Journal of Information Processing Vol.25 341–350 (Apr. 2017)

[DOI: 10.2197/ipsjjip.25.341]

Regular Paper

Supervised Approaches for Japanese Wikification

Shuangshuang Zhou[1,a)] Naoaki Okazaki[1,b)] Koji Matsuda[1,c)] Ran Tian[1,d)] Kentaro Inui[1,e)]

Received: September 8, 2016, Accepted: January 5, 2017

Abstract: Wikification is the task of connecting mentions in texts to entities in a large-scale knowledge base, Wikipedia. In this paper, we present a pipeline system for Japanese Wikification that consists of two components, namely candidate generation and candidate ranking. We investigate several techniques for each component, using a recently developed Japanese Wikification corpus. For candidate generation, we find that a name dictionary using anchor texts of Wikipedia is more effective than other methods based on similarity of surface forms. For candidate ranking, we verify that a set of features used in English Wikification is effective in Japanese Wikification as well. In addition, by using a corpus that links mentions to Japanese Wikipedia entries instead of to English Wikipedia entries, we are able to acquire rich contextual information from Japanese Wikipedia articles, which leads to improvements for Japanese mention disambiguation. We take this advantage by exploring several embedding models that encode context information of Wikipedia entities. The experimental results demonstrate that they improve candidate ranking. We also report the effect of each feature in detail. To sum, our system achieves 81.60% accuracy, significantly outperforming the previous work.

Keywords: named entity disambiguation, entity linking, Wikification, SVM

1. Introduction

Wikification [30], also known as a specialization of Entity Linking (EL) or named entity disambiguation, is the task of linking mentions in texts to entities in a large-scale knowledge base, Wikipedia [*1]. Wikification is useful in many Natural Language Processing (NLP) tasks including information retrieval [1], question answering [24], searching digital libraries [13], coreference resolution [8], [12], named entity recognition [8] and knowledge base population [7], [41].

A typical Wikification system first performs Named Entity Recognition to detect and classify spans of texts that mention certain types of entities. Then, the system links the mentions to entries in Wikipedia. A major challenge here is the ambiguity of mention; for example, given the sentence "*The I.B.M. is the world's largest organization dedicated to the art of magic,*" a Wikification system should associate "*I.B.M*" with the organization "*International Brotherhood of Magicians*", rather than the American technology and consulting company. An orthodox approach to address this issue is a pipeline of two components, **candidate generation** that generates a candidate list of possible entities for each mention, and **candidate ranking** that ranks candidates based on multiple features (**Fig. 1**). For candidate generation, another challenge is the variety of mentions. For example, both "*Big Blue*" and "*I.B.M.*" could refer to "*International Business Machines Corporation*".

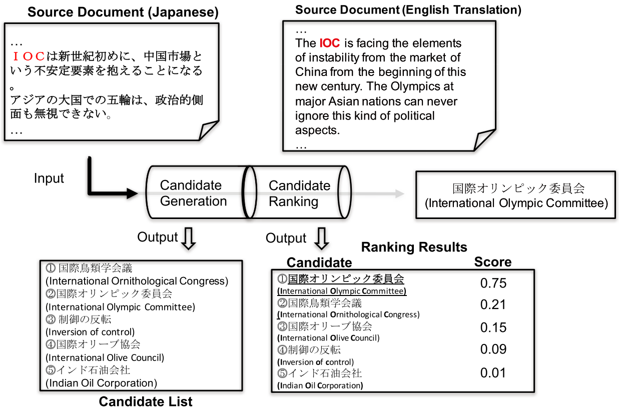

Source Document (Japanese)
…
ＩＯＣは新世紀初めに、中国市場という不安定要素を抱えることになる。
アジアの大国での五輪は、政治的側面も無視できない。
…
Source Document (English Translation)
…
The IOC is facing the elements of instability from the market of China from the beginning of this new century. The Olympics at major Asian nations can never ignore this kind of political aspects.
…
Input
Candidate Generation
Candidate Ranking
国際オリンピック委員会
(International Olympic Committee)
Output
Output
① 国際鳥類学会議
(International Ornithological Congress)
②国際オリンピック委員会
(International Olympic Committee)
③ 制御の反転
(Inversion of control)
④国際オリーブ協会
(International Olive Council)
⑤インド石油会社
(Indian Oil Corporation)
Candidate List
Ranking Results
Candidate
Score
①国際オリンピック委員会
(International Olympic Committee)
0.75
②国際鳥類学会議
(International Ornithological Congress)
0.21
③国際オリーブ協会
(International Olive Council)
0.15
④制御の反転
(Inversion of control)
0.09
⑤インド石油会社
(Indian Oil Corporation)
0.01

Fig. 1 A Wikification system generates and ranks a list of candidate entities for the mention "*IOC*".

English Wikification has been widely investigated [3], [4], [5], [33], [37], [38]. Ling et al. [26] has achieved high accuracy. Comparing with the great progress in English Wikification, development of Japanese Wikification is slow. Notwithstanding, developing Japanese Wikification has many merits. First, it contributes to populating Japanese knowledge base (e.g., DBpedia) by extracting relations of entities from texts. Moreover, it is also beneficial for populating English knowledge base by using cross-lingual information extraction. For example, TAC KBP [21] has proposed a tri-lingual EL shared task for English knowledge population. In addition, it is useful for improving text understanding on Japanese texts.

