Creating an Extended Named Entity Dictionary from Wikipedia

Ryuichiro Higashinaka   Kugatsu Sadamitsu
Kuniko Saito   Toshiro Makino   Yoshihiro Matsuo
NTT Media Intelligence Laboratories, NTT Corporation
1-1 Hikarinooka Yokosuka-Shi Kanagawa 239-0847 Japan
`{higashinaka.ryuichiro, sadamistu.kugatsu, saito.kuniko, makino.toshiro, matsuo.yoshihiro}@lab.ntt.co.jp`

ABSTRACT
Automatic methods to create entity dictionaries or gazetteers have used only a small number of entity types (18 at maximum), which could pose a limitation for fine-grained information extraction. This paper aims to create a dictionary of 200 extended named entity (ENE) types. Using Wikipedia as a basic resource, we classify Wikipedia titles into ENE types to create an ENE dictionary. In our method, we derive a large number of features for Wikipedia titles and train a multiclass classifier by supervised learning. We devise an extensive list of features for the accurate classification into the ENE types, such as those related to the surface string of a title, the content of the article, and the meta data provided with Wikipedia. By experiments, we successfully show that it is possible to classify Wikipedia titles into ENE types with 79.63% accuracy. We applied our classifier to all Wikipedia titles and, by discarding low-confidence classification results, created an ENE dictionary of over one million entities covering 182 ENE types with an estimated accuracy of 89.48%. This is the first large scale ENE dictionary.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE (JAPANESE)

Wikipediaを用いた拡張固有表現辞書の構築

従来の固有表現辞書では，少ない数（最大で18）の固有表現タイプが用いられてきたため，ピンポイントな情報抽出に適用することが難しいという問題があった．そこで，本稿では，200の拡張固有表現タイプを用いた固有表現辞書の構築を目指す．具体的には，教師あり学習による多クラス分類器を用い，Wikipediaの見出し語を拡張固有表現タイプに分類することで辞書を構築する．特徴量として，見出し語そのもの，本文，そして，カテゴリ等のメタデータに関するものを数多く列挙し用いた．結果として，見出し語を，79.63%の精度で，拡張固有表現タイプに分類できることが分かった．学習された多クラス分類器を，Wikipediaのすべての見出し語に適用し，また，信頼度の低い分類結果については除外するようにしたところ，推定分類精度が89.48%で，また，182の拡張固有表現タイプをカバーする，百万以上のエントリを持つ拡張固有表現辞書を構築することができた．この辞書は，初の大規模な拡張固有表現辞書である．

KEYWORDS: Extended Named Entity, Dictionary, Wikipedia.

KEYWORDS IN JAPANESE: 拡張固有表現，辞書，Wikipedia.

Proceedings of COLING 2012: Technical Papers, pages 1163–1178,
COLING 2012, Mumbai, December 2012.

1 Introduction

For information extraction, it is important to recognize named entities (NEs) in texts. NEs are typically recognized by such techniques as support vector machines (SVMs) (Isozaki and Kazawa, 2002) and conditional random fields (Suzuki et al., 2006), using words surrounding a target entity as cues to determine if that entity belongs to a certain NE type. The limitation is that it is difficult to recognize NEs when there are few contextual cues, such as in search queries and snippets of web search results. In such cases, an NE dictionary, or a gazetteer, is particularly useful. Here, an NE dictionary means a list of entities associated with their NE types (e.g., Tokyo → LOCATION, Barack Obama → PERSON). Such a dictionary is also useful for deriving gazetteer features for training an NE recognizer (Kazama and Torisawa, 2008).

A number of studies have focused on automatically creating NE dictionaries; e.g., (Toral and Muoz, 2006; Saleh et al., 2010). Such studies generally use a small number of entity types, mostly adopting those defined in the Message Understanding Conference (MUC) (Grishman and Sundheim, 1996) or Information Retrieval and Extraction Exercise (IREX) (Sekine and Isahara, 2000). To enable more fine-grained information extraction, some attempted to cover more NE types: (Chang et al., 2009), (Watanabe et al., 2007), and (Tkatchenko et al., 2011), using up to 18 NE types. Still, we consider the current granularity of the NE types used to be too coarse, especially for tasks such as question answering (Voorhees and Dang, 2005), where systems need to pinpoint exact entities requested by users.

This paper proposes to create a dictionary of **extended NEs (ENEs)**. An ENE hierarchy was proposed by Sekine et al. (2002); Sekine and Nobata (2004), and it defines three levels of NE types. At the leaf level, it has 200 ENE types. We aim to create an ENE dictionary that covers all these 200 types. In our approach, using Wikipedia as a basic resource, we classify Wikipedia titles into ENE types to create an ENE dictionary. We perform supervised learning; we derive a large number of features for Wikipedia titles and train a multiclass classifier. The features encode various aspects of Wikipedia titles, including those related to the surface string of a title, the content of an article, and the meta data, such as categories and infoboxes. We devise an extensive list of features for accurate classification into a large number of ENE types. The idea of using Wikipedia for creating an NE dictionary is not new; however, no work has sought to use such a large number of NE types. We want to verify whether it is possible to create an NE dictionary covering such a large number of NE types. We also want to know what types of features are useful in classifying entities into fine-grained ENE types. Note here that this work uses Japanese Wikipedia. Although we want to make our features as language-independent as possible, we introduce some possibly language-specific features for better accuracy.

