Olympic Committee), common concepts (e.g. apple (fruit)), positional titles (e.g. President). In entity linking task, a mention will be solving into two classes: InKB (NonNIL) and NIL. InKB mean mentions have referent named entities in the KB while NIL mean the referent named entities of mentions do not exist in the KB.

Then, a generic framework for EL systems is discussed. Figure 2.1 shows a general procedure of entity linking. Entity linking contains several stages: detecting mentions of named entities (Mention Detecting), retrieving candidates for each mention (Candidate Retrieval or Candidate Generation), entity disambiguation (Candidate Ranking) and NIL Determination.

2.2.1 Mention Detecting

Before linking mentions to their referent named entities, it is necessary to find possible mentions of named entities in a text. It could be realized by using named entity recognition tools. Moreover, several less elaborate approaches collect a huge number of candidate mentions by

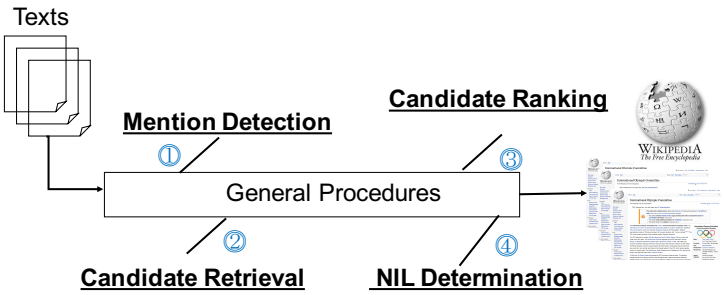

Figure 2.1: General procedure of Entity Linking.

involving simply spotting words starting with upper case letters or looking for character strings corresponding to all alias of all available entities included in the knowledge base.

2.2.2 Candidate Retrieval

Once possibles mentions in a text has been achieved, these mentions need to be linked to entities in the KB or NIL. However, comparing each mention with all of the entries in the KB would be costly and inefficient. Thus, we need to select candidate entities for each mention. Conventional approaches utilized non-contextual factors of mentions, e.g., the surface forms of mentions and possible variants (or alias) are considered as possible entities. Useful information regarding entities may be extracted from the knowledge base in order to generate a list of possible name variants for these entities, such as Wikipedia disambiguation pages, Wikipedia redirection pages, etc.

2.2.3 Candidate Ranking

The most important stage in entity linking is to disambiguate candidate named entities. In order to disambiguate candidate entities, several available clues could be used, such as the context of mention, descriptions of entries in KB, etc.).

In most cases, using clues as features leads us to apply a numerical approach. These approaches are based on the computation of a distance between a mention and each candidate en-

tity. The computation could by carried out utilizing heuristics or machine learning approaches. Each available feature is assigned a weight if using heuristics. Machine learning approaches could be applied on large scale copora. The computed distances are then used to sort candidate entities. After ranking, the top 1 candidate entity will be linked.

2.2.4 NIL Determination

There is no KB could contains overall named entities in the world while the range of named entities dramatically increasing daily. Those uncovered named entities (NILs) are addressed in knowledge base population (KBP) task. Therefore, it is necessary to recognize NILs (NIL Determination) in entity linking task, which is defined as a preprocessed task in TAC-KBP (Text Analysis Conference Knowledge Base Population) which aims to populate KB with additional information of entities in the KB and uncovered named entities [39].

NIL determination aims to judge whether the highest ranked candidate entity from candidate ranking step could be linked to the mention or there is no proper entry for the mention in the KB.

2.3 An Overview of Previous Work

To give a general overview, this section reviewed recent approaches in entity linking from different aspects. Since entity linking has attracted a lot of attention in recent years, there has been a huge number of publications. Here, we mainly concentrate on supervised approaches that link entity mentions in text to Wikipedia. Approaches could be summarized into two classes: Linguistic Features (Section 2.3.1) and Link Features (Section 2.3.2). Section 2.3.3 focuses on the development of entity linking in Japanese language. We summarized several advanced embedding approaches that applied in entity related tasks in Section 2.3.4.

2.3.1 Linguistic Features based Candidate Ranking

Linguistic features showed promising results in previous studies [5, 10, 60, 16], such as document similarity, word overlapping, entity-level word overlapping, document topics. However, only partial linguistic features are explored by previous work. Dredze et al. [10] captured features based on mentions, source documents and KB entries, but features about document topics are not involved. Zhang et al. [60] made big efforts on candidate selection and acronym expansion, but their disambiguation method only depended on document topics. Therefore, we summarize and refine effective linguistic features of previous work, and propose a broad range of linguistic features in this paper.

Linguistic features are used to measure the coherence between mentions and candidates, which are also called local methods by previous studies [49, 8, 18, 20, 23]. Combining local methods with global features or global ranking methods, the NED system performance is improved significantly [49]. Among of them, TF/IDF cosine similarity is mostly used by global inference systems for multiple purposes: ranking candidates [49, 8], filtering out noisy candidate [18], and assigning an initial confidence score for subsequent ranking phrase [20, 23]. However, TF/IDF cosine similarity is insufficient to capture the coherence between mentions and entities.

Moreover, entity popularity is a salient measure of mentions and entities [44, 12, 2, 1], and it could check how likely a surface refer to an entity. Entity popularity is a strong baseline for entity linking [14]. However, this feature could ignore unpopular correct entities.

2.3.2 Link Features based Candidate Ranking

On the other hand, link-based features strongly depend on the link structure of knowledge base (Wikipedia), e.g., link statistics (incoming links and outgoing links), and category information, etc. Link-based features are mostly used by global inference systems for candidate ranking. *Relatedness* is widely used by [44, 24, 35, 19, 49, 28], which is to compute the similarity between two KB entries based on the in/out links. *Relatedness* is effective to measure the

relationship between candidates and co-occurring entities in context.

Although some previous work reviewed various ranking methods (unsupervised or supervised) and evaluation results [39, 31], they lack comparing effects of linguistic features systemically. Moreover, Garcia et al. [14] systemically reviewed and evaluated several state-of-the-art link-based approaches, but they did not mention linguistic-based context features.

2.3.3 Previous Work on Japanese Entity Linking

The previous research on Japanese Wikification mostly links mentions to English Wikipedia [13, 46, 26]. That might be impractical because about 44.4% (440k out of 991k articles) of Japanese Wikipedia articles do not have corresponding English Wikipedia articles. The slow development of Japanese Wikification is partly due to the lack of a publicly available Japanese Wikification corpus. Recently, Jargalsaikhan et al. [30] built a Japanese Wikification corpus in which mentions are linked to Japanese Wikipedia entries. However, their baseline method did not achieve good performance because it was a simple unsupervised method that relies on the popularity and category information of candidate entities without the context information of mentions.

2.3.4 Embedding Methods to Model Named Entities

Moreover, in order to better represent the texts and compute the similarity between the context of mentions and Wikipedia articles, the previous work on English EL explores embedding models in candidate ranking [56, 4]. Embedding models provide useful representations of linguistic units such as words [42], entities [56], paragraphs [36], *etc*. This dense and low-dimensional representation is useful to compute the semantic similarities. For example, Blanco et al. [4] propose an EL method for web queries by representing entities and mentions with the averages of their respective vectors. He et al. [27] encode the representations of the input document containing the mention as well as the Wikipedia article by Stacked Denoising Auto-encoders [58]. Sun et al. [56] disambiguate mentions by computing the vector similarity between the two con-

tinuous vector of mentions and candidate entities. In this previous work, a candidate entity is represented with a combination of the sum of vectors of surface words and the sum of vectors of category words of the entity. Mention and the context of mentions are represented with the sum of word vectors and encoded as a continuous vector by a neural tensor network. Since the encoded information of candidate entities is inadequate in the work of Sun et al. [56], we learn a new representation of entities by leveraging their context in the Wikipedia articles.

2.4 Evaluation for Entity Linking

In this chapter, we will introduce the measures and data for evaluating the performance of entity linking. The remainder of this section is structured as follows. First, we introduce the evaluation measure in our experiments (Section 2.4.1). The the following section (Section 2.4.2) consider evaluation datasets in detail.

2.4.1 Evaluation Measures

Our evaluation metric is micro-averaged accuracy, which is used in TAC KBP 2009 and 2010 entity linking task [39]. The metric is computed by,

$$Accuracy_{micro} = \frac{NumCorrrect}{NumMentions} \tag{2.1}$$

2.4.2 Evaluation Datasets

In the following we will introduce the evaluation datasets for English EL and Japanese EL respectively. In Section 2.4.2.1, we will introduce the most widely used EL data set constructed by TAC KBP. In Section 2.4.2.2, a new released Japanese Wikification corpus is considered.

2.4.2.1 TAC KBP Data Set

We use the training data from the 2014 TAC KBP Entity Discovery and Linking (EDL) track [32]. The TAC data set consists of 5878 mentions over 158 documents. The statistics of the

data set is shown in Table 2.1. We use the gold mention query file of the data set.

Table 2.1: Statistics of 2014 TAC KBP Data set.

	PER	ORG	GPE	Total
NIL	1819	591	216	2626
Non-NIL	1390	709	1153	3252
Total	3209	1300	1369	5878

In mention query files, information about one mention is given: the name surface, the document ID, and the position of this mention in the source document (UTF-8 character offsets). For example,

```
<query id="EDL14_ENG_TRAINING_3091">
    <name>St. Andrews</name>
    <docid>WPB_ENG_20101221.0031</docid>
    <beg>1123</beg>
    <end>1133</end>
</query>
```

The example texts in Figure 2.2 are from source documents. The TAC KBP official reference KB is extracted from an October 2008 dumps of English Wikipedia and consists of 818,741 entries.

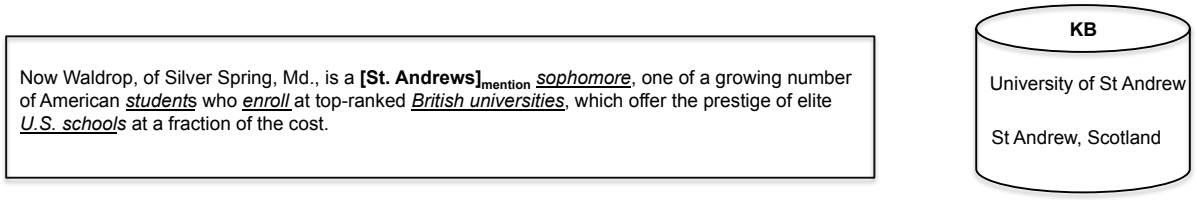

Figure 2.2: Example of documents containing mentions for linguistic features.

Systems are required to generate a link-ID file, which contains pairs of a query and the resolved result (corresponding KB entry ID or NIL). For example, system should output a KB

…<International_Organization wiki="ja:125804" title="国際オリンピック委員会"ＩＯＣ</International_Organization>は新世紀初めに、<Country wiki="ja:270031" title="中国">中国</Country>市場という不安定要素を抱え<Continental_Region wiki="ja:339792" title="アジア">アジア</Continental_Region>の大国での<Game wiki="ja:1421" title="近代オリンピック">五輪</Game>は、政治的側面も無視できない。…

Figure 2.3: A snapshot of annotation document with the document ID of 'PN1a_00008'.