The previous research on Japanese Wikification mostly links mentions to English Wikipedia [10], [15], [34]. That might be impractical because about 44.4% (440 k out of 991 k articles) of

[1] Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi 980–8579, Japan
[a)] shuang@ecei.tohoku.ac.jp
[b)] okazaki@ecei.tohoku.ac.jp
[c)] matsuda@ecei.tohoku.ac.jp
[d)] tianran@ecei.tohoku.ac.jp
[e)] inui@ecei.tohoku.ac.jp

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

Japanese Wikipedia articles do not have corresponding English Wikipedia articles [18]. The slow development of Japanese Wikification is partly due to the lack of a publicly available Japanese Wikification corpus. Recently, Jargalsaikhan et al. [18] 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.

In this study, we aim to build a Japanese Wikification system that makes use of the context of mentions in candidate ranking. In addition, we explore candidate generation methods that can provide a high-coverage list of entity candidates without increasing the number of unrelated entities. The contributions of this paper are in three aspects.

(1) For candidate generation, most methods rely on an implementation of full-text search (e.g., Lucene Solr [*2] or Wikipedia search) or CrossWikis [40] for retrieving entity candidates. Since the highly dependence of the target language (e.g., spelling variations, tokenization) and corpus (e.g., newswire or Twitter), candidate generation for Japanese Wikification deserves addressing. Thus, we for the first time address the issues regarding strategies for candidate generation in Japanese Wikfication.

In this work, we explore an advanced method for a string matching by using SimString, but we find the recall of this method unsatisfactory; instead, an alias dictionary turns out to be effective in finding correct candidates. Therefore, we present a method for building an alias dictionary that realizes the idea of Crosswikis [40] by extracting alias from Japanese Wikipedia resources (e.g., Wikipedia anchor texts, redirect pages).

(2) For candidate ranking, we followed and investigated the features that the state-of-the-art systems for English employed [4], [5], [7], [46]. We verified their effectiveness for Japanese Wikification as well. Moreover, we present a novel method for learning embeddings of words and entities jointly with skip-gram model on Japanese Wikipedia. We apply several embedding models to encode context information of entities in Wikipedia articles, including word embedding, entity embedding, paragraph embedding. The experiments show that the embeddings are useful for disambiguating mentions in texts. In addition, those proposed embedding features for Japanese Wikification also provide possibility to improve English Wikification.

(3) Overall, our system achieves 81.60% accuracy, outperforming the previous work significantly. It is the first paper where various research topics and issues regarding Japanese Wikification (e.g., evaluating the effectiveness of each feature, error analyses for candidate generation and ranking) are discussed.

In addition, distinguishing with the previous English Wikification systems [9], [33], our proposed system restricts to named entities excluding general concepts. Because it is high subjective and confusing to determine the referent entities for general concepts [18], [27].

2. Related Work

English Wikification, or more broadly English Entity Linking (EL), is a widely studied topic. There are several public corpora for English Wikification [5], [45]. TAC-KBP is a well known workshop for promoting the research on entity linking [19], [20], [21], [22], [28].

To address the variety of mentions, the previous work on English EL generated entity candidates from the results of a search engine [7], [11], [48]. In these previous studies, surfaces of entities in Wikipedia are indexed into a certain search engine (e.g., Lucene Solr). Candidate entities are generated by retrieving search engine with the mention. Another common approach utilizes various resources such as Wikipedia disambiguation, Wikipedia redirect pages, GeoNames, etc. [7], [40], [48]. Our work constructs an alias dictionary following the previous work [40].

To address the ambiguity of mentions, the previous work has explored linguistic features, string similarity between entity surfaces and mention surfaces [2], [7], [48], cosine similarity of bag-of-words [2], [7], [11], [48], entity type information [7], [46], [48], topic model [7], [48], etc. When building a Japanese Wikification system, we explore several effective linguistic features, following the previous work on English Wikification.

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 [1], [42]. Embedding models provide useful representations of linguistic units such as words [31], entities [42], paragraphs [25], etc. This dense and low-dimensional representation is useful to compute the semantic similarities. For example, Blanco et al. [1] propose an EL method for web queries by representing entities and mentions with the averages of their respective vectors. He et al. [16] encode the representations of the input document containing the mention as well as the Wikipedia article by Stacked Denoising Auto-encoders [44]. Sun et al. [42] disambiguate mentions by computing the vector similarity between the two continuous vector of mentions and candidate entities. In this previous work, a candidate entity is represented with a combination of the average of vectors of surface words and the average of vectors of category words of the entity. Mention and the context of mentions are represented with the average 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. [42], we learn a new representation of entities by leveraging their context in the Wikipedia articles. In additional, we use entity class features instead of entity category features.

The research on Japanese Wikification has received less attention. These previous studies are not comparable with the ones for English Wikification. First, the domain of the previous work on Japanese is limited. For example, Furukawa et al. [10] focused on

[*2] http://lucene.apache.org/solr/

linking mentions in academic fields, and link technical terms to English Wikipedia. Some studies only focus on linking geopolitical entities in local news articles [17], [36] as well. Second, a few studies address Wikification in generic domains, but they use the English Wikipedia as a target from Japanese mentions [15], [34]. This setting has two problems: translating Japanese mentions into English and the insufficient coverage of English Wikipedia for Japanese entities.