In the next section, we describe related work. Section 3 describes our proposed method and shows the complete list of features used for training a classifier. Section 4 describes our experiments to verify our proposed method, the effect of the size of training data, and how to refine the acquired dictionary by using probability estimates of the learned classifier. Section 5 summarizes and mentions future work.

2 Related Work

The work to create NE dictionaries has centered around Wikipedia, with the focus on classifying Wikipedia titles (articles) into NE types. As far as we know, the earliest work is by Toral and Muoz (2006). They focused on the first sentence of a Wikipedia article (generally a definition statement) and counted the number of nouns related to three NE types, namely, Location, Organization, and Person, and applied heuristic rules that take into account the numbers in order to determine the article's NE type. Bhole et al. (2007) followed and proposed a supervised machine learning approach to the same task, involving the training of an SVM classifier. The features used were the bag-of-words of Wikipedia articles.

In addition to the texts of articles, there are rich meta data in Wikipedia, which can be helpful in distinguishing the NE types of articles. In addition to the nouns of first sentences, Nothman et al. (2008) used head nouns of categories assigned to articles and used heuristic rules to map them to one of four NE types; namely, LOC, PER, ORG, and MISC. Dakka and Cucerzan (2008) trained an SVM classifier by using features related to the structure of Wikipedia articles. Specifically, they introduced bag-of-words features of abstracts (Wikipedia provides short abstracts for articles), tables (infoboxes and contents boxes), and links to other articles. As NE types, they used five: LOC, PER, ORG, MISC, and COMM (common object). Saleh et al. (2010) also used features derived from abstracts, infoboxes, and categories to train their SVM classifier. A similar feature set was also used by (Tardif et al., 2009) who worked on six NE types. Richman and Schone (2008) exploited the multilingual nature of Wikipedia. By using links to articles in other languages, they classified non-English articles into NE types by using their English counterparts, whose NE types can be estimated by using their category information. They used four NE types: DATE, GPE (geographical and political entity), ORG, and PERSON.

The above studies used a relatively small number of NE types, but there are also studies that aimed to cover a larger number of NE types. Chang et al. (2009) used nine NE types (person, act, communication, location, animal, artifact, time, object, and group), which are a subset of supersenses defined in WordNet. To cope with the larger number of NE types, they introduced new features, such as the supersenses of the head nouns of first sentences and the synsets of category names, for training their maximum entropy classifier. Watanabe et al. (2007) worked on 13 NE types taken selectively from the ENE hierarchy. They had to avoid using the full 200 ENE types because of a sparseness problem for their graph-based algorithm. Tkatchenko et al. (2011) used 18 NE types taken from the BBN's question answering taxonomy (Brunstein, 2002). The features used for their SVM and naive Bayes classifiers were conceptually identical to those in (Tardif et al., 2009).

Our work is similar to the previous literature in that we use Wikipedia to create an NE dictionary, but different in that we aim to deal with a much larger number of NE types: 200 ENE types. We want to verify the feasibility of creating such a fine-grained NE dictionary and want to explore useful features for the classification into ENE types. To this end, we make an extensive list of features, adopting those previously proposed and also proposing new ones, for training our classifier. We describe our features later in Section 3.2.

Although not directly related to creating an NE dictionary, there is a good body of work that aims at constructing an ontology (a hierarchy of words or concepts connected with relations such as *is-a* and *has-a*) from Wikipedia (Ponzetto and Strube, 2007; Suchanek et al., 2008; Nagata et al., 2010). Since ontologies are useful resources for the deep processing of texts, such as the inference, once we have created our ENE dictionary, our next step would include constructing an ontology by relating the ENEs.

3 Proposed Method

Following the previous studies, we also propose to use supervised learning and learn a classifier that classifies Wikipedia titles (articles) into ENEs. We have three steps: the creation of training data, extraction of features, and training of a classifier. Since we deal with many NE types (200 ENE types), we place a special emphasis on the extraction of features for accurately distinguishing the ENE types. In what follows, we describe each step in detail.

3.1 Creating training data

As our training data, we need Wikipedia titles tagged with ENE types. However, since there are many ENE types and we need a reasonable amount of training data for each ENE type to avoid data sparseness, the manual creation of training data will be very costly.