ID e.g., "E0127848" or NIL for the query "EDL14_ENG_TRAINING_3091". In this task, NIL means mentions that do not have entries in the KB. TAC KBP added the mention detection task in 2014. A system should detect possible mentions in raw documents.

2.4.2.2 Japanese Wikification Corpus

We use a Japanese Wikification corpus [30] that consists of 340 newspaper articles from Balanced Corpus of Contemporary Written Japanese (BCCWJ)[1]. Mentions in each document are annotated with fine-grained named entity classes that are defined by Sekine's Extended Named Entity Hierarchy [51][2]. In this corpus, 19,121 InKB mentions are linked to Wikipedia, whereas 6,554 NIL mentions do not have corresponding Wikipedia articles. In total, 7,118 distinct mentions are linked to 6,008 distinct entities. As the corpus was built on top of annotations of named entities, we omit the step of mention detection.

Figure 2.3 shows a snapshot of a news article with the document ID of 'PN1a_00008'. A mention is annotated with the entity class information, the unique ID of the corresponding Wikipedia article, and the title of the Wikipedia article. For example, "IOC" is annotated with the entity class "International Organization", the Japanese Wikipedia article ID "ja:125804", and the Wikipedia article titled "国際オリンピック委員会".

[1] http://pj.ninjal.ac.jp/corpus_center/bccwj

[2] https://sites.google.com/site/extendednamedentityhierarchy/

CHAPTER 3

Exploring Candidate Retrieval for Entity Linking

3.1 Introduction

In the candidate retrieval phase, if candidate retrieval cannot include correct entity into candidate lists, the next candidate ranking process will be fall. Thus, it is common to generate a candidate list as long as possible. A high-recall and short length candidate list may benefit the performance of the next process, candidate ranking, so we need high-recall candidate lists, which contain small amounts of candidate entities.

However, previous studies did not explore the research of candidate retrieval, there are two main problems in previous work:

- Lack of comparable study on the performance of candidate retrieval
- Lack of thorough analysis to mentions unreached to correct entities

In this chapter, we study on conventional candidate retrieval approaches, such as search-based approaches and alias-based approaches. For search-based approach, we investigate methods based on different searching fields and strategies, and evaluate off-the-self search engines

(Section 3.2). For alias-based approach, we firstly discuss how to extract name alias from various resources. Then we construct an alias dictionary containing numerous many-to-many mappings between aliases and name entities (Section 3.3). In Section 3.6, we compare the performance of search-based method with that of alias-based method. We also analyze and classify failure cases unreached to the correct entities in Section 3.6.2.

Moreover, in order to achieve the goal of generating high-recall and short candidate list, we consider a mention extension process (Section 3.4) for improve the recall and a pruning process for eliminating noisy candidates (Section 3.5). In Section 3.7, we evaluate the effectiveness of mention extension and pruning processes.

3.2 Search-based Candidate Retrieval

Given a mention, the most common way is to retrieve the mention in the knowledge base. In this section, we investigate several searching fields and searching strategies in previous studies. We also evaluate an off-the-self search tool (Freebase search API) for retrieving named entities on Freebase. Finally, because this tool has been discontinued, we implement and evaluate an alternative search tool for entities in Freebase.

3.2.1 Searching Fields

3.2.1.1 Searching on Title Field

In the knowledge base (KB), for example, Wikipedia, titles of entries are distinct. In consequence, title strings in the KB are unique. Since names are the primary unique symbols of named entities, the most common way of retrieving candidates is to match strings or characters of mentions with the title (name) field of entries in KB. Figure 3.1 shows an example,

Give the mention “St Andrews”, we can match it on the title field of named entity “St Andrews”. Searching on title field is the most direct way to retrieve candidate entities. The recall of this approach is low but average number of candidates is small (nearly equals 1).

Figure 3.1: Searching the mention on title field of entity entries in the KB.

3.2.1.2 Searching on Document Field

When searching on document field, candidate entities can be retrieved when mention strings occur in the description text of entries in the KB. The surface text of mentions is usually different from that of correct entities, but mentions may appear in the description of entities. Thus, searching on document field will increase the recall. Figure 3.2 shows an example,

Figure 3.2: Searching the mention on document field of entity entries in the KB.

We can find the occurring of mention “St Andrews” in the description of “University of St. Andrews” and consider the latter may be the referent entity of the former. Although the recall could be improved while it may bring lots of noisy candidates.

3.2.2 Searching Strategies

When matching between surface text of mentions and text of different fields of entries in the KB, there are two main strategies: exact matching and fuzzy matching.

3.2.2.1 Exact Matching

Exact matching is the most direct way to compare surface text of mentions with surface text of entries in the KB. Figure 3.3 shows an example of exact matching on title field,

If the mention “St Andrews” and the title field of the entity “St Andrews” are identical, the latter will be selected into the candidate list. The recall of exact matching is low while the average number of candidates are small (nearly equals 1).

Figure 3.3: Searching the mention with exact matching.

3.2.2.2 Fuzzy Matching

Name strings occurring in article text may resemble titles of entries in the KB but may not be exactly matches. Thus, fuzzy matching can increase the possibility of finding the correct entities. Figure 3.4 shows an example of fuzzy matching on title field.

Figure 3.4: Searching the mention with fuzzy matching.

There are several widely-used similarity measures, such as, *cosine similarity*, *dice coefficient*, *jaccard coefficient*, *overlap ratio*, etc. Among them, the cosine similarity is mostly unitized. The performance of search-based method is evaluated in Section 3.6.

3.2.3 Search Engines for Candidate Retrieval

Retrieving candidate entities via a off-the-self searching engine is a common approach as well. Especially, the search APIs of knowledge base are widely used, such as, Freebase Search API, DBpedia Search Interface, etc. In this section, we first evaluate a widely used search API, Freebase Search API on candidate retrieval.

3.2.3.1 Freebase Search API

Before evaluating the Freebase Search API, this section will firstly introduce Freebase. Freebase was a large knowledge base consisting of structure data harvested from many sources, including individual, user-submitted wiki contributions. Google's Knowledge Graph was powered in part by Freebase.

The Freebase Search API provided access to Freebase data [1] given a free text query. The results of query are ordered by relevancy scores.

Next, we will evaluate Freebase Search API on candidate retrieval. We select Freebase Search API because it is a free and quick tool to retrieve entities and related properties of entities, such as birth place, birth day, etc. for person entity.

Here, we utilize two question-answering data sets, including *WebQuestions*[2] and *Wikianswers*[3]. We choose these two data sets because questions are supposed to be answerable by the KB and mostly centered around a single named entity. When constructing evaluation data, we remain the questions in data set which contain named entities. We compare three evaluation measures including recall, average amount of candidates per mention, and the position of correct entity. Table 3.1 shows the performance of Freebase Search API.

Table 3.1: Performance of Freebase Search API on candidate retrieval.

Data Set	**Top-10 coverage**	**AveNumOfCan**	**POS of correct entity**
WebQuestions	95.4%	19.3	1.37
WikiAnswers	90.4%	19.85	1.81

Although search APIs are effective, there are several demerits of applying them on entity linking. The disadvantages are as follow,

1. The coverage of entities is not independent with search APIs. Since those search APIs are specifically designed for its corresponding KBs, it is difficult to apply them on a new domain. Therefore, the recall of candidate retrieval can not be guaranteed.

2. The service time is not independent because Freebase was officially shut down on 2 May 2016. Moreover, Freebase Search API is deprecated from June, 2015. The Knowledge Graph API [4] is a replacement to the Freebase API from 2015.

[1] `https://developers.google.com/freebase/v1/search-overview`
[2] `https://worksheets.codalab.org/worksheets/0xba659fe363cb46e7a505c5b6a774dc8a/`
[3] `http://sparem0507.unas.cz/page.php?q=wikianswers-dataset`
[4] `https://developers.google.com/knowledge-graph/`

Therefore, it is necessary to develop the searchable technology of candidate retrieval, which can be independent with the serve time and the limited of coverage.

3.2.3.2 Designing Search Engines

Thus, we duplicate search APIs by using an open source enterprise search platform, Apache Solr[5]. We index Freebase dump data into Solr, the dump data is in RDF format and contains multiple fields of entities that can be used for searching.

We select the 'OR' combination of name (title) field and keys_en (alias from Wikipedia) field as the final setting We compare our method with the Freebase Search API on WebQuestions and Wikianswers. We use the Mean Reciprocal Rank (MRR) to represent the position of correct entity in ranking list because it is more specific for the evaluation of information retrieval.

Table 3.2 shows our search engine can reach the similar level of the performance of Freebase Search API. Further, we noticed that alias information (keys_en) dramatically improve the coverage than only using titles. Thus, alias information is indispensable factor for candidate retrieval. The Section 3.3 presents a searchable alias dictionary which contains alias information extracting from various resources.

Table 3.2: Performance of proposed approaches based on search engine for candidate retrieval.

Methods	**Top-10**		**Top-100**	
	Coverage	MRR	Coverage	MRR
API	95.4%	0.932	97.3%	0.927
name	85.0%	0.750	92.3%	0.732
name + keys_en	92.1%	0.864	96.2%	0.830

[5] `http://lucene.apache.org/solr/`

3.3 Alias-based Candidate Retrieval

Since some mentions are orthographically different from the titles of their referents in the KB, it may cause failures in approaches based on string/character similarity (search-based). For example, for the mention 'IOC', the approach based on string/character similarity could not reach the correct entity "International Olympics Committee". Therefore, alias information is necessary.

Moreover, as it is mentioned in Section 3.2, string similarity between the entity name (Wikipedia article title) and the mention surface is a common method for generating candidate entities. However, some mentions will be failure by using string similarity.

In addition, it is verified in Section 3.2.3 that search engines are effective, especially by incorporating with alias information. Therefore, it is effective to use aliases extracted from various resources. For example, Wikipedia disambiguation pages, Wikipedia redirect pages, Wikipedia anchors, *etc*. Therefore, in this section, we will introduce how to extract alias from resources and to construct the searchable alias dictionary.

3.3.1 Extracting Alias from Wikipedia

Many aliases or nicknames of named entities are non-trivial to guess. For example "'Big Blue" is the nickname for 'IBM', and "Ginger Spice" is a stage name of "Geri Halliwell". We need additional resources to extract alias information. Therefore, we extract alias form the following resources:

- **Wikipedia redirect pages**

  Redirect pages are one type of linking structure in Wikipedia which we can take advantage of to enrich alias of named entities. A Wikipedia redirect page typically contains only a hyper-link to the reference entity page. It is a pseudo page with a title that is an alternate name or spelling for the entity e.g., "St. Andrews University" for "University of St. Andrews". When a user attempts to access a redirect page like "St. Andrews University", Wikipedia will redirect the canonical page "University of St. Andrews" which

contains the actual contents for describing the entity. Figure 3.5 shows an example of Wikipedia redirect page of “アメリカ” for “アメリカ合衆国”.