In this study, we link Japanese mentions in general domains directly to Japanese Wikipedia by using refined methods for candidate generation and multiple linguistic and embedding features for candidate ranking.

3. System Architecture

In this section, we present our pipeline system for Japanese Wikification. Given results of Named Entity Recognition (NER) as input, the system links the named entity mentions to Wikipedia articles as output.

Our system consists of two standard components: candidate generation and candidate ranking (Fig. 1). In candidate generation, our system generates a list of Wikipedia articles that can be referred to by each mention in text. For example, given a mention “*IOC*”, the candidates include Wikipedia articles titled “国際鳥類学会議 (*International Ornithological Congress*)”, “国際オリンピック委員会 (*International Olympic Committee*)”, etc. In candidate ranking, the system computes a ranking score for each Wikipedia article in the candidate list via a scoring function based on a supervised learning method. For example, in Fig. 1, “国際オリンピック委員会 (*International Olympic Committee*)” is identified as the reference of “*IOC*”. We describe the details of the two components in this section.

3.1 Candidate Generation

If the candidate generation in a Wikification system cannot include correct Wikipedia articles in a candidate list for a mention, the subsequent process (candidate ranking) cannot recover from the error. For this reason, the previous English Wikification systems tend to over-generate candidate entities. String similarity between the entity name (Wikipedia title) and the surface form of the mention is a common method for generating candidate entities. In addition, it is effective to consider mention aliases by using resources from Wikipedia, for example, disambiguation pages, redirect pages, anchor texts, etc. In this work, we compare a conventional string similarity method with an alias dictionary extracted from resources in Wikipedia.

For string similarity method, we use a simple and efficient tool, `SimString` [35]. 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. We use the cosine similarity in this study. Extract all Japanese Wikipedia titles, we index them as a `SimString` database.

For building an alias dictionary, we base our approach on the previous English Wikification work [40]. 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*)”. The candidate generation looks up the alias dictionary to reach an entity from a mention. We apply exact matching on alias because fuzzy matching on alias texts may slightly improve the recall at the expense of increasing noisy candidates.

In addition, we find some correct entities cannot be reached only by anchor texts. 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 appear in the target document.

We recognize family names, and full names of people based on the results of a morphological analyzer, MeCab [*3]. 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.

Therefore, we extend mentions of person names before retrieving on the alias dictionary. In Section 5.3, we demonstrate the effect of this treatment for improving the coverage of candidate generation.

3.2 Candidate Ranking

We formulate the candidate ranking problem by referring the previous studies [2], [29]. Namely, we construct a scoring function $f(m, e)$ based on features extracted from the mention m and the candidate Wikipedia article e. We select the candidate $\hat{e}$ with the highest score from a candidate list E_m, according to the score,

$$\hat{e} = \arg\max_{e \in E_m} f(m, e).$$

Therefore, the scoring function $f(m, e)$ should be trained such that the correct Wikipedia article $\hat{e}$ is linked to the mention m. We use SVM$^{\text{rank}}$ [23] with linear kernel for training the scoring function.

In addition to obtaining the top-1 Wikipedia article $\hat{e}$, we need to determine whether the mention in the text is NIL mention (not in Wikipedia) or InKB mention (in Wikipedia). We employ two rules in order to recognize a mention as NIL mention: when the list of candidate entities for the mention is empty; and when the value of $f(m, \hat{e})$ is below a threshold (heuristically set to 2.9).

3.2.1 Feature Sets

In this section, we describe the features for constructing the scoring function. These are powerful features used by the state-of-the-art English Wikification systems, and several new embedding features. **Table 1** shows a complete list. As a running ex-

[*3] http://taku910.github.io/mecab/

Table 1 Features for candidate ranking.

Feature Type	Description	Example
String Similarity (S)	string similarity between mention and entity title	the Levenshtein edit-distance between "*IOC*" and "*International Olympic Committee*" is 11
Entity Popularity (P)	distribution of anchor texts in Wikipedia	68% of mention "*IOC*" in Japanese Wikipedia is linked to article "*International Olympic Committee*"
Bag-of-Word (Bw)	BoW similarity between text and Wikipedia article	words {"*face*", "*market*", ...} from text and {"*modern*", "*Olympic*", ...} from Wikipedia article
Bag-of-Entity (Be)	BoE similarity between text and Wikipedia article	entities {"*China*", "*Olympic*", ...} in text and {"*Olympic Games*", ...} in Wikipedia article
Average of Word Vector (WV)	cosine similarity between the average vectors of word vectors	cosine similarity between the average vector of $\mathbf{w}_{face}, \mathbf{w}_{market}, \ldots$ for text of the mention and the average vector of $\mathbf{w}_{modern}, \mathbf{w}_{Olympic}, \ldots$ for the Wikipedia article
Average of Entity Vector (EV)	cosine similarity between the average vectors of entity vectors	cosine similarity between the average vector of $\mathbf{e}_{China}, \mathbf{e}_{Olympic}, \ldots$ for text of the mention and the average vector of $\mathbf{e}_{Olympic\_Games}, \ldots$ for the Wikipedia article
Paragraph Vector (PV)	cosine similarity between paragraph vectors	cosine similarity between paragraph vector for text and paragraph vector for Wikipedia article
Entity Category (Cate)	word in text is category of Wikipedia article	Wikipedia article "*International Olympic Committee*" belongs to categories "*Olympic movement*", "*Committees*"
Entity Class (Class)	overlap of Sekine's entity class	mention "*IOC*" in text is labeled *Sports Organization Other* and Wikipedia entry "*International Olympic Committee*" is labeled *International Organization*