Therefore, to facilitate the creation of training data, our basic idea is to turn to an existing corpus annotated with ENE types. In Japanese, there is one such corpus publicly available (Hashimoto et al., 2008). The corpus contains newswire articles in which entities are annotated with ENE tags. From such a corpus, we can extract entities together with their ENE tags to create what we call a **seed dictionary**, whose entries can be matched against Wikipedia titles to take an intersection so that Wikipedia titles can be automatically annotated with their ENE types. Of course, when there are other corpora or gazetteers available, they can also be exploited to augment the seed dictionary. The approach we employ here is similar to (Bhole et al., 2007) and (Zhang and Iria, 2009) in that entries of external dictionaries/gazetteers are intersected with Wikipedia titles to create training data. The difference is that we use an annotated corpus to create such entries. Below, we enumerate the steps we performed to create our seed dictionary and training data.

1. From the Hashimoto corpus (Hashimoto et al., 2008), we extracted all tagged sections. There are 8828 newswire articles (Mainichi Shimbun newspaper '95) in the corpus with 255407 tagged sections. Since some tags are not related to ENE types, we first discarded such tagged sections. We also discarded entities that were annotated with multiple ENE tags because such entities can be ambiguous and would introduce noise. For example, the entity "Rakuten" is a company but also a sports organization owned by the same company; hence it is given Company in one context and Pro_Sports_Organization in the other. By retaining only the unambiguous entities, we obtained 59318 unique entities with their ENE tags.
2. We performed the same procedure as above on our in-house corpus, which we have been maintaining and consists of newswire articles and blogs annotated with ENE types. There are 7184 documents with 118051 ENE tagged sections, from which we extracted 40231 unique entities with their ENE tags.
3. We maintain gazetteers of ENEs for evaluating our ENE recognizer. From the gazetteers, we first removed ambiguous entities (some entities are listed in multiple gazetteers) and then extracted 35858 unique entities with their ENE tags.
4. We merged the results of steps 1–3 to create a seed dictionary. After removing overlaps and ambiguous entities (NB. different tags can be given to the same entity depending on the source of the entity), we created a seed dictionary that contains 128213 unique entities with their ENE tags.
5. The entries of the seed dictionary were matched against Wikipedia titles to take an intersection. We used the Japanese Wikipedia dump of 2012-08-06, which contains 1411994 titles (we use this dump of Wikipedia throughout the paper). The titles here contain all entries including redirects, categories, and disambiguation pages. We found 51576 Wikipedia titles whose surface forms exactly matched the entries in the seed dictionary. We then removed, matched titles that consisted only of a single character, bare numbers, or two or less Hiragana/Katakana (Japanese phonologic characters) because they are potentially vague; their ENE tags could have been incidentally unique due to the lack of instances in the corpus or gazetteer; e.g., bare numbers can be of any ENE type concern-

Rank	ENE type	Count	Ratio	Accum
1	Person	9456	18.93%	18.93%
2	City	1897	3.80%	22.73%
3	Position_Vocation	1827	3.66%	26.38%
4	Product_Other	1805	3.61%	30.00%
5	Company	1725	3.45%	33.45%
6	Doctrine_Method_Other	1380	2.76%	36.21%
7	Date	1070	2.14%	38.35%
8	Dish	786	1.57%	39.93%
9	Book	736	1.47%	41.40%
10	Character	723	1.45%	42.85%
11	Broadcast_Program	618	1.24%	44.08%
12	School	601	1.20%	45.29%
13	Food_Other	600	1.20%	46.49%
14	Movie	594	1.19%	47.68%
15	GOE_Other	556	1.11%	48.79%
16	Show_Organization	554	1.11%	49.90%
17	Music	542	1.08%	50.98%
18	Corporation_Other	537	1.07%	52.06%
19	Station	458	0.92%	52.98%
20	Game	454	0.91%	53.89%

Table 1: Top 20 ENE types in our training data.

ing a number, such as Date, Age, or N_Person. This process left us with 49956 Wikipedia titles (3.54% of all titles), and this becomes our training data.

Our training data (49956 titles) cover 191 ENE types; unfortunately, we could not cover all 200 ENE types, because some ENE types, especially those related to numeric expressions, were scarcely seen in Wikipedia titles. The ones that could not be covered were Address_Other, Weight, Email, URL, Calorie, Intensity, Postal_Address, Seismic_Magnitude, and Volume. Table 1 shows the top-20 ENE types in the training data. We can see that, although Person is by far the most frequent, the decreasing pace of the frequencies of the subsequent types is slow. See Fig. 1 for the relationship between the ranks of ENE types and their counts. We can observe that, for a large proportion of ENE types, we have a reasonable number of training data, with a median around 100.

3.2 Feature Extraction

For each Wikipedia title in the training data, we extract features. We created an extensive list of features to cope with the many ENE types, covering many of the previously proposed features and also introducing new ones that we thought would be useful for distinguishing fine-grained types. Some features can be specific to Japanese. We propose to use 22 kinds of features in all. They are divided into three categories: surface string of a title, content of an article, and meta data of Wikipedia. It should be noted here that when a title has a redirect to another title, we extract features for both titles and merge them to create its features.