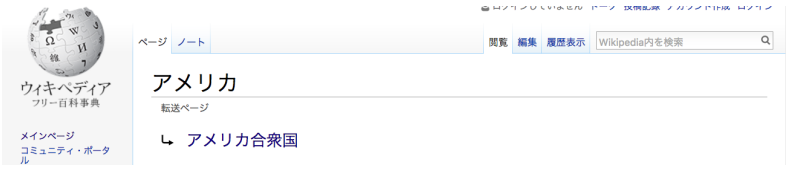

Figure 3.5: Extracting alias from Wikipedia redirect pages

Redirect pages could cover a wide variety of alias information, such as acronyms, synonyms, translations from other languages, common misspellings, etc. By visiting the redirect pages and their associated canonical pages, we can extract these alias under the same entity names.

- **Wikipedia disambiguation pages**

  Disambiguation pages are another type of linking structure in Wikipeida that we can utilize to extract alternate surface texts of named entities. A disambiguation page is created for ambiguous names, such as, names that denote more than two named entities in Wikipedia.

  It is marked with special texts and edited in the form of “Entity name (disambiguation)”. The text body of disambiguation pages contains a list of references to pages of entities that are typically mentioned. For example, in Figure 3.6, the disambiguation page “Apple (disambiguation)” lists more than 40 associated entities, including ‘Plants and plant parts’, ‘Companies’, ‘Films’, ‘Television’, ‘Music’, ‘People’, ‘Places’, ‘Technology’, and others.

  By extracting surface forms mappings from the disambiguation pages, additional aliases can be acquired. Usually, these aliases have some additional information added on to the query, e.g. “Apple Inc.”

- **Wikipedia anchor texts**

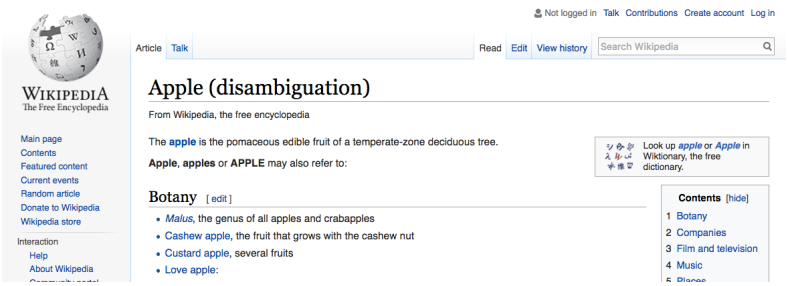

Figure 3.6: Extracting alias from Wikipedia disambiguation pages

For extracting alias from anchors, we base an approach in the previous English entity linking work [54]. This approach gathers hyper-links that jump to each Wikipedia article, and regards an anchor text (surface text) of a hyper link as an alias (possible mention) to the article. For example, we can collect alias mentions to the Wikipedia article "国際オリンピック委員会 (*International Olympic Committee*)": "*IOC*", "*I.O.C*" and "国際オリンピック委員会 (*the Olympic Committee*)". Figure 3.7 shows an example of extracting "Apple" for "Apple Inc.".

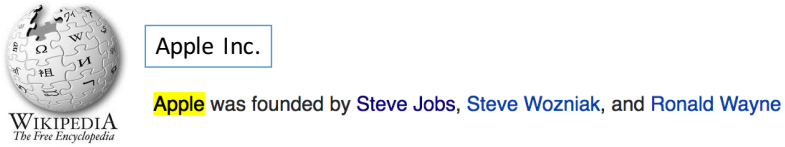

Figure 3.7: Extracting alias from Wikipedia anchor texts

In addition, we calculate the probability $p(e|m)$ of an anchor text m linking to a Wikipedia article e. The probability is estimated as:

$$p(e|m) = \frac{\text{\# times of } m \text{ jumping to } e}{\text{\# occurrence of anchor text } m}.$$

As discussed in [44], this probability reflects the "commonness" or "popularity" of a Wikipedia article. The candidate retrieval looks up each items of alias-entity pairs to reach an entity from a mention. The probability can be used as popularity of entities in the candidate ranking phase.

- **Geonames**

The GeoNames[6] geographical database is specific for geo-political entities. It covers all countries and contains over eleven million place names. The core of GeoNames database is provided by official public sources, the quality of which may vary. The abbreviation of places (countries, US states , etc.) could be used as alias for geo-political entities.

3.3.2 Searchable Alias Dictionaries for Entity Linking

Finally, we obtain a huge number of alias-entity pairs from various resources. For example, we extracted name variations like *Barcodes*, *Toon*, *mags*, *magpies*, and *Newcastle* for *Newcastle United F.C.*, a famous England football club. The next step is to integrate them into search engine to construct a searchable and updatable alias dictionary.

We index the extracted alias-entity pairs into Solr. The incorporating strategy is to merge all alias extracted from various resource. The dictionary is constructed in the form of one KB entry title as the key, aliases of this entity as values. Our alias dictionary contain a huge number of alias-entity mappings so it is adequate for entity linking, even for disambiguate general concept (e.g. the fruit 'apple' and the company"Apple Inc.") For example, for English named entities, it contains 548,084 entities and 2,080,491 aliases.

3.4 Mention Extension

In addition, we extend the alias dictionary because some correct entities cannot be reached only by alias information. We first group mentions in the source document to handle the following scenes:

- **Misspellings**

  The source of text is various in entity linking task, such as new articles, document forums, web pages, etc.Misspelling problems often occurs in informal gene of text (e.g. document forums). For example, the candidate mentions *Gretzy* and *Wayne Gretzky* occur in the

[6] `http://www.geonames.org`

same source document, and they likely refer to the same entity. If we search candidates by using both of them, the possibility of correct entity appearing in the candidate list of *Gretzy* could be increased.

- **Abbreviations**

  The abbreviations of organization or person name is very common in text. Because names generally appear at the first time in a document with the full form, abbreviations of names are used in the remaining part of a document, such as 'BW' for "Barbara Walters".

- **Person name extension**

  In addition, we find shortened instances of person names in text that can be extended. For example, we can not acquire "佐藤秀夫" (Hideo Sato) for "佐藤" (Sato) only by retrieving alias dictionary. For person named entities, this problem can be solved by name extension.

  Since, it is common that a full name of a person appears only once in a document and that the person is referred to by a shortened form (family name or given name). Based on the assumption of *one sense per discourse*, we can assume that shortened forms refer to the same person in a document. Thus, we extend "佐藤" (Sato) to "佐藤秀夫" (Hideo Sato) locally (within a target document) when the latter appears in the target document.

  We recognize family names, and full names of people based on the results of a morphological analyzer, MeCab[7]. We can acquire the detail information of entity type, '姓(family name)' and '名(first name)'. For the mention of full name, we can acquire both. Then, we extend partial names (family names or first names) to full names in the same document if the edit distance between them is less than 2.

We demonstrate the effect of mention extension on improving the recall of candidate retrieval in Section 3.7.

[7] `http://taku910.github.io/mecab/`

3.5 Pruning Noisy Candidates

Since if the candidate retrieval in a entity linking system can not include correct Wikipedia articles in a candidate list for a mention, the subsequent process (candidate ranking) cannot recover from the error. To achieve the goal of high-recall, the previous entity linking systems tend to over-generate candidate entities. However, it needs to minimize the amount of candidates, at the same time, to maximize the possibility that the target entity existence.

Our strategy of pruning noisy candidates is to rank candidates based on document similarity and to remove candidates which are in the tail. The document similarity is calculated between the context of mention and the description text of entities in KB. We leverage two methods to compare the document similarity: TF/IDF cosine similarity and Latent Semantic Index (LSI). We use an off-the-shelf tool, gensim [50], to calculate those two similarities. Then, We demonstrate the effect of pruning on decreasing the amount of candidates in Section 3.7.

3.6 Comparable Experiments between Search-based and Alias-based Methods

In this section, we compare the performance of search-based method and alias-based method, and analyze mentions that are unreached to the correct entities. First, we emphasize the importance of candidate retrieval phase again. If the correct entity is not reached by candidate retrieval, the candidate ranking will be in vain. Thus, we use two performance measures to verify which method is more effective:

Recall Recall is used to evaluate whether the correct entity is included in the candidate list or not. Here, recall is the percentage of mentions that can be correctly linked to the gold entities and calculated by the following equation 3.1,

$$Recall = \frac{Numbers\ of\ mentions\ reached\ to\ the\ correct\ entities}{Number\ of\ mentions} \tag{3.1}$$

Here, high-recall is the goal.

Average number of candidates We also calculate the average number of candidates, because a high recall is easily achieved by increasing the number of entity candidates, e.g., including irrelevant entities in the candidate list. If we generate many noisy candidates in candidate retrieval, the ambiguity and the processing time may be added to candidate ranking. The average number of candidates is used to evaluate how many candidates that a mention reached to. This measure is calculated by the following equation 3.2,

$$AveNumOfCan = \frac{Sum\ of\ the\ length\ of\ each\ candidate\ list}{Number\ of\ mentions} \tag{3.2}$$

Here, small number of candidates is the goal.

3.6.1 Experimental Results

In this section, we show the experimental results of search-based method and alias-based method on a Japanese Wikification corpus [30]. We use two baseline approaches for search-based method:

Exact matching We apply exact matching between surface forms of mentions and titles of Wikipedia articles.

Cosine similarity For applying the approach based on cosine similarity, we use a simple and efficient tool, `SimString` [47]. Given a query string, this tool can retrieve strings that have similarity values greater than a specified threshold. The tool provides common similarity measures including cosine similarity, jaccard similarity, overlap coefficient, *etc*. Since the title of entry is unique in the referent KB, we extract all titles of entities. Then we index them as a `SimString` database. Here, we take advantage of cosine similarity on tri-grams of mentions and named entities. Figure 3.8 helps account for the calculation processing.

We normalize surface forms of mentions to eliminate differences between half-width characters and full-width characters in advance. We compare cosine similarity with thresholds between 0.5 and 0.9, respectively.

Table 3.3 shows the recall and average number of candidates with different thresholds of

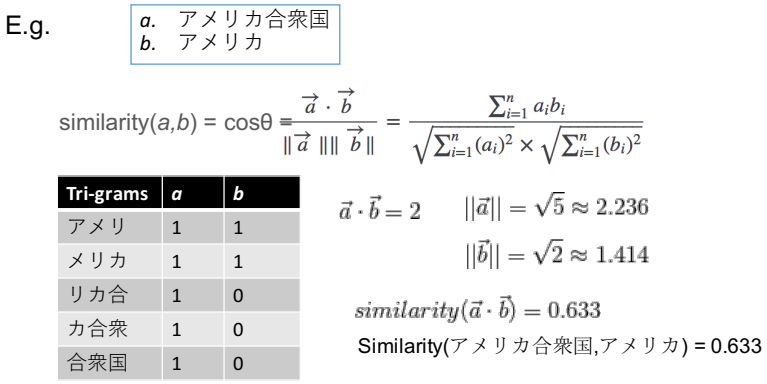

Figure 3.8: Example of calculating cosine similarity of tri-grams of mentions and named entities.

cosine similarity. We found that the increase of recall is much less than the increase of number of candidates. Especially, when setting the threshold to 0.5, the recall (93.3%) is slightly increased by dramatically increasing the number of entity candidates (523.76). For the alias-based method, we look up the alias dictionary with the mention with exact matching. We compare the alias dictionary method with the string similarity method.