ample, we consider the following text snippet (translated from Japanese) around a mention "*IOC*" (m):

> **IOC** は新世紀初めに、中国 市場という不安定要素を抱えることになる。アジアの大国での 五輪 は、政治的側面も無視できない。
>
> 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 article "国際オリンピック委員会 (*International Olympic Committee*)" (e):

> 国際オリンピック委員会は、近代オリンピック を主催する団体であり、また オリンピック に参加する各種国際スポーツ統括団体を統括する組織である。2009年に国際連合総会オブザーバー資格を得たため国際機関の一つと思われがちだが、非政府組織 (NGO) の非営利団体 (NPO) である。
>
> 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.

String Similarity This type of features measures the string similarity between mentions and the titles of Wikipedia articles. We use several similarity measures explored in the previous work [6], [11], including of Levenshtein distance and Jaccard coefficient.

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

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

As discussed in Ref. [33], this probability reflects the 'popularity' of a Wikipedia article.

Bag-of-Word Similarity 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. We consider two similarity measures including of cosine similarity of TF-IDF weights [47] and Jaccard coefficient [6].

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.

Embedding Similarity This feature group measures the similarity between texts and Wikipedia contents based on learned embeddings instead of bag-of-words or bag-of-entities. We construct vectors for texts and Wikipedia articles, and assess cosine similarity between the vectors. We consider three types of vectors, namely the word vector (WV), entity vector (EV), and paragraph vector (PV). We describe the details of the embedding models in Section 4.

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.

Entity Class Unlike 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 [26], [27] than using a coarse-grained entity class. The corpus [18] in this work has annotated each named entity with a fine-grained entity class label, called Sekine's entity class [39]. Suzuki et al. [43] automatically label Wikipedia articles with Sekine's entity class based on a multi-label classification method. Here, each mention is allowed to have more than one semantic 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 [18]. In this case, this feature does not fire for the Wikipedia article "*International Olympic Committee*".

4. Embedding Models

In this section, we describe the embedding models for constructing low-dimensional vectors for the context of mentions and Wikipedia articles.

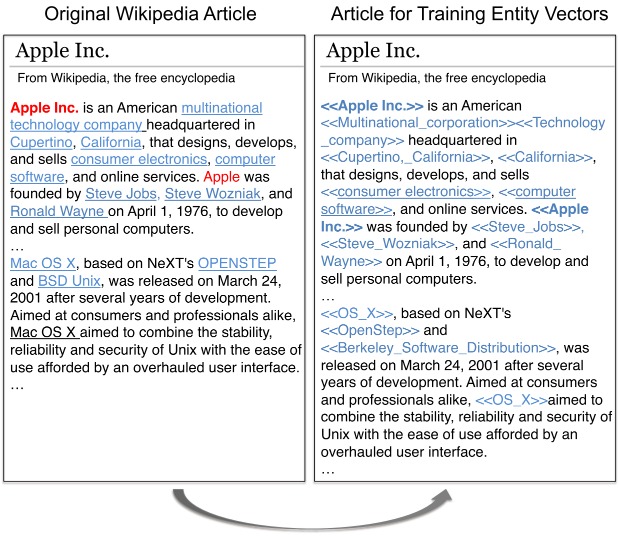
4.1 Word Vector

For this model, we apply the `word2vec`[*4] tool to the entire text of Japanese Wikipedia for training word vectors. The vectors of a mention context and a Wikipedia article are defined by the averages of the vectors in the mention context and the Wikipedia article, respectively.

4.2 Entity Vector

We also tested embedding features of Wikipedia articles (entities) proposed by Ref. [43]. In this method, distributed representation of the articles are learned by Skip-gram model [32] with the "one-sense-per-discourse" assumption by replacing anchor texts with their corresponding article titles.

For example, in **Fig. 2**, we replace all anchor texts with strings referred to by the anchor texts (e.g., replacing the hyper-link "*Mac OS X*" to ⟪*OS_X*⟫ that represents the Wikipedia article "*OS X*"), and train vectors for Wikipedia articles as well as words. In this way, we can learn embeddings for entities and words in the same space. The vector of a mention context is defined by the average vector of the vectors in the mention context. To experiment, we used 200-dimensional embedding provided by [43] trained with Wikipedia dump[*5].



Original Wikipedia Article

Apple Inc.