3.2.1 Features related to the surface string of the title of an article

The surface string of a title could be greatly indicative of its ENE type. For example, a title that ends with “shi (city)”, is likely to be the name of a city and therefore should be given

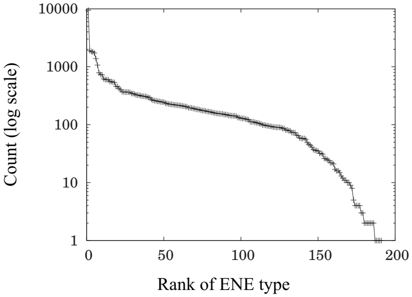

Figure 1: Relationship between the ranks of ENE types and their counts in the training data.

City as its ENE type. Rivers and Mountains, especially in Japanese, have names that end with "kawa (river)" or "yama (mountain)"; therefore, such names can be easily classified into River and Mountain ENE types. We have 16 features (T1–T16: 'T' stands for title) regarding various aspects of the surface string of a title. As far as we know, conventional studies have never put an emphasis on the surface string of a title, probably because most previous work worked with the English language. One exception is (Tardif et al., 2009) that used bag-of-unigrams in the title, corresponding to our T1. T2 through T16 are our newly introduced features. We enumerate the features below.

(T1-T2) Word unigram/bigram We first run the title in question through a morphological analyzer and separate it into words. Note that there are no marked word boundaries in Japanese. Then, we extract word unigrams and bigrams as features. Here, the features are bag-of-words features, indicating the existence of particular unigrams or bigrams with a binary value (i.e., 1 or 0). As a morphological analyzer, we use JTAG (Fuchi and Takagi, 1998).

(T3-T4) Character unigram/bigram We split a title into character tokens and then create bag-of-words features of character unigrams and bigrams. We have this feature since characters, especially Kanji (Chinese-origin) characters in Japanese, have individual meanings even when they form part of a word.

(T5-T6) POS unigram/bigram Using the results of the morphological analysis, we create bag-of-words features for the unigrams and bigrams of part-of-speech (POS) tags. Japanese POS taggers, including JTAG, generally output POS tags that correspond to subcategories for proper nouns, which can be useful in distinguishing ENE types. Other POS information, such as the existence of numbers, could also be useful.

(T7) Last common noun or noun/counter suffix We create a feature from the last word of a title whose POS tag is either a common noun or noun/counter suffix. This is a binary feature, indicating the existence of such a word. The rationale for using the last word of a title is that it is usually the head in Japanese. We chose to use common nouns and noun/counter suffixes because they denote general conceptions or entity categories in Japanese, and are hence more suitable than other POS tags. Here, the idea of using common nouns is similar to using the plural form of nouns in English (Suchanek et al., 2008).

(T8) Semantic category of the last common noun or noun/counter suffix JTAG outputs semantic categories for words. Here the semantic categories are those defined in the Japanese Goi-Taikei ontology (Ikehara et al., 1997). There are 2715 semantic categories in all. We extract features that indicate the existence of semantic categories for the last common noun or noun/counter suffix. It is natural to use these categories because they can be directly indicative of certain ENE types. The semantic categories can also provide abstract meanings for some words, which can be useful when there is sparseness in training data.

(T9-T11) Last one/two/three character(s) We have three features that use the last characters of a title because some characters alone could indicate certain ENE types in Japanese, especially when they are found at the end of a word. For example, words "ninja" and "geisha" both end with the Kanji character "sha", which by itself indicates a person. We use the existence of the last characters (one to three characters) of a title as features. We decided to use up to three characters to increase the coverage of subwords.

(T12) Semantic categories We extract semantic categories for all words in the title by using JTAG and create bag-of-words features. We use this feature to compensate for any possible lack of information that arises from using only the semantic categories of the last common noun or noun/counter suffix.

(T13) Proper noun semantic categories JTAG outputs special semantic categories for proper nouns. They are defined in the Goi-Taikei ontology and there are 130 such categories. Since proper nouns are conceptually similar to NEs, these categories will be useful for the classification of ENE types. We extract proper noun semantic categories for all words in a title and create their bag-of-words features.

(T14) IREX-based NEs We run an off-the-shelf NE recognizer, NameLister (Saito and Nagata, 2003; Suzuki et al., 2006), and extract IREX-based NEs in the title, from which we create bag-of-words features, indicating the existence of each of eight NE types in IREX. Here, the granularity of NEs is coarse; we regard this feature as a complement to other semantic category related features.