Table 3.3 also indicates that the alias dictionary based on anchor texts is suitable for achieving a high-recall (91.98%) with the small number of candidates per mention (17.58). Although the recall of cosine similarity (threshold=0.5) is about 1.4 higher than the recall of the alias dictionary, it brings a huge number of irrelevant candidate entities.

Moreover, we extend mentions of person names before retrieving on the alias dictionary. Extending family names and given names to their full names further improved the recall (94.14%) with a little increase of candidates per mention (17.79). Therefore, we use the alias-based method with the name expansion step in the evaluation of candidate ranking in Chapter 4.

3.6.2 Error Analysis

Table 3.4 summarizes error types of the proposed alias-based approach. The majority (77.56%) of the errors was caused by the lack of alias information. For example, the candidate generation could not retrieve “聖王(百済) (King Seong (Baekje))” from the mention “明王(Ming Wang)”,

Table 3.3: Comparable results of search-based and alias-based approaches on InKB mentions.

Methods	**Recall**	**AveNumCan**
cosine (Threshold=0.9)	74.49%	1.58
cosine (Threshold=0.8)	76.80%	4.96
cosine (Threshold=0.7)	82.50%	27.12
cosine (Threshold=0.6)	89.01%	123.55
cosine (Threshold=0.5)	93.33%	523.76
alias dictionary	91.98%	17.58
alias dictionary (+ person name extension)	94.14%	17.79

which was not included in the anchor texts. In order to address these kind of errors, we are required to collect more anchor texts not only from Wikipedia but also from other Web pages.

Furthermore, 2.59% of the errors were due to the errors in the original corpus [25]. For example, "新華社電(Xinhua reported)" is annotated with the incorrect boundary while the correct mention is "新華社(Xinhua)". Approximately 17.14% of the errors were caused by orthographic variations between Kanji and Katakana/number. For example, the system could not retrieve the correct entity "頬(cheek)" from the mention "ほお(cheek)" because of the difference between Kanji and Kana spellings.

Similarly, we found that about 1.23% of the errors were caused by spelling variations of Kanji, e.g., "柳沢(Yanagisawa)" and "柳澤(Yanagisawa)". We can handle these cases by forcing these spelling variants to be included in the alias dictionary. In addition, about 1.48% of the errors were caused by transliteration; for example, referring the entity "Love you" from a transliterated mention "ラブ・ユー". It may be possible to integrate a transliteration model in the candidate retrieval. However, we leave these treatments as a future work, which may increase the number of false entities in candidate retrieval.

Table 3.4: Different types of error examples of proposed alias-based method.

Error Class	**Ratio**	**Mention Examples**	**Gold Entity Examples**
Lack of aliases	77.56% (629/811)	明王(Ming Wang)	聖王(百済) (King Seong (Baekje))
Orthographic difference between Kanji and katakana/number	17.14% (139/811)	ほお(cheek)	頬(cheek)
Errors in the original corpus (e.g. errors of mention detection)	2.59% (21/811)	新 華 社 電(Xinhua reported)	新華社(Xinhua)
Transliteration	1.48% (12/811)	ラ ブ ・ ユ ー(Love you)	Love you
Alternate spelling	1.23% (10/811)	柳沢(Yanagisawa)	柳澤(Yanagisawa)

3.7 Evaluating Mention Extension and Pruning

In order to improve the recall and eliminate noisy candidates, we design two processes: mention extension (Section 3.4) and pruning (Section 3.5). Therefore, in this section, we evaluate and verify the effectiveness of those two processes.

Here, we use 2014 TAC KBP training data set. We process one mention at one time on 2014 TAC KBP training data set. For each mention, we search both the original mention and the extended mentions. Table 3.5 shows that the alias-based approach achieved 96.23% recall on the training set. The average number of candidate per list is 186. After adding mention extension, we achieved 98.43% recall. The average number of candidate per list is 245.

We apply Latent Semantic Index (LSI) to rank each candidate list and retain the top 50 candidates as the final candidate list. LSI achieved 97.39% recall on the training set while TF/IDF got 74.84%. The average number of candidates per list is 41 by using LSI. According to our preliminary experiment, we found that Latent Semantic Index (LSI) is superior to TF/IDF

Table 3.5: Performance of candidate retrieval approaches on InKB mentions in TAC KBP Data Set.

Methods	**Recall**	**AveNumOfCan**
alias dictionary (all)	96.23%	186
alias dictionary (all) + Mention Extension (ME)	98.43%	245
alias dictionary (all) + ME + Pruning (with TF/IDF cosine similarity)	74.84%	46
alias dictionary (all) + ME + Pruning (with LSI cosine similarity)	97.39%	41

cosine similarity.

3.8 Summary

In candidate retrieval phase, we need high-recall candidate lists, which also contain small amounts of candidate entities. However, previous works did not explore the research of candidate retrieval. Therefore, we investigate conventional candidate retrieval approaches, such as search-based method and alias-based method. After evaluation experiments, we find off-the-self search engines could perform excellently on candidate retrieval. We also verify that our approach based search engine can achieve the nearly same good performance for candidate retrieval with a well-done search engine, Freebase Search API.

Moreover, for alias-based approach, we discuss several resources for extracting alias for named entities. We construct an alias dictionary containing numerous many-to-many mappings between aliases and name entities. Additionally, we compare the approach based on string/character similarity (search-based) with the approach based on alias dictionary (alias-based). Finally, we confirm that applying fuzzy matching on alias dictionary can provide a high-coverage candidate sets containing seldom noisy candidates, especially adding mention extension and pruning.

CHAPTER 4

Exploring Candidate Ranking for Entity Linking

4.1 Introduction

In this chapter, we formulate the candidate linking task and utilize a supervised model to rank candidate entities (Section 4.2). We study and apply various prevalent linguistic features for English EL in multiple aspects, such as surface string similarity, text similarity, entity characteristics, etc. (Section 4.3). The effectiveness of linguistic features have been verified on TAC KBP data set (Section 4.4). What's the more, we deeply study the performance of context related features in Section 4.4.3. In addition, we analyze different performance of features on mentions that have different entity types in Section 4.4.2.

Furthermore, we reuse several effective features for Japanese entity linking and confirm their strength via experiments on Japanese Wikification corpus (Section 4.5). Ultimately, we evaluate the overall performance of pipeline system on TAC KBP data set (Section 4.6). Finally, we analyze the failure errors of the proposed system (Section 4.6.3).

4.2 A Supervised Learning Model

We formulate the ranking problem similar to [6] and [40] in candidate ranking phase. We generate a score function $f(m, e)$ based on features that extracted from a mention m and a candidate entity e. We select the candidate $\hat{e}$ with the highest score from a candidate list E_m, according to the score,

$$\hat{e} = \underset{e \in E_m}{\operatorname{argmax}} f(m, e). \tag{4.1}$$

Therefore, the scoring function $f(m, e)$ should be trained such that the correct Wikipedia article $\widehat{e}$ is linked to the mention m. We use SVM$^{\text{rank}}$ [33] with the linear kernel to handle the optimization problem.

4.3 Studies of Linguistic Feature

We utilize powerful features explored by state-of-the-art English EL systems and develop several new features for candidate ranking. Linguistic features could be mainly divided into several classes: surface related features, context related features, entity type related features and other features (e.g. entity popularity feature).

4.3.1 Surface related Features

Surface related features focus on the surface properties of surface text of mention and titles of entities in KB. We summarize the basic features in Table 4.1.

The *IsAcronym* and *IsAbbrMatch* features [10] capture characteristics of acronyms. For example, given a mention 'WTO', acronym features can detect "**W**orld **T**rade **O**rganization". The *SurfaceSimScore* and *EqualWordNumSurface* features [61] calculate how similar is the mention surface to the KB title. The *TokenLenInCandidate* and *CharLenInCandidate* [16] count the terms and characters of the KB titles. We also incorporate other similarity features used in previous work [16, 9], such as dice coefficient scores and jaccard index scores.

Table 4.1: Basic Features of Candidate Ranking Module.

Feature	**Description**
SurfaceSimScore	Levenshtein edit distance between the KB title and the mention surface
EqualWordNumSurface	Maximum of count of exact matches between mentions in the same group and the KB title
HasQueryGroup	Whether the KB title belongs to a mention group
QueryGroupMatch	Whether the KB title matches any surface in the same group
QueryGroupOverlap	Whether a surface in the same group is substring of the KB title, or vice versa
QueryGroupMaxSim	Maximum similarity between the KB title and surfaces in the same group
TokenLenInCandidate	Term count in the KB title
CharLenInCandidate	Characters count in the KB title
IsAcronym	Whether the mention surface is an acronym
IsAbbrMatch	Whether the capital character of the KB title match any surface in the same group
DiceTokenScore	Maximum value of the dice coefficient between the KB title token set and the surface token set
DiceToken JaccardTokenScore	Whether DiceTokenScore is above 0.9 Maximum value of jaccard index between the KB title token set and the surface token set
DiceCharacterScore	Maximum value of dice coefficient between the KB title character set and the surface character set
DiceCharacter	Whether DiceCharacterScore is above 0.9
DiceAlignedTokenSocre	Maximum character dice coefficient of left and right aligned token sets
DiceAlignedToken	Whether DiceAlignedTokenScore is above 0.9
DiceAlignedCharacterSocre	Maximum character dice coefficient of left and right aligned character sets
DiceAlignedCharacter	Whether DiceAlignedCharacterSocre is above 0.9

4.3.2 Context related Features

We extract context information from both mention source documents and texts of knowledge base entries (candidates) for disambiguation.

4.3.2.1 Title Appearance

The title appearance feature [16] are related with the appearance of a candidate title in the source document, or the appearance of mentions in candidate texts. For example, if a given mention is the family name of a person like *Daughtry*, the title of a candidate like *Chris Daughtry* may appear in the source document. Similarly, this given mention *Daughtry* may occur in the text of KB entry *Chris Daughtry*. Among them, a salient feature [16] detects disambiguators in candidate titles, e.g., *magazine* in People (magazine) and *basketball* in Maurice Williams (basketball).