From Wikipedia, the free encyclopedia

Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services. Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne on April 1, 1976, to develop and sell personal computers.
…
Mac OS X, based on NeXT's OPENSTEP and BSD Unix, was released on March 24, 2001 after several years of development. Aimed at consumers and professionals alike, Mac OS X aimed to combine the stability, reliability and security of Unix with the ease of use afforded by an overhauled user interface.
…

Article for Training Entity Vectors

Apple Inc.

From Wikipedia, the free encyclopedia

<<Apple Inc.>> is an American <<Multinational_corporation>><<Technology_company>> headquartered in <<Cupertino,_California>>, <<California>>, that designs, develops, and sells <<consumer electronics>>, <<computer software>>, and online services. **<<Apple Inc.>>** was founded by <<Steve_Jobs>>, <<Steve_Wozniak>>, and <<Ronald_Wayne>> on April 1, 1976, to develop and sell personal computers.
…
<<OS_X>>, based on NeXT's <<OpenStep>> and <<Berkeley_Software_Distribution>>, was released on March 24, 2001 after several years of development. Aimed at consumers and professionals alike, <<OS_X>>aimed to combine the stability, reliability and security of Unix with the ease of use afforded by an overhauled user interface.
…

Fig. 2 Training entity vectors from anchor texts.

4.3 Paragraph Vector

The paragraph vector [25] is a simple and powerful method of learning representations of arbitrary lengths of texts. We use the Distributed Memory Model of Paragraph Vectors model to train paragraph vectors of Wikipedia articles. The model is an extension of the CBoW model [31] implemented in `word2vec`. We regard a Wikipedia article as a 'paragraph' and train vectors for Wikipedia articles.

5. Experiments

In this section, we first explain the training and evaluation sets. Then we evaluate the performance of candidate generation and candidate ranking. Finally, we compare the proposed system with the previous work [18].

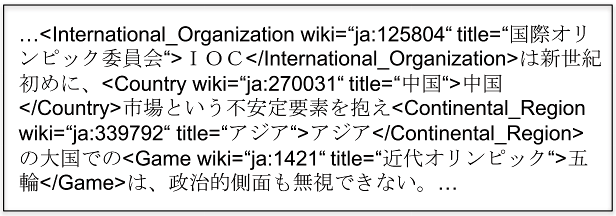
5.1 Data Set

We use a Japanese Wikification corpus [18] that consists of 340 newspaper articles from Balanced Corpus of Contemporary Written Japanese (BCCWJ)[*6]. Mentions in each document are annotated with fine-grained named entity classes that are defined by Sekine's Extended Named Entity Hierarchy [39][*7]. 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 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".

5.2 Experimental Setup

We utilize a Japanese Wikipedia dump[*8] as the reference knowledge base. We tokenize and remove punctuations in documents by using Mecab, a Japanese morphological analyzer. We train word embeddings, entity embeddings, and paragraph vec-



…<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。…

Fig. 3 A snapshot of annotation document with the document ID of 'PN1a_00008'.

[*4] https://code.google.com/p/word2vec/

[*5] http://www.cl.ecei.tohoku.ac.jp/˜m-suzuki/jawiki_vector/

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

[*7] https://sites.google.com/site/extendednamedentityhierarchy/

[*8] We used a Wikipedia dump as of March 5, 2016.

tors on this pre-processed corpus. The word and entity vectors were learned by setting the dimensions d to 200, the size of context window c to 10, and the number of negative samples to 5. Following the setting in Ref. [25], the paragraph vectors were learned by setting the dimensions d to 400, the size of context window c to 5, and the number of negative samples to 5.

5.3 Evaluation of Candidate Generation

We evaluate the methods of candidate generation on the dataset [18]. In candidate generation, we aim to confirm that the correct entity is retrieved (Recall). Here, recall is the percentage of mentions that can be correctly linked to the gold entities. We also compute the average number of candidates (AveNumCan), 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 generation, the ambiguity and the processing time may be added to candidate ranking. We normalize mention surfaces to eliminate differences between half-width characters and full-width characters in advance.

For string similarity method, we first compare cosine similarity with thresholds between 0.5 and 0.9. **Table 2** shows the recall and average number of candidates with different thresholds of 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 dictionary 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 2 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, 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 dictionary with the name expansion step throughout this paper.

Table 2 Performance of candidate generation 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 (+extended)	94.14%	17.79

5.4 Overall Performance

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 [18]. 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 3** shows that the supervised method in this paper greatly improved the accuracy of InKB mentions. Here, we utilize the set of features with the best performance, which is showed in **Table 4**. Since, the processing of NIL determination may incorrectly determine an InKB mention as a NIL, there is different accuracy of InKB mentions between Table 3 and Table 4. As a whole, the proposed system achieved an accuracy of 81.60% across the 5-folds, outperforming the previous method by a significant margin.

5.5 Feature Study

We conducted ablation 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 string similarity feature set, we added various features incrementally, and reported their impact on the top-1, top-2 and top-5 accuracy of InKB mentions.

Table 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 [18]	39.75%	92.23%	53.31%
Popularity+Class [18]	39.68%	92.23%	53.26%

Table 4 Performance on InKB mentions by incremental feature study.