(T15) Last character type Japanese has several characters types, and certain types can be indicative of certain NE types. For example, Katakana characters are likely to be used for entities of a foreign origin such as cars and products, whereas Hiragana characters are likely to be used for Japanese entities. Here, we distinguish five types of characters: Hiragana, Katakana, Kanji, Alphabet, and Other, and use the type of the last character of a title as a feature.

(T16) Character type construction In addition to the last character type, this feature focuses on how character types constitute a title. We first split the title into character tokens and converted them into character types. Then, we concatenated the character types as a single string that represents the title's character type construction. For example, "London" is written with four consecutive Katakana characters in Japanese. Therefore, we have "K-K-K-K" (K stands for Katakana) as a binary feature.

3.2.2 Features related to the content of an article

The content of the article of a title obviously has important information about the entity. In this paper, we use two features (C17–C18, where 'C' stands for content) about the content of an article. We do not use all the words of an article because it would make the feature space too sparse and could cause over-fitting to the training data. We focus on the representative parts of an article; namely, the first sentence and headings (section titles). These features have been used in previous studies (Toral and Muoz, 2006; Dakka and Cucerzan, 2008; Tkatchenko et al., 2011).

(C17) Last common noun or noun/counter suffix of the first sentence It is widely known that the first sentence of an article in Wikipedia is a definition statement. To obtain the first sentence of an article, we first obtain the abstract text of the article from the abstract data provided with a Wikipedia dump and then select its first sentence by selecting a text span from the beginning to the first punctuation mark. Here, we use the abstract data to facilitate our extraction of the initial part of an article. Then, we analyze the sentence with JTAG to extract the last common noun or noun/counter suffix in the first sentence. Finally, we create a binary feature indicating the existence of the last common noun or noun/counter suffix.

(C18) Headings Headings or section titles summarize what is written in an article. For example, the article of "Nobunaga Oda", a famous warlord in Japan, has section titles such as life, personality, portrait, and policies, which clearly indicate that this article is about a person. Therefore, we create bag-of-words features of section titles, each of which indicates the existence of a particular section title. Here, we only use top-level section titles. We ignore such section titles as "links to other articles" and "references" because they can be found in arbitrary articles and therefore would not be useful for distinguishing NE types. We extract section titles from the XML dump of Wikipedia by locating texts enclosed by "==" (section title markers in the Wiki format).

3.2.3 Features related to the meta data of an article

Ever since the work of Nothman et al. (2008), meta data in Wikipedia have been vigorously used for the classification of articles into NE types and have proved their usefulness. In this paper, we extract four features (M19–M22: 'M' stands for meta data) regarding the meta data of an article. Here, M20 and M22 are our newly introduced features.

(M19) Direct categories Categories are one of the most widely used features for distinguishing NE types. We analyze each category assigned to an article with JTAG and extract the last common noun or noun/counter suffix to create a bag-of-words feature. This is similar to using head nouns of categories (Nothman et al., 2008).

(M20) Upper categories In Wikipedia, categories have a network (mostly hierarchical) structure where articles are the sub-nodes of the categories. Starting from the article in question, we find the shortest path to the root category ("Main Category") and use the categories on the path. Such categories can be regarded as the upper categories for the article, and can be useful for distinguishing NE types especially when the direct categories are too specific. For each category on the path (except for the root), we extract the last common noun or noun/counter suffix to create a bag-of-words feature. We use Wik-IE (Mori et al., 2009) to create the network structure, which calculates the distance between nodes in Wikipedia using the Dijkstra's algorithm.

(M21) Infobox attributes Infoboxes provide tabular data for articles. Since the attributes (attribute names) of a table generally indicate the attributes of the entity in question, in a similar manner to (Saleh et al., 2010), we use the infobox attributes to create our features. For each attribute in the infobox, we create a bag-of-words feature indicating the existence of that attribute. We used the Japanese version of DBPedia (http://ja.dbpedia.org/), which offers the infobox data for Japanese articles as triples.

(M22) Instance types Inspired by Richman and Schone (2008), who used the English version of Wikipedia for non-English articles, we also turn to the English version for useful information. We noticed that the English version is heavily linked with the English version of DBPedia (http://en.wikipedia.org/wiki/DBpedia), which offers instance types for English articles. The instance types are given in the vocabulary of various ontologies, such as the DBPedia ontology, schema.org, and Friend of a Friend (FOAF), and are likely to be helpful in distinguishing NE types. For example, “Paris” has Place, PopulatedPlace, Settlement, and Thing as instance types. To create this feature, we first search for the English version of an article by using the external language links and find its instance types by looking up DBPedia. Then, for each instance type, we create a bag-of-words feature. Note that Japanese DBPedia currently does not provide instance type data.

3.3 Training a classifier

For each title in the training data, we extract the features and train a multiclass classifier that classifies the title into one of the ENE types. Here, we employ logistic regression as a learning algorithm. We first create binary logistic regression classifiers for all ENE types. Then, in classifying a title, we find the classifier that outputs the best probability estimate for that title, and use the ENE type for that classifier as an output ENE type.