4.3.2.2 Text Similarity

We use two measures to compare the text similarity between source documents and KB texts: cosine similarity with TF/IDF [61] and dice coefficient [9]on tokens. Since the first paragraphs of KB and text surrounding mention are supposed to be more informative, we consider using different ranges of source documents and KB texts. We divide text in a source document into local text (window size = 50 tokens) and global text (the whole source document), and use the first paragraph and the whole KB text receptively. We further consider only use words and entities as follow,

Bag-of-Word Similarity To illustrate these features, we use the following instance,

> The **IOC** is facing the elements of instability from the market of China from the beginning of this new century. We can never ignore this kind of political aspects for the Olympics at the major Asian nation.

Here, underlined words denote named entities. We also show a snippet of the corresponding Wikipedia articles "*International Olympic Committee*":

> International Olympic Committee is an organization who hosts the modern Olympics and unifies various international sports groups attending the Olympics Games. IOC is a non-profit organization (NPO) of the non-governmental organizations (NGO), however, it may be always misconstrued as one of the international authorities because it has obtained credential of United Nations General Assembly Observers at 2009.

Here, underlined words are anchor texts (hyper-links). In this work, we utilize the whole texts that mentions exist in as the context of mentions. This feature measures the similarity between texts that the mention exist in and the contents of the Wikipedia article. For example, we assess the similarity between the set of words {"*face*", "*market*", ...} extracted from the context of mention, and the set of words {"*modern*", "*Olympic*", ...} extracted from the Wikipedia article.

Bag-of-Entity Similarity This is similar to Bag-of-Word Similarity, except that we only consider named entities in the given text and anchor texts of the Wikipedia article. For example, we compute the similarity between the set of entities {"*China*", "*Olympic*", ...} extracted from the context of mention, and the set of anchor texts {"*Olympic Games*", ...} extracted from the Wikipedia article.

Similarity of part-of-speech tokens We hypothesize that nouns and verbs could contribute more on disambiguating than other type of words. Therefore we collect this two type of tokens in context and calculate cosine similarity with TF/IDF weighting respectively.

4.3.2.3 Entity Mention Occurrence

Named entities in mention context are more salient than common words. This feature is used in [10], which could capture the count of co-occurring named entities between source documents and KB texts. For example, for a given mention 'Obama', the named entities "White House" and "'United States" may appear in both the source document and the KB text if it refers to the American president "Barack Obama".

4.3.2.4 Entity Fact

The infobox of KB contains important attributes of named entities. For example, for the entity "Apple Inc.", we can extract attributes, such as Founder (*Steve Jobs*) and CEO (*Tim Cook*). Therefore we extract fact texts from the referent KB and check whether those fact texts appear in the source documents, which is inspired by [10]

4.3.2.5 Document Topics

Semantic information can not be detected by simply counting occurrences of tokens, n-grams, and entities. Therefore we use topic models to discover the implicit topics of source documents and KB texts. We train LDA (Latent Dirichlet Allocation) model with gensim [50], which provides a fast online LDA model. We treat each KB entry as one document and use two different corpus for training. Zhang et al. [60] trained a topic model on the KBP knowledge base, we additionally train another topic model on the latest wikidump[1]. The KBP knowledge base is a partial KB of Wikipedia and contains about one third of Wikipedia entities. We use two similarity measures to check the topic similarity between source documents and KB entries including cosine similarity and Hellinger distance. We also generate topics of partial text surrounding mention as the local topics to compare with using the whole source document (global topics).

4.3.3 Entity Type related Features

4.3.3.1 4-Class Entity Type

We use matching on entity types to detect whether the KB entity type is identical to the mention entity type, which is similar with [10]. For example, the mention "St. Andrew" is an ORG (Organization) entity in the first text in Figure 2.2. The candidate "University of St.Andrew" (ORG) is more likely than "St. Andrew, Scotland" (GPE) because of entity type matching. Therefore, we predict named entity types for both non-NIL mentions and NIL mentions in the

[1] http://dumps.wikimedia.org/enwiki/20140707/enwiki20140707-pages-articles.xml.bz2

final output results.

Since the KBP KB provides entity type information, we concern that it is more credible to predict non-NIL mention types by using KBP KB labeled type. However there are almost 64.9% 'unknown' entities in the official KBP KB. It means that we need to re-tag remaining 'unknown' entities. Different from [10], we use the re-tag entity types according to our re-tagging results.

Clarke et al. [7] classified unknown type entities based on the infobox class in the KBP KB. They also found that matching between infobox classes and entity types approximately has no ambiguity. Different from [7] classified infobox class using learning method, we resolved around 2370 infobox classes manually. Our re-tagged results contain four entity types: PER (person), ORG (organization), GPE (geo-politics), and MISC (none of the aboves). Table 4.2 shows the ratio of before and after re-tagging process.

Table 4.2: Entity types before and after re-tagging.

Type	**KBP KB**	**Our System**
PER	14.0%	23.5%
ORG	6.8%	12.3%
GPE	14.2%	22.0%
UKN	64.9%	0.0%
MISC	0.0%	42.2%

4.3.3.2 Fine-grained Entity Class

Different from the general description in the category of a Wikipedia article, entity classes could specifically represent the type information of entities. It was verified that using a finer-grained entity class set is more suitable for English Wikification [38, 37] than using a coarse-grained entity class. Name entities in the Japanese Wikification corpus [30] has been annotated with a fine-grained entity class label, called Sekine's entity class [51]. Suzuki et al. [57] automatically label Wikipedia articles with Sekine's entity class based on a multi-label classification method.

Here, each mention in corpus may have more than one entity class.

This feature indicates whether the Sekine's entity class of a mention is the same with one of the semantic classes of a Wikipedia article. For example, a Wikipedia article "*International Olympic Committee*" has Sekine's entity class of "Sports Organization Other" assigned while the mention"*IOC*" is labeled as "International Organization" in the corpus [30]. In this case, this feature does not fire for the Wikipedia article "*International Olympic Committee*".

4.3.3.3 Entity Category

This feature counts how many words in category names of a Wikipedia article also appear in text. For example, the Wikipedia article "*International Olympic Committee*" belongs to categories "*Olympic movement*", "*Committees*", *etc.*, and some words in the category names, such as "*Olympic*", also appear in text. This feature reflects such overlaps.

4.3.4 Entity Popularity Features

This is the probability $p(e|m)$ of an anchor text m linking to a Wikipedia article e. The probability is estimated as:

$$p(e|m) = \frac{\text{\# times of } m \text{ jumping to } e}{\text{\# occurrence of anchor text } m}.$$

As discussed in [44], this probability reflects the 'popularity' of a Wikipedia article.

4.4 Evaluation Linguistics Features on English Entity Linking Data Set

4.4.1 Addition Experiments of Features

Since we focus on the ranking performance of each group of linguistic-based context features, we compute the accuracy of mentions that are resolved by our system. In order to eliminate the effect of feature combination, we add only one feature group to the basic feature group each

time. We perform 5-fold cross-validation on the training set. Table 4.3 shows micro-averaged accuracies of feature addition experiments.

Table 4.3: Feature additive test results on TAC KBP Data Set.

Feature Group	**Non-NIL**	**NIL**	**ALL**
Surface	0.5910	0.7000	0.6394
Title Appearance	0.6138	0.7086	0.6558
Entity Fact	0.6024	0.6664	0.6306
Entity Mention Occurrence	0.6134	0.7668	0.6814
Document Similarity	0.6594	0.7733	0.7059
Document Similarity (LOCAL)	0.6422	0.7403	0.6860
Document Similarity (GLOBAL)	0.6474	0.7881	0.7096
Document Topic	0.6322	0.6912	0.6580
Document Topic (WIKI)	0.6224	0.6912	0.6528
Document Topic (KBP)	0.6280	0.6880	0.6544
Similarity of POS	0.6224	0.7420	0.6754
Similarity of POS (Noun)	0.6236	0.7364	0.6736
Similarity of POS (Verb)	0.5986	0.6970	0.6416
Type	0.5908	0.7030	0.6400
All Features	0.7330	0.7454	0.7378

4.4.2 Experiments Results of Different Entity Types

In this section, we evaluate features on subsets of mentions grouped by entity types. We plot the performance on three types respectively in Figure 4.1,4.2, and 4.3. From the results, we found that the performance on PER entity is better than ORG and GPE entity. In our data set, the amount of PER mentions is two times larger than the amount of ORG mentions or GPE mentions. Moreover, simple string matching linguistic features fail to disambiguate ORG and

GPE entities because of multiple name variation, especially the confusion between different entity types. For example, city names could be part of sport teams (*Orlando* is short for *Orlando Magic*) and people names could be part of company names (*Disney* is short for *Walt Disney Company* or *Walt Disney Animation Studio*).

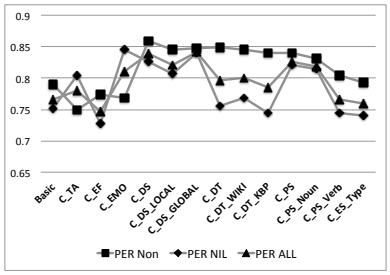

Figure 4.1: Person Type.

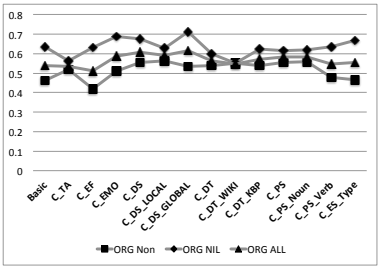

Figure 4.2: Organization Type.

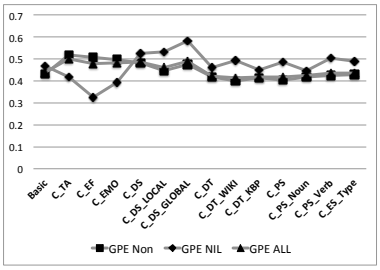

Figure 4.3: Geo-political Type.

4.4.3 Experiments of Context related Features

In order to clarify feature effects, we divide features into more fine-grained groups, such as local topics (DT_WIKI_LOC, DT_KBP_LOC), global topics (DT_WIKI_GLO, DT_KBP_GLO), and document similarity by using the first paragraph of KB texts (DS_CON_FIR) or the whole KB texts (DS_CON_ALL). Table 4.4 shows the increment of each fine-grained feature group to basic features on non-NIL mentions before NIL classification processing, and feature group names are capitalized referring to Table 4.3.

4.4.4 Discussions

Basic features only including features related to surface similarity are not effective enough to find correct entities. Features based on document similarity (both words and part-of-speech levels), named entities co-occurrence, and document topics contribute the most gains of accuracy.

Document similarity In both document similarity and document topics, global features are

Table 4.4: Accuracy increment on non-NIL mentions before NIL classification.