Feature sets	Accuracy (Top-1)	Accuracy (Top-2)	Accuracy (Top-5)
StringSim (S)	57.64%	65.71%	86.44%
S+Popularity (P)	62.10%	67.24%	92.53%
S+P+Bag-of-words (Bw)	84.03%	88.47%	93.14%
S+P+Bag-of-entities (Be)	76.33%	81.01%	92.86%
S+P+Bw+Be	84.25%	88.68%	93.29%
S+P+Bw+Be+Word Vectors (WV)	84.46%	88.80%	93.26%
S+P+Bw+Be+Entity Vectors (EV)	84.85%	89.10%	93.43%
S+P+Bw+Be+Paragraph Vectors (PV)	84.25%	88.68%	93.29%
S+P+Bw+Be+WV+EV+PV	84.87%	89.11%	93.42%
S+P+Bw+Be+WV+EV+PV+Entity Class (Class) **(Best)**	**85.84%**	**89.93%**	**93.52%**
S+P+Bw+Be+WV+EV+PV+Entity Category (Cate)	84.95%	89.17%	93.38%
S+P+Bw+Be+WV+EV+PV+Class+Cate	85.73%	89.76%	93.46%

Table 5 Different types of error examples in candidate generation.

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)

Table 6 Unsuccessful instances of candidate ranking.

Mention Examples	Examples of Gold Entity	Examples of System Output
米 (United States of America) スズキ (Japanese sea bass/Suzuki) 日本 (Japan)	アメリカ合衆国 (United States of America) スズキ (魚) (Japanese sea bass) 日本放送局 (Japan Television Network Corporation)	米 (Rice) スズキ (企業) (Suzuki Motor 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)

From the results of Table 4, we found that the rates of increase of top-2 and top-5 accuracy are consistent with the rate of top-1 accuracy. The high accuracy at top-5 shows that correct entities have been placed in the top results by the proposed candidate ranking model. Since the accuracy at top-1 is more significant, we focused on the results of top-1 accuracy in the following discussions.

We found that our system obtained the performance of approximately 18% higher than the previous work only by using string similarity features. Adding popularity features slightly improved the performance. We observe a significant improvement when adding bag-of-words features. Only adding bag-of-entities features led the performance drop of about 8%. However, adding both the bag-of-words and bag-of-entities features together, the system performance was better than using only the bag-of-words feature.

In addition, 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) (about a 0.4% increase in accuracy). We found that the accuracy was improved to 84.87% (about a 0.5% increase) after adding all of the features of WV, EV and PV.

The best accuracy (85.84%) was obtained after adding the features of entity class. Adding the features of fine-grained entity class is better than adding the category features. Therefore, we completely remove the category feature from the system.

6. Error Analysis

6.1 Error Analysis of Candidate Generation

Table 5 summarizes error types of the presented Wikification system. 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)", 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 [14]. 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 generation. However, we leave these treatments as a future work, which may increase the number of false entities in candidate generation.

6.2 Error Analysis of Candidate Ranking

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 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 6** 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.

We found that matching on the surface forms 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 [26].

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 forms of candidate entities 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.

7. Conclusions and Future Work

In this paper, we constructed a pipeline system for Japanese Wikification that consists of two standard components, candidate generation and candidate ranking. We built a name dictionary in order to retrieve Wikipedia articles for Japanese mentions. The name dictionary extracted from Wikipedia was more effective for generating candidate entities than the methods based on similarity of surface forms.

Moreover, we verified the effectiveness of the features used in English Wikification on a Japanese Wikification corpus. We jointly learned embeddings for words and entities in the same space, and improved the performance by adding features based on the learned entity embeddings. We demonstrated that word embeddings and paragraph vectors also improve the system performance effectively. Overall, our system outperformed the baseline system on the same data set with a significant margin.

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 6.1. The candidate ranking method can be improved by leveraging advanced methods, such as Convolutional Neural networks (CNN) and Long Short Term Memory (LTSM) networks, instead of using the average of vectors. Finally, we will plan to incorporate a mention detection component with the current system in order to provide an end-to-end Japanese Wikification system.

Acknowledgments This work was partially supported by JSPS KAKENHI Grant Number 15H01702 and 15H05318, and JST, CREST.

References

[1] Blanco, R., Ottaviano, G. and Meij, E.: Fast and space-efficient entity linking for queries, *Proc. Eighth ACM International Conference on Web Search and Data Mining*, pp.179–188, ACM (2015).

[2] Bunescu, R.C. and Pasca, M.: Using Encyclopedic Knowledge for Named entity Disambiguation, *Proc. EACL*, Vol.6, pp.9–16 (2006).

[3] Cai, Z., Zhao, K., Zhu, K.Q. and Wang, H.: Wikification via link co-occurrence, *Proc. 22nd ACM international conference on Conference on information & knowledge management*, pp.1087–1096, ACM (2013).

[4] Cheng, X. and Roth, D.: Relational inference for wikification, *Urbana*, Vol.51, p.61801 (2013).

[5] Cucerzan, S.: Large-Scale Named Entity Disambiguation Based on Wikipedia Data, *Proc. EMNLP-CoNLL*, Vol.7, pp.708–716 (2007).

[6] Dietz, L. and Dalton, J.: A cross document neighborhood expansion: UMass at TAC KBP 2012 entity linking, *Proc. Text Analysis Conference* (*TAC*) (2012).