Although previous studies have extensively used SVMs for NE type classification, we chose logistic regression because of its capability to output probability estimates. Such estimates can be used as confidence scores and can be used to refine the classification results by discarding low-confidence entries with a threshold (See Section 4.4). Note that the scores output by SVMs cannot be directly used as confidence scores because they denote distances from hyperplanes whose absolute values cannot be compared with a fixed threshold. For training the classifiers and calculating probability estimates, we use the LIBLINEAR toolkit (http://www.csie.ntu.edu.tw/˜cjlin/liblinear/).

4 Experiments

We performed a series of experiments to verify our method. First of all, we created two sets of data by dividing our training data into two: one for training a classifier (TRAIN SET) and the other for testing (TEST SET). From the 49956 titles in our training data, we randomly sampled 10000 titles as TEST SET and made the rest TRAIN SET (39956 titles). TRAIN SET and TEST SET cover 190 and 179 ENE types, respectively.

We performed four experiments. The first experiment examined the classification accuracy when only one of the features is employed. The purpose of this experiment is to verify the effectiveness of each feature. Then, in the second experiment, we combined the features to maximize the classification accuracy. In the third experiment, we examined how the classification accuracy is affected by the size of training data. In the last experiment, we examined how the obtained ENE dictionary can be refined by discarding low-confidence entries. As an evaluation metric, we mainly use the classification accuracy, which is the rate of accurately classified ENE types in TEST SET. At the end of this section, we also describe our final ENE dictionary that we created by applying the trained classier to all titles in Wikipedia.

4.1 Results by individual features

Table 2 shows the classification accuracy when the classifiers were trained by using one of the features. Since some features can only be extracted for certain titles and since it is impossible

Feature		Acc	Rise in Acc
(T1)	Word unigram (BASE)	50.63%	0.00%
(M19)	Direct categories	72.54%	21.91%
(C17)	First sentence common noun	65.95%	15.32%
(T12)	Semantic categories	63.51%	12.88%
(M20)	Upper categories	63.40%	12.77%
(M21)	Infobox attributes	62.28%	11.65%
(C18)	Headings	62.08%	11.45%
(T4)	Character bigram	61.38%	10.75%
(M22)	Instance types	59.12%	8.49%
(T8)	Semantic category of last common noun	58.78%	8.15%
(T3)	Character unigram	58.39%	7.76%
(T9)	Last character	57.75%	7.12%
(T6)	POS bigram	56.99%	6.36%
(T5)	POS unigram	56.03%	5.40%
(T10)	Last two characters	55.37%	4.74%
(T13)	Proper noun semantic categories	54.06%	3.43%
(T14)	IREX NEs	53.37%	2.74%
(T11)	Last three characters	52.39%	1.76%
(T16)	Character type construction	51.94%	1.31%
(T15)	Last character type	51.59%	0.96%
(T2)	Word bigram	51.21%	0.58%
(T7)	Last common noun	50.94%	0.31%

Table 2: Classification accuracy for the individual features. All features are sorted by their performance except for the word unigram feature, which is used as a base feature.

to learn classifiers without a feature, we trained classifiers by using each of the features with an obligatory use of the word unigram as a base feature. As can be seen in the table, all features contribute to the improvement in the classification accuracy. The features related to the content and meta data, which proved their usefulness for a smaller number of NE types, also proved their effectiveness for ENE types. The direct category feature was by far the most effective. We have two title-related features, T12 and T4, that contributed to the rise of more than 10% over the base feature, suggesting the usefulness of using the title information for the classification into ENEs. We find it interesting that character bigrams contribute greatly to the classification accuracy, suggesting the importance of characters/subwords in Japanese.

4.2 Results by the combination of features

We use a backward selection method to find the best combination of features; that is, we first use all the features and then remove one feature at a time whose removal improves the performance the most. As a result, we removed seven features: (T3) Character unigram, (T4) Character bigram, (T5) POS unigram, (T10) Last two characters, (T13) Proper noun semantic categories, (T16) Character type construction, and (T7) Infobox attributes.

The feature selection improved the classification accuracy from 79.10% to 79.63%, the best accuracy we attained in this work. See Table 3 for the results. We believe the classification accuracy of 79.63% is reasonably high when considering the large number of ENE types. The table also shows the drop in performance when other features are individually eliminated from the best combination, which shows that even when we remove the most influential feature, i.e.,