Fine-grained Feature Group	**Accuracy Increment**
C_DS_LOCAL	0.0736
C_DS_GLOBAL	0.1044
C_DS_CON_FIR	0.0214
C_DS_CON_ALL	0.0726
C_DT	0.0582
C_DT_WIKI	0.0338
C_DT_KBP	0.0576
C_DT_WIKI_GLO	0.0576
C_DT_WIKI_LOC	0.0344
C_DT_KBP_GLO	0.0534
C_DT_KBP_LOC	0.0510
C_PS_Noun	0.0658
C_PS_Verb	0.0150

better than local features. Since we leverage measures based on bag-of-words calculation, the larger text of context contains more co-occurring words than the window-size context. Although we suggest that the first paragraph in the KB is much informative, using the whole KB text (DS_CON_ALL) is much better than only using the first paragraph (DS_CON_FIR). We found that, in the KBP KB, several first paragraphs of KB texts are very short, sometimes only one sentence. For example, for "Jeff Perry (American actor)", there is only one sentence, "Jeff Perry (born August 16, 1955 in Highland Park, Illinois) is an American character actor." We found that around 28.74% entries of the KBP KB contain one simple sentence in the first paragraph.

Document topics Moreover, based on the results in Table 4.4, the increment of global topics is more than that of local topics by 0.024 (KBP corpus). Since the distribution of partial

document topics is inconsistent with document topics, global topics can better represent the semantic context of a mention.

Although the KBP KB contains around one third entities of Wikipedia, the performance on the KBP KB corpus is better because we use the KBP KB as the entities database. We found that words of KBP KB topics could represent source document better than using the Wikipedia corpus for some entities. For example, *Salvador Dali* entity is a painter, who is also known for writing and film. Words of top topics are given by the KBP LDA corpus of this entity are *film*, *book*, *album*, *play*. However, words given by the Wikipedia LDA corpus are *Louisiana*, *disease*, *species*, and so on. The Wikipedia LDA corpus is not well-built, which may also affect the performance, because we follow an off-the-shelf training process.[2]

However, the performance on Wikipedia corpus is more effective on NIL. Contrast to KBP KB, Wikipedia contains much more NIL entities.

Similarity of POS tokens We found that nouns and verbs are more informative than other type of words. Nouns contain more information than verbs because named entities are more salient.

4.5 Evaluation Linguistics Features on Japanese Wikification Corpus

We conducted the feature study on each feature set by a 5-fold cross validation. We applied experiments on NonNILs, entities that exist in the Wikipedia. We begin with the string similarity feature set, added various features to it incrementally and reported their impact on performance.

From the results of Table 4.5, we found that our system obtained the performance with approximately 3 percents higher than previous work by only using string similarity features. Adding popularity features slightly further improved the performance.

We observed significant improvement when adding Bag-of-words features. However, only adding Bag-of-entities features led the performance to drop by about 9 percents. Adding both

[2]https://radimrehurek.com/gensim/wiki.html

Bag-of-words and Bag-of-entities together, the system performance is improved to 84.88%.

Moreover, adding the features of fine-grained entity class is better than adding the category features. Therefore, we remove the category feature in the remaining experiments.

Table 4.5: Performance on NonNILs by incremental feature study on Japanese Wikification Corpus.

Feature sets	**Accuracy**
Popularity[30]	53.31%
StringSim (S)	56.13%
S+Popularity (P)	61.87%
S+P+Bag-of-words (Bw)	84.48%
S+P+Bag-of-entities (Be)	75.26%
S+P+Bw+Be	84.88%
S+P+Bw+Be+Entity Category (Cate)	84.77%
S+P+Bw+Be+Entity Class (Class)	85.54%
S+P+Bw+Be+Cate+Class	85.37%

4.6 Overall Evaluation of Proposed Supervised English EL System

4.6.1 NIL determination

We use two heuristic rules to determine the final label for a mention:

- Mentions are labeled as NIL if there is no candidate in the candidate list.
- Mentions are labeled as NIL if the ranking score of the top 1 candidate is below a threshold.

4.6.2 System Performance in 2014 TAC KBP Workshop

In order to verify the performance of the EL system, we attend the 2014 TAC KBP English Entity Discovery and Linking workshop. The overall evaluation results are given in [32], The evaluation report [32] also contains the results of mention detection component and NIL clustering, in order to describe our system performance on EL, we only show those results in Table 4.6. Our system provides a good performance on English EL task. After adding document topics, the system can be improved by 2 percentages.

Table 4.6: Linking performance in 2014 TAC KBP Workshop

System	**Accuracy**
Max	0.865
Ours (w/o Document Topics)	**0.806**
Median	0.704
Ours (+ Document Topics)	**0.826**

4.6.3 Error Analysis

We analyze some error instances from the results on English data set. Although our current context related features is a effective feature set, some mentions are still fail. Here are two main failure reasons:

- **Surface variation information are not captured**

  Given the bellow text example,

  > That's higher than the 82 percent occupancy and $462.41 average room rate at New York luxury chain hotels, according to **Hendersonville**, **Tenn.** -based Smith Travel Research Inc.

  the mention "Hendersonville" was incorrectly linked by system to the entity "Hendersonville, North Carolina " while its correct entity is "Hendersonville, Tennessee". We

find that they are both location entities, and share common words about locations. Furthermore, the KB texts length of "Hendersonville, North Carolina" is longer than "Hendersonville, Tennessee". We fail to capture the abbreviation "Hendersonville, Tenn." of "Hendersonville, Tennessee", which cause we miss the appearance the correct entity in the source text.

- **Confusion of Similar Entities**

  If the correct entity and system failure output share the similar surface, similar entity type and similar description in the KB, it is very difficult to distinguish them only based on current context related features. For example,

  > **Roy Edward Disney** was born in Los Angeles on January 10, 1930. Sevrn years earlier, **his father Roy** and uncle Walt had co-founded the Disney entertainment.

  Here, since the failure output "Roy Edward Disney" appeared in the source text of the mention 'Roy', it will mislead the system to link the wrong one.

4.7 Summary

In this chapter, we utilize a supervised model to rank candidate entities. We study and apply various prevalent linguistic features for English EL. Furthermore, we reuse several effective features for Japanese EL and confirm their strength via experiments. Ultimately, we evaluate the overall performance of pipeline system on English corpus.

CHAPTER 5

Embedding Features for Candidate Ranking

5.1 Introduction

In this Chapter, we describe the embedding models for constructing low-dimensional vectors for the context of mentions and Wikipedia articles (Section 5.2). Moreover, we design embedding features based on the embeddings we learned (Section 5.3). Furthermore, we explore and verify their effectiveness for candidate ranking (Section 5.5). The proposed Japanese EL system incorporating with embedding features outperforms previous work with huge margins (Section 5.6). Finally, we analyze the errors of the proposed Japanese EL system (Section 5.6.2).

5.2 Learning Embedding Models

Because context words are very effective for disambiguating entities. For example, in the sentence "The I.B.M. is the world's largest organization dedicated to the art of magic.", the context word "magic" helps link the mention "I.B.M." to the correct entity "International Brotherhood

of Magicians". Therefore, we jointly learn embeddings of entities and words on a unstructured corpus. The new entity embeddings can model latent semantics between entities and context words. Then we can realize accurate computation of the semantic similarity among word pairs, entity pairs and word-entity pairs.

5.2.1 Learning Word and Entity Embedding

Our new representation of entities and words is based on the idea of predicting the context words of entities, which can be realized by the skip-gram model. Figure 5.1 gives a good account of skip-gram model. The model uses the current word to predict the surrounding window of context words.

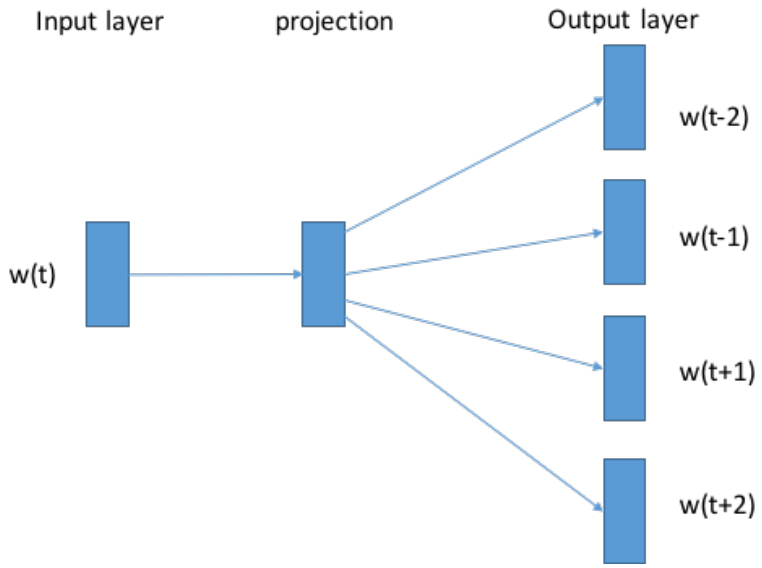

Figure 5.1: Skip-gram model for learning word vectors

Equation 5.1 defines the objective function that we use.

$$L = \sum_{e_i, C \epsilon E} \sum_{w_c \epsilon C} \log P(w_c|e_i) \tag{5.1}$$

where E denotes a set of texts, each of which contains a entity e_i and its context words set C. C contains the previous c words and the next c words. In this model, we also aim to maximize

$$p(w_c|e_i) = \frac{\exp(\mathbf{e_i} \cdot \mathbf{w_c})}{\sum_{w \epsilon W} \exp(\mathbf{e_i} \cdot \mathbf{w})} \tag{5.2}$$

where $\mathbf{e_i}$ and $\mathbf{w_c}$ are the vector representation of an entity e_i and its neighbor word w_c inside the context window while $p(w_c|e_i)$ is the probability to have w_c in the context of e_i. The sum is over the whole vocabulary.

Wikipedia effortlessly facilitates us to learn the model because entities in Wikipedia are automatically tagged as anchor texts. Before learning on Wikipedia, several preprocessing steps are necessary. Figure 5.2 help to demonstrate why and how to make preprocessing.

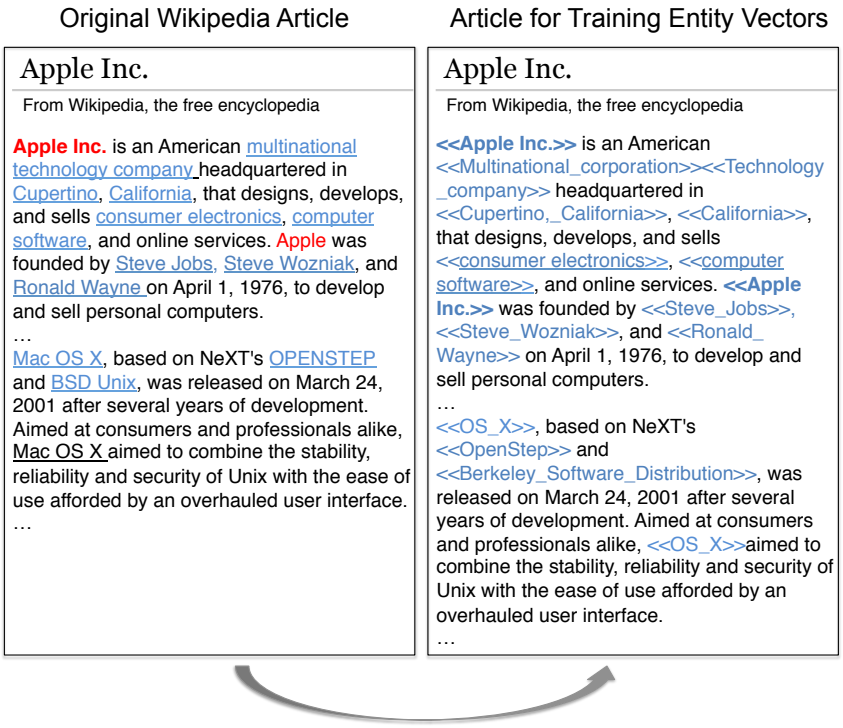

Figure 5.2: An example of preprocessing on a Wikipedia article 'Apple Inc.'.