[7] Dredze, M., McNamee, P., Rao, D., Gerber, A. and Finin, T.: Entity disambiguation for knowledge base population, *Proc. 23rd International Conference on Computational Linguistics*, pp.277–285 (2010).

[8] Durrett, G. and Klein, D.: A joint model for entity analysis: Coreference, typing, and linking, *Proc. TACL*, Vol.2, pp.477–490 (2014).

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

[10] Furakawa, T., Sagara, T. and Aizawa, A.: Semantic disambiguation for cross-lingual entity linking (in Japanese), *Journal of Japan Society of Information and Knowledge*, Vol.24, No.2, pp.172–177 (2014).

[11] Graus, D., Kenter, T., Bron, M., Meij, E., Rijke, M. et al.: Context-based entity linking-University of Amsterdam at TAC 2012 (2012).

[12] Hajishirzi, H., Zilles, L., Weld, D.S. and Zettlemoyer, L.S.: Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves, *Proc. EMNLP*, pp.289–299 (2013).

[13] Han, H., Zha, H. and Giles, C.L.: Name disambiguation in author citations using a k-way spectral clustering method, *Proc. 5th ACM/IEEE-CS Joint Conference on*, pp.334–343, IEEE (2005).

[14] Hashimoto, T., Inui, T. and Murakami, K.: Constructing extended named entity annotated corpora (in Japanese), *IPSJ Natural Language Processing* (*2008-NL-188*), pp.113–120 (2008).

[15] Hayashi, Y., Yamakuchi, K., Nagata, M. and Tanaka, T.: Improving Wikification of Bitexts by Completing Cross-lingual Information (in Japanese), *Proc. 28th Annual Conference of the Japanese Society for Artificial Intelligence*, pp.1A2–2 (2014).

[16] He, Z., Liu, S., Li, M., Zhou, M., Zhang, L. and Wang, H.: Learning Entity Representation for Entity Disambiguation, *Proc. ACL*, pp.30–34 (2013).

[17] Inoue, T., Suenaga, K., Seiya, N. and Tateishi, K.: Tagging geopolitical information on news article by using entity linking (in Japanese), *Proc. 22nd Annual Meeting of the Association for Natural Language Processing* (2016).

[18] Jargalsaikhan, D., Okazaki, N., Matsuda, K. and Inui, K.: Building a Corpus for Japanese Wikificaiton with Fine-Grained Entity Classes, *ACL student research workshop. to appear* (2016).

[19] Ji, H., Grishman, R. and Dang, H.T.: Overview of the TAC2011 Knowledge Base Population Track, *Proc. 3rd Text Analysis Conference* (*TAC 2011*) (2011).

[20] Ji, H., Grishman, R., Dang, H.T., Griffitt, K. and Ellis, J.: Overview of the TAC 2010 knowledge base population track, *Proc. 3rd Text Analysis Conference* (*TAC 2010*), Vol.3, No.2, pp.3–3 (2010).

[21] Ji, H., Nothman, J. and Hachey, B.: Overview of tac-kbp2014 entity discovery and linking tasks, *Proc. Text Analysis Conference* (*TAC2014*) (2014).

[22] Ji, H., Nothman, J., Hachey, B. and Florian, R.: Overview of TAC-

KBP2015 Tri-lingual Entity Discovery and Linking, *Proc. 3rd Text Analysis Conference* (*TAC 2015*) (2015).

[23] Joachims, T.: Training linear SVMs in linear time, *Proc. 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.217–226, ACM (2006).

[24] Khalid, M.A., Jijkoun, V. and De Rijke, M.: The impact of named entity normalization on information retrieval for question answering, *Proc. Advances in Information Retrieval*, pp.705–710, Springer (2008).

[25] Le, Q.V. and Mikolov, T.: Distributed representations of sentences and documents, arXiv preprint arXiv:1405.4053 (2014).

[26] Ling, X., Singh, S. and Weld, D.S.: Design challenges for entity linking, *Proc. TACL*, Vol.3, pp.315–328 (2015).

[27] Ling, X. and Weld, D.S.: Fine-Grained Entity Recognition, *Proc. AAAI* (2012).

[28] McNamee, P. and Dang, H.T.: Overview of the TAC 2009 knowledge base population track, *Text Analysis Conference* (*TAC*), Vol.17, pp.111–113 (2009).

[29] McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D. and Dreyer, M.: HLTCOE approaches to knowledge base population at TAC 2009, *Proc. Text Analysis Conference* (*TAC*) (2009).

[30] Mihalcea, R. and Csomai, A.: Wikify!: linking documents to encyclopedic knowledge, *Proc. 16th ACM Conference on information and knowledge management*, pp.233–242, ACM (2007).

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

[32] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Proc. Advances in neural information processing systems*, pp.3111–3119 (2013).

[33] Milne, D. and Witten, I.H.: Learning to link with wikipedia, *Proc. 17th ACM conference on Information and knowledge management*, pp.509–518, ACM (2008).

[34] Nakamura, T., Shirakawa, M., Hara, T. and Nishio, S.: An entity linking method for closs-lingual topic extraction from social media (in Japanese), *Proc. DEIM Forum 2015*, pp.A3–1 (2015).