Feature		Acc	Drop in Acc
	Best combination (w/o T3–T5, T10, T13, T16, M21)	**79.63%**	0.00%
(T11)	w/o Last three characters	79.50%	-0.13%
(T7)	w/o Last common noun	79.49%	-0.14%
(T14)	w/o IREX NEs	79.44%	-0.19%
(T8)	w/o Semantic category of last common noun	79.44%	-0.19%
(T2)	w/o Word bigram	79.38%	-0.25%
(T12)	w/o Semantic categories	79.35%	-0.28%
(T6)	w/o POS bigram	79.35%	-0.28%
(C18)	w/o Headings	79.33%	-0.30%
(T15)	w/o Last character type	79.31%	-0.32%
(M22)	w/o Instance types	79.29%	-0.34%
(T1)	w/o Word unigram	79.29%	-0.34%
(M20)	w/o Upper categories	79.23%	-0.40%
(T9)	w/o Last character	79.02%	-0.61%
(C17)	w/o First sentence common noun	78.93%	-0.70%
(M19)	w/o Direct categories	77.83%	-1.80%

Table 3: Classification accuracy for the best combination and when one of the features is removed from the best combination. The features are sorted by their performance.

Feature set	Acc
T (T1–T16)	69.37%
C (C17, C18)	68.57%
M (M19–M22)	75.40%
T+C	76.14%
T+M	78.64%
C+M	75.90%
T+C+M	**79.10%**
Previous work only (T1, C17, C18, M19, M21)	75.11%
Newly introduced only (T2–T16, M20, M22)	75.09%

Table 4: Classification accuracy for the combinations of feature sets. When the word unigram (T1) is not included in a feature set, we had it included as a base feature.

(M19) Direct categories, from the best combination, the drop is small (1.80%). This indicates that most of the features possess complementary information. The magnitude of a drop can be considered as the specific information carried by a certain feature. We consider it interesting that (T9) Last character has the third largest drop, indicating again the effectiveness of using characters in Japanese.

Table 4 shows the classification accuracy for some combinations of feature sets. We can see that, when we use the features of all three categories (T+C+M), the best accuracy is achieved. The contribution of the newly introduced features is clear, raising the accuracy from 75.11% to 79.10%.

Table 5 shows the precision, recall, and F-measure for the most frequent 20 ENE types in TEST SET (the feature set used is the best combination in Table 3). We can see that the performance varies greatly depending on the ENE types. For some ENE types, such as Product_Other, Doctrine_Method_Other, Character, and GOE_Other, our method performs very poorly. We suspect

Rank	ENE type	Precision		Recall		F
1	Person	0.851	(1840/2162)	0.952	(1840/1933)	0.899
2	City	0.859	(371/432)	0.907	(371/409)	0.882
3	Product_Other	0.525	(255/486)	0.646	(255/395)	0.579
4	Company	0.756	(301/398)	0.820	(301/367)	0.787
5	Position_Vocation	0.766	(301/393)	0.875	(301/344)	0.817
6	Doctrine_Method_Other	0.501	(177/353)	0.697	(177/254)	0.583
7	Date	0.887	(197/222)	0.956	(197/206)	0.921
8	Dish	0.809	(127/157)	0.789	(127/161)	0.799
9	Book	0.723	(102/141)	0.646	(102/158)	0.682
10	School	0.948	(127/134)	0.907	(127/140)	0.927
11	Character	0.639	(62/97)	0.449	(62/138)	0.528
12	Broadcast_Program	0.735	(83/113)	0.654	(83/127)	0.692
13	Food_Other	0.644	(76/118)	0.650	(76/117)	0.647
14	Show_Organization	0.876	(85/97)	0.746	(85/114)	0.806
15	GOE_Other	0.653	(49/75)	0.450	(49/109)	0.533
16	Movie	0.745	(70/94)	0.648	(70/108)	0.693
17	Music	0.772	(71/92)	0.676	(71/105)	0.721
18	Station	0.942	(81/86)	0.880	(81/92)	0.910
19	Corporation_Other	0.673	(72/107)	0.791	(72/91)	0.727
20	Game	0.892	(74/83)	0.831	(74/89)	0.860

Table 5: Precision, recall, and F-measure for the most frequent 20 ENE types in TEST SET.

this low performance is partly attributable to the variation of their names. We leave it to our future work to improve the accuracy for these hard-to-guess ENE types.

4.3 Learning curve

We split TRAIN SET into blocks of 1000 (1K) titles, and examined how the performance improves when each block is added to the training data. TEST SET was used for testing. Figure 2 shows the learning curve. We can see a steady improvement until we reach around 10K. The classification accuracy at 10K is 75.07%. However, after that, the pace of improvement is very gradual. From 30K on, the gradient is just 0.14% per K. Even were this gradient to continue, to reach 100%, we would need an additional 145K of training data, which would be too hard to create manually. For further improvement, it would definitely be necessary to devise new useful features. In addition, it would be helpful to bring in external gazetteers and link them with ENE types so that they can be used to augment our seed dictionary. The idea of linking with other gazetteers has been successful in the linked data community (http://linkeddata.org/), which makes this approach promising. We would like to pursue this approach in the future.