In a Wikipedia article, referent entities are tagged by hyper-links in anchor texts. Therefore, an entity has various expressions in different articles. For example, in the article "Apple Inc.", the entity "Berkeley Software Distribution" is expressed to "BSD Unix". We replace anchor texts with their referent entities to guarantee the consistency of entity expressions in the learning corpus.

In an article, the title entity need not be formed in anchor texts, e.g. "Apple Inc." and "Apple" in Figure 5.2. Moreover, if an entity is tagged in the beginning part of the article, the

remaining expressions of this entity are often omitted in the following part, e.g. “Mac OS X” in Figure 5.2. To improve the completeness of the learning corpus, we assign anchor texts for those entities of which anchor texts are omitted . We collect all existing entity anchors in the same article. We add anchor texts by searching the longest matched anchors in the existing entity anchors. To distinguish words and entities, we use double angle quotes to represent the text range of entities, such as $\langle\langle$Apple Inc.$\rangle\rangle$, $\langle\langle$Cupertino, California$\rangle\rangle$, etc.

The number of dimensions d of the embedding is set to 200. We set the size of context window c = 10 and the negative samples as 5. For Japanese entity linking, we use a well-learned word and entity embeddings in [57]. Finally, we got 911,965 word and entity vectors.

In this new entity embedding model, entities with similar meaning are close to each other in the same vector space in the same way of words. Moreover, we can get the entity analogy, e.g. Hokkaido - Sapporo + Okinawa $\approx$ Naha.

5.2.2 Learning Paragraph Vectors

Instead of averaging word embeddings, paragraph vectors can directly represent document after learning. PV can inherit the semantics of the words of the word embeddings. The good performance has been verified in previous works [36]. However, PV has not been applied to Entity Linking in previous works. Therefore, we explore the effectiveness of paragraph vectors in EL and compare with embedding of words and entities.

The paragraph vector [36] is a powerful unsupervised method of learning representations of arbitrary lengths of texts and has the advantages of simplicity and versatility. We except to use paragraph vectors to model KB articles based on the Distributed Memory Model of Paragraph Vectors (PV-DM) model [36], which is extended from CBoW in word2vec [42, 43].

Figure 5.3 explain the learning process of distributed memory model. In the Paragraph Vector framework, every paragraph is mapped to a unique vector, represented by a column in matrix **D** and every word is also mapped to a unique vector, represented by a column in matrix **W**. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

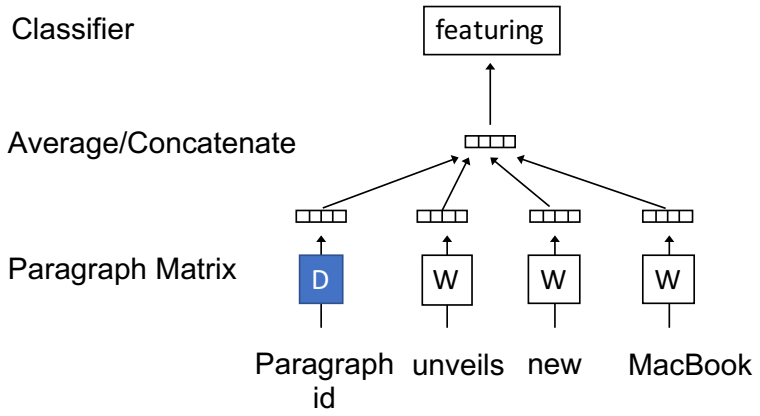

Figure 5.3: A Distributed Memory Model (PVDM) for learning paragraph vectors.

We set the dimensions of document vectors is 400. The size of context window c is set to 5 and the negative samples as 5. Finally, we got 991,284 paragraphs vectors.

5.3 Designing Embedding Features

We design three embedding features based on learned embedding models. To illustrate those features, we use the following instance,

> The **IOC** is facing the elements of instability from the market of China from the beginning of this new century. We can never ignore this kind of political aspects for the Olympics at the major Asian nation.

Here, underlined words denote named entities. We also show a snippet of the corresponding Wikipedia articles "*International Olympic Committee*":

> International Olympic Committee is an organization who hosts the modern Olympics and unifies various international sports groups attending the Olympics Games. IOC is a non-profit organization (NPO) of the non-governmental organizations (NGO), however, it may be always misconstrued as one of the international authorities because it has obtained credential of United Nations General Assembly Observers at 2009.

Here, underlined words are anchor texts (hyper-links). In this work, we utilize the whole texts that mentions exist in as the context of mentions. We consider the following features.

- **Similarity between Average vectors of Word Vectors** We calculate cosine similarity between average of word vectors. For example, cosine similarity between the average vector of $\mathbf{w}_{face} + \mathbf{w}_{market} + \ldots$ for text and average vector of $\mathbf{w}_{modern} + \mathbf{w}_{Olympic} + \ldots$ for Wikipedia article

- **Similarity between Average vectors of Entity Vectors** Wu calculate cosine similarity between average of entity vectors

  For example, cosine similarity between the average vector of $\mathbf{e}_{China} + \mathbf{e}_{Olympic} + \ldots$ and the average vector of $\mathbf{e}_{Olympic\_Games} + \ldots$

- **Paragraph Vector Similarity** We calculate similarity between paragraph vectors For example, cosine similarity between paragraph vector for text and paragraph vector for Wikipedia article.

5.4 Evaluating Candidate Retrieval for Candidate Ranking

In this section, we account for the important motivation to use alias-based approach for candidate ranking. We clarity how the performance of candidate retrieval affects the accuracy of candidate ranking. Table 5.1 shows the accuracy of different candidate retrieval. For candidate ranking, we use all features. Table 5.1 indicates that alias-based approach plus person name extension achieved the best accuracy of candidate ranking. That's the reason we use it in the whole thesis.

Table 5.1: Performance on InKB mentions with different candidate retrieval methods .

Methods	**Accuracy (InKB)**
exact matching on titles	64.33%
cosine (Threshold = 0.9)	64.56%
cosine (Threshold = 0.8)	65.32%
cosine (Threshold = 0.7)	77.01%
cosine (Threshold = 0.6)	70.19%
cosine (Threshold = 0.5)	70.30%
alias dictionary	82.54%
alias dictionary + person name extension	85.64%

5.5 Evaluating Embedding Features for Candidate Ranking

5.5.1 Experiments and Results on Japanese Wikification Corpus

We conducted incremental tests for the features by using 5-fold cross validations. We measured the performance for InKB mentions so that we can exclude the effects of the simple rules for judging NIL mentions. Beginning with the feature setting of linguistic features, we added embedding features incrementally, and reported their impact on the top-1 accuracy of InKB mentions.

From the results of Table 5.2, We find both the features of word vectors (WV) and entity vectors (EV) further improved the performance。There is no change after adding the embedding feature of paragraph vectors (PV). Here, features of entity vectors (EV) is more effective than features of word vectors (WV). Further, the best performance of our system reached to 86.79% after adding paragraph embedding features.

Table 5.2: Performance on InKB mentions by incremental feature study of embedding features.

Feature sets	**Accuracy (Top-1)**
S+P+Bw+Be+Entity Class (Class)	85.54%
S+P+Bw+Be+Class+Word Vectors (WV)	85.58%
S+P+Bw+Be+Class+Entity Vectors (EV)	86.22%
S+P+Bw+Be+Class+WV+EV	85.79%
S+P+Bw+Be+Class+EV+Paragraph Vectors (PV)	86.68%

5.6 Overall Evaluation of Proposed Supervised Entity Linking System with Embedding Features

5.6.1 Performance on Japanese Wikification Corpus

We performed a 5-fold cross validation, and calculated the average of accuracy values of the folds. We compared the accuracy of NIL mentions and InKB mentions with a unsupervised baseline method [30]. We evaluate how correctly the system could determine NIL for NIL mentions (the accuracy of NIL mentions) and link correct entities for InKB mentions (the accuracy of InKB mentions). The baseline method relies on the popularity of entities in the anchor texts of the mention, which is similar to the *Entity Popularity* feature. They also estimate probability distributions conditioned on a mention and its fine-grained semantic classes. Table 5.3 shows that the supervised method in this paper greatly improved the accuracy of InKB mentions. As a whole, the proposed system achieved an accuracy of 81.60% across the 5-folds, outperforming the previous method by a significant margin.

5.6.2 Error Analysis

There are three type of errors for our candidate ranking approach: the system linked an InKB mention to an incorrect entity (44.48%); the system determined an InKB mention as a NIL mention (12.90%); the system determined a NIL mention as an InKB mention and assigned a

Table 5.3: Comparing the system performance of the proposed method with a unsupervised method.

Methods	**Acc (InKB mentions)**	**Acc (NIL mentions)**	**Acc (All)**
Our system	85.87%	69.38%	81.60%
Popularity[30]	39.75%	92.23%	53.31%
Popularity+Class [30]	39.68%	92.23%	53.26%

reference entity (42.62%). Since we use simple rules to determine NIL mentions, we expect to improve NIL determining rules to solve 55.52% unsuccessful instances of NIL mentions.

We analyzed the 44.48% failure instances of InKB mentions in details. Table 5.4 lists some of the unsuccessful instances for the InKB mentions. Because we used a supervised method for candidate ranking, we cannot identify the exact cause of an error, which has various features intertwined to compute the score.

Table 5.4: Unsuccessful instances of candidate ranking.

Mention Examples	**Examples of Gold Entity**	**Examples of System Output**
米(United States of America)	アメリカ合衆国(United States of America)	米(Rice)
スズキ(Japanese sea bass/-Suzuki)	スズキ(魚) (Japanese sea bass)	スズキ(企業) (Suzuki Motor Corporation)
日本(Japan)	日本放送局(Japan Television Network Corporation)	日本(Japan)
ピオリア(Peoria)	ピオリア(アリゾナ州) (Peoria, Arizona)	ピオリア(イリノイ州) (Peoria, Illinois)
ヒルマン(Hillman)	トレイ・ヒルマン(Trey, Hillman)	エリック・ヒルマン(Eric, Hillman)

We found that the surface matching provides a strong bias for some incorrect instances. For example, the system maps the mention “米(United States of America)” with “米(Rice)”

incorrectly because they have the same surface character but the gold entity “アメリカ合衆国” (United States of America) does not share any character with the mention. Calculating the string similarity between a mention and each alias of a candidate entity and utilizing the maximum of similarity values may solve this problem, because the alias “米(United States of America)” exists in the alias list of the gold entity “アメリカ合衆国(United States of America)”.