[35] Okazaki, N. and Tsujii, J.: Simple and Efficient Algorithm for Approximate Dictionary Matching, *Proc. 23rd International Conference on Computational Linguistics* (*Coling 2010*), Beijing, China, pp.851–859 (2010), available from ⟨http://www.aclweb.org/anthology/C10-1096⟩.

[36] Osada, S., Suenaga, K., Shogo, Y., Shoji, K., Yoshida, T. and Hashimoto, Y.: Assigning geographical point information for document via entity linking (in Japanese), *Proc. 21st Annual Meeting of the Association for Natural Language Processing*, pp.A4–4 (2015).

[37] Ratinov, L., Roth, D., Downey, D. and Anderson, M.: Local and global algorithms for disambiguation to wikipedia, *Proc. ACL*, Association for Computational Linguistics, pp.1375–1384 (2011).

[38] Roth, D., Ji, H., Chang, M.-W. and Cassidy, T.: Wikification and Beyond: The Challenges of Entity and Concept Grounding, *ACL* (*Tutorial Abstracts*), p.7 (2014).

[39] Sekine, S., Sudo, K. and Nobata, C.: Extended Named Entity Hierarchy, *Proc. LREC* (2002).

[40] Spitkovsky, V.I. and Chang, A.X.: A Cross-Lingual Dictionary for English Wikipedia Concepts, *Proc. LREC*, pp.3168–3175 (2012).

[41] Suchanek, F. and Weikum, G.: Knowledge harvesting from text and Web sources, *Proc. Data Engineering* (*ICDE*), pp.1250–1253, IEEE (2013).

[42] Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z. and Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation, *Proc. IJCAI*, pp.1333–1339 (2015).

[43] Suzuki, M., Matsuda, K., Sekine, S., Okazaki, N. and Inui, K.: Multi-label classification of wikipedia articles into fine-grained named entity types (in Japanese), *Proc. 22nd Annual Meeting of the Association for Natural Language Processing* (2016).

[44] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders, *Proc. 25th international conference on Machine learning*, pp.1096–1103, ACM (2008).

[45] Yosef, M.A., Hoffart, J., Bordino, I., Spaniol, M. and Weikum, G.: Aida: An online tool for accurate disambiguation of named entities in text and tables, *Proc. VLDB Endowment*, Vol.4, No.12, pp.1450–1453 (2011).

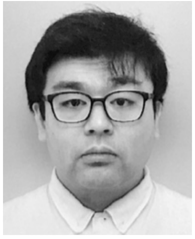
[46] Zhang, W., Sim, Y.C., Su, J. and Tan, C.L.: Entity Linking with Effective Acronym Expansion, Instance Selection, and Topic Modeling, *Proc. IJCAI*, Vol.2011, pp.1909–1914 (2011).

[47] Zheng, Z., Li, F., Huang, M. and Zhu, X.: Learning to link entities with knowledge base, *Proc. NAACL*, Association for Computational Linguistics, pp.483–491 (2010).

[48] Zhou, S., Kruengkrai, C., Okazaki, N. and Inui, K.: Exploring Linguistic Features for Named Entity Disambiguation, *International Journal of Computational Linguistics and Applications*, Vol.5, No.2, p.49 (2014).

Shuangshuang Zhou received her bachelor's degree in engineering from Northeastern University, China in 2010, and her M.S. degree in engineering from Northeastern University, China in 2012. She has been a Ph.D. student of Graduate School of Information Sciences, Tohoku University since 2013. Her current research interest is natural language processing, especially entity linking and knowledge acquisition.

Naoaki Okazaki is an associate professor at Graduate School of Information Sciences, Tohoku University. Prior to his faculty position, he worked as a research fellow in National Centre for Text Mining (NaCTeM) (in 2005) and as a postdoctoral researcher in University of Tokyo (in 2007–2011). He obtained his Ph.D. from Graduate School of Information Science and Technology, University of Tokyo in 2007. He has served as a technical consultant in SmartNews Inc. since 2013. He is also a visiting research scholar of the Artificial Intelligence Research Center (AIRC), AIST. His research interests include natural language processing, text mining, and machine learning.



Koji Matsuda received his bachelor's degree in engineering from Toyohashi Institute of Technology in 2006, and his M.E. degree in engineering from Tokyo Institute of technology in 2012. He has been a researcher of Graduate School of Information Sciences, Tohoku University since 2014. His current research interest is natural language processing, especially entity linking and automatic knowledge acquisition.

Ran Tian received his Ph.D. of mathematical science from the University of Tokyo in 2012. Since then, he became a project researcher at the National Institute of Informatics. He studied algebraic geometry during his doctoral course, and is currently working on natural language processing, especially semantic processing and logical inference. From 2014, he became a research assistant professor at Tohoku University.

Kentaro Inui received his doctorate degree of engineering from Tokyo Institute of Technology in 1995. He has experience as an assistant professor at Tokyo Institute of Technology and an associate professor at Kyushu Institute of Technology and Nara Institute of Science and Technology, he has been a professor of Graduate School of Information Sciences at Tohoku University since 2010. His research interests include natural language understanding and knowledge processing. He currently serves as the IPSJ director and ANLP director.