The popularity feature also has a strong preference to major entities. For example, we found some cases where “スズキ” was mapped to an incorrect entity “スズキ(企業)” that are linked from the anchor “スズキ” more than “スズキ(Japanese sea bass)” in Wikipedia, even if the input document describes fish. The bias of popularity is a common, ongoing problem mentioned in the previous work [37].

Other error categories are caused by failing to disambiguate candidate entities that have the same entity class. For example, the mention “ピオリア(Peoria)” was linked to “ピオリア(イリノイ州)(Peoria, Illinois)” instead of the correct entity “ピオリア(アリゾナ州) (Peoria, Arizona)”, although the strong hint word “アリゾナ州(Arizona)” appeared in the context. To correct these errors, we plan to incorporate features that capture overlap between the surface of candidate entity and words in the context.

Some incorrect instances were due to the high ambiguity of the incorrect system outputs and the correct entities. For example, our system linked the mention “ヒルマン(Hillman)” to the incorrect entity “エリック・ヒルマン(Eric, Hillman)” instead of the correct entity “トレイ・ヒルマン(Trey, Hillman)”, which is difficult to disambiguate by our features because both are baseball players.We expect that utilizing coherence between candidate entity and co-occurring entities is useful to disambiguate such instances. For example, if the entity “北海道日本ハムファイターズ(Hokkaido Nippon-Ham Fighters)” exists in the context, it can help to link to the correct entity “トレイ・ヒルマン(Trey, Hillman)” because they have the relation “Coaching career”. We plan to incorporate these features in our future system to improve the performance of InKB mentions.

5.7 Summary

In this chapter, we describe three embedding models for constructing low-dimensional vectors for the context of mentions and description text of entities: word, entity and paragraph vectors. We first illustrate the preprocessing and procedures of learning embedding models on a large-scale unstructured corpus, Wikipedia. What is more, we verity the effectiveness of embedding features on Japanese Wikification corpus.

CHAPTER 6

Conclusions

6.1 Conclusions

In this thesis, we investigated and developed two key components of EL, candidate retrieval and candidate ranking. We build a searchable alias dictionary to generate referent Wikipedia articles for the given mentions. Comparing with the methods based on string similarity, the alias dictionary extracted from Wikipedia was verified more effective on generating candidate lists with high-recall and short length.

Moreover, we integrated several advanced linguistics feature sets and comprehensively studied their effectiveness o on English EL. In addition, we applied linguistics feature sets on Japanese EL and verified they are effective for Japanese EL as well.

Furthermore, we jointly learned a new entity representation model and improved the system performance by adding features based on the learned entity embeddings. We verified that word vectors and paragraph vectors also effectively improve the system performance. All in all, our system overcome the previous work on Japanese corpus with significant margins.

6.2 Future Perspective

We consider and plan to improve our system in the following aspects:

- **NIL Determination** Since we use a simple heuristic method to determine non-NIL and NIL mentions, the accuracy significantly drops after NIL determination process. In the future work, we will explore effect features for determining NIL entities and improve the NIL determination method by using supervised approaches.

- **Candidate Retrieval** In future work, we plan to use the technology of cross-lingual information retrieval to solve the transliteration problems between Japanese and English. We also consider developing methods for matching abbreviations between Japanese mentions and Wikipedia articles. We plan to incorporate some additional features that are mentioned in Section 3.6.2.

- **Candidate Ranking** The candidate ranking method may be improved by leveraging advanced methods, such as Convolutional Neural networks (CNN) and Long Short Term Memory (LTSM) networks, instead of simply using the average of vectors. Moreover, we plan to combine linguistic features with link-based methods to further improve our system.

- **Mention Detection** Finally, we will plan to incorporate a mention detection component with the current system in order to provide an end-to-end Japanese entity linking system.

List of Figures

List of Tables

Bibliography

[1] Ayman Alhelbawy and Robert Gaizauskas. Collective named entity disambiguation using graph ranking and clique partitioning approaches. In *Proceedings of COLING*, pages 1544–1555, 2014.

[2] Ayman Alhelbawy and Robert Gaizauskas. Graph ranking for collective named entity disambiguation. pages 75–80, 2014.

[3] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics, 1998.

[4] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 179–188. ACM, 2015.

[5] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16, 2006.

[6] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16, 2006.

[7] James Clarke, Yuval Merhav Ghalib Suleiman, and Shuai Zheng David Murgatroyd. Basis technology at tac 2012 entity linking. In *Proceedings of TAC 2012*, 2012.

[8] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[9] Laura Dietz and Jeffrey Dalton. A cross document neighborhood expansion: Umass at tac kbp 2012 entity linking. In *Proceedings of Text Analysis Conference (TAC)*, 2012.

[10] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285, 2010.

[11] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Proceedings of TACL*, 2:477–490, 2014.

[12] Paolo Ferragina and Ugo Scaiella. 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*, pages 1625–1628. ACM, 2010.

[13] Tatsuya Furakawa, Takeshi Sagara, and Akiko Aizawa. Semantic disambiguation for cross-lingual entity linking (in japanese). *Journal of Japan society of Information and Knowledge*, 24(2):172–177, 2014.

[14] Norberto Fernández Garcıa, Jesús Arias Fisteus, and Luis Sánchez Fernández. Comparative evaluation of link-based approaches for candidate ranking in link-to-wikipedia systems. *Journal of Artificial Intelligence Research*, 49:733–773, 2014.

[15] Chung H Gooi and James Allan. Cross-document coreference on a large scale corpus. Technical report, DTIC Document, 2004.

[16] David Graus, Tom Kenter, Marc Bron, Edgar Meij, M Rijke, et al. Context-based entity linking-university of amsterdam at tac 2012. 2012.

[17] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING*, volume 96, pages 466–471, 1996.

[18] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *HLT-NAACL*, pages 1020–1030, 2013.

[19] Yuhang Guo, Guohua Tang, Wanxiang Che, Ting Liu, and Sheng Li. Hit approaches to entity linking at tac 2011. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*. Citeseer, 2011.

[20] Zhaochen Guo and Denilson Barbosa. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 499–508. ACM, 2014.

[21] Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke S Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of EMNLP*, pages 289–299, 2013.

[22] Hui Han, Hongyuan Zha, and C Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, pages 334–343. IEEE, 2005.

[23] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM, 2011.

[24] Xianpei Han and Jun Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 215–224. ACM, 2009.

[25] Taichi Hashimoto, Takashi Inui, and Koji Murakami. Constructing extended named entity annotated corpora (in japanese). *In IPSJ Natural Language Processing (2008-NL-188)*, pages 113–120, 2008.

[26] Yoshihiko Hayashi, Kenji Yamakuchi, Masaaki Nagata, and Takaaki Tanaka. Improving wikification of bitexts by completing cross-lingual information (in japanese). In *Proceedings of The 28th Annual Conference of the Japanese Society for Artificial Intelligence*, pages 1A2–2, 2014.

[27] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of ACL*, pages 30–34, 2013.

[28] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, pages 782–792, 2011.

[29] Tatsukuni Inoue, Keigo Suenaga, Nagata Seiya, and Kenji Tateishi. Tagging geopolitical information on news article by using entity linking (in japanese). In *Proceedings of the Twenty-second Annual Meeting of the Association for Natural Language Processing*, 2016.

[30] Davaajav Jargalsaikhan, Naoaki Okazaki, Koji Matsuda, and Kentaro Inui. Building a corpus for japanese wikificaiton with fine-grained entity classes. In *ACL student research workshop. to appear*, 2016.

[31] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Proceedings of Third Text Analysis Conference (TAC 2010)*, volume 3, pages 3–3, 2010.

[32] Heng Ji, Joel Nothman, and Ben Hachey. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proceedings of Text Analysis Conference (TAC2014)*, 2014.

[33] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[34] Mahboob Alam Khalid, Valentin Jijkoun, and Maarten De Rijke. The impact of named entity normalization on information retrieval for question answering. In *Proceedings of Advances in Information Retrieval*, pages 705–710. Springer, 2008.

[35] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[36] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[37] Xiao Ling, Sameer Singh, and Daniel S Weld. Design challenges for entity linking. *Proceedings of TACL*, 3:315–328, 2015.

[38] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedings of AAAI*, 2012.

[39] Paul McNamee and Hoa Trang Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113, 2009.

[40] Paul McNamee, Mark Dredze, Adam Gerber, Nikesh Garera, Tim Finin, James Mayfield, Christine Piatko, Delip Rao, David Yarowsky, and Markus Dreyer. Hltcoe approaches to knowledge base population at tac 2009. In *Proceedings of Text Analysis Conference (TAC)*, 2009.

[41] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[42] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[43] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in neural information processing systems*, pages 3111–3119, 2013.

[44] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[45] Tatsuya Nakamura, Masumi Shirakawa, Takahiro Hara, and Shojiro Nishio. An entity linking method for closs-lingual topic extraction from social media (in japanese). In *Proceedings of DEIM Forum 2015*, pages A3–1, 2015.

[46] Naoaki Okazaki and Jun'ichi Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 851–859, Beijing, China, August 2010.

[47] Seiya Osada, Keigo Suenaga, Yoshizumi Shogo, Kazumasa Shoji, Tsuneharu Yoshida, and Yasuaki Hashimoto. Assigning geographical point information for document via entity linking (in japanese). In *Proceedings of the Twenty-first Annual Meeting of the Association for Natural Language Processing*, pages A4–4, 2015.

[48] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL*, pages 1375–1384. Association for Computational Linguistics, 2011.

[49] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC: Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[50] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *Proceedings of LREC*, 2002.

[51] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 793–803. Association for Computational Linguistics, 2011.

[52] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012-015*, 2012.

[53] Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of LREC*, pages 3168–3175, 2012.

[54] Fabian Suchanek and Gerhard Weikum. Knowledge harvesting from text and web sources. In *Proceedings of Data Engineering (ICDE)*, pages 1250–1253. IEEE, 2013.

[55] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of IJCAI*, pages 1333–1339, 2015.

[56] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. Multi-label classification of wikipedia articles into fine-grained named entity types (in japanese). In *Proceedings of the Twenty-second Annual Meeting of the Association for Natural Language Processing*, 2016.

[57] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[58] Yorick Wilks and Mark Stevenson. The grammar of sense: Is word-sense tagging much more than part-of-speech tagging? *arXiv preprint cmp-lg/9607028*, 1996.

[59] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity linking with effective acronym expansion, instance selection, and topic modeling. In *Proceedings of IJCAI*, volume 2011, pages 1909–1914, 2011.

[60] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Proceedings of NAACL*, pages 483–491. Association for Computational Linguistics, 2010.