4.4 Refinement by using probability estimates

Our choice of logistic regression was motivated by its capability to output probability estimates so that low-confidence results can be discarded. We examined how this discarding process improves the accuracy. We set up a variable t for a cut-off threshold and moved it from 0 to 1 by increasing it by 0.05. When t becomes larger, erroneous (low-confidence) results would be discarded, and the classification accuracy would improve. However, the improvement in accuracy would come at the cost of decreased coverage (the number of retained samples over the total number of samples). Figure 3 shows how the accuracy and coverage change as

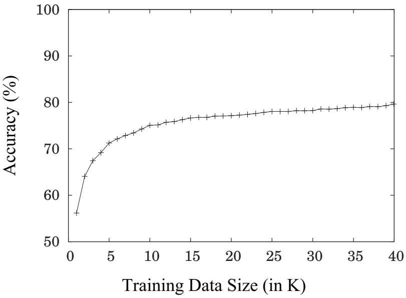

Figure 2: Learning Curve.

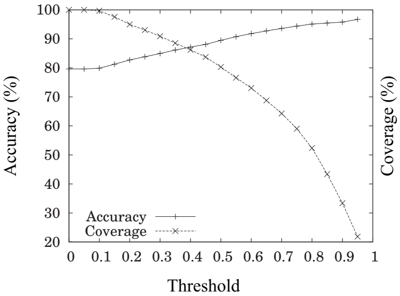

Figure 3: Accuracy and coverage by a varying threshold.

t increases. One can see that the improvement of accuracy is gradual between 80% to 95%, whereas that of the coverage is rather sharp. We also see a clear trade-off between the accuracy and coverage. We use this finding in the creation of our ENE dictionary in the next section.

4.5 Creating the final ENE dictionary

We apply our ENE classifier to all titles in Wikipedia in order to create our ENE dictionary. For this purpose, we newly trained an ENE type classifier using all of our training data (49956 titles). Then, we extracted the features (best combination in Table 3) for all Wikipedia titles (1411994 titles). Using the extracted features, we classified the titles into ENE types with their probability estimates. The result becomes our initial ENE dictionary.

As a dictionary resource, a weight should be given to accuracy over coverage. According to Fig. 3, when t is 0.5, about 90% accuracy (89.48%) can be achieved with over 80% coverage (80.29%). Since we viewed this as a fair setting, satisfying the requirements of a good dictionary, we decided to use 0.5 for t for refinement. After the refinement, 1015563 (71.92%) entries remained. Althouth the number of retrained entries were slightly smaller than the expected 80.29%, we could still retain over one million entries. Here, from Fig. 3, the estimated accuracy of the entries is 89.48%. The final dictionary covers 182 ENE types.

Table 6 shows the top 20 ENE types in the created ENE dictionary. Although Person occupies a good proportion of the entries, we also see large numbers of entities of other ENE types. Compared to the statistics of the training data (Table 1), the orders of the ENE types had some changes, reflecting the differences between the ENE corpus and Wikipedia. Although we cannot show the relationship between the ranks of ENE types and their counts by a figure similar to Fig. 1 for lack of space, the shape of the curve is nearly identical. Up to the 148th rank, we have over 100 entries, suggesting that we have a reasonable number of entities even when the ENE type is rather rare. A fine-grained ENE dictionary such as this one would be greatly helpful for various information extraction tasks.

5 Summary and Future Work

Using Wikipedia as a basic resource, we classified Wikipedia titles into ENE types to create an ENE dictionary. We derived a large number of features for the titles and trained a multiclass classifier. The features used encode various aspects of Wikipedia titles, such as those related

Rank	ENE type	Count	Ratio	Accum
1	Person	301625	29.700%	29.700%
2	Product_Other	53751	5.293%	34.993%
3	School	42179	4.153%	39.146%
4	Company	36462	3.590%	42.737%
5	City	34139	3.362%	46.098%
6	Road	32252	3.176%	49.274%
7	Music	29262	2.881%	52.155%
8	Position_Vocation	27712	2.729%	54.884%
9	Broadcast_Program	26759	2.635%	57.519%
10	Doctrine_Method_Other	24616	2.424%	59.943%
11	Station	21408	2.108%	62.051%
12	Book	15638	1.540%	63.591%
13	Game	14033	1.382%	64.972%
14	Movie	12956	1.276%	66.248%
15	Show_Organization	11778	1.160%	67.408%
16	Worship_Place	11076	1.091%	68.499%
17	Train	11014	1.085%	69.583%
18	Date	10624	1.046%	70.629%
19	Province	10213	1.006%	71.635%
20	GOE_Other	10190	1.003%	72.638%

Table 6: Top 20 ENE types in the created ENE dictionary.

to the surface string of a title, the content of the article, and the meta data provided with Wikipedia. By experiments, we successfully showed that it is possible to classify Wikipedia titles into ENE types with a reasonable accuracy of 79.63%. We also showed that, by applying the classifier to all Wikipedia titles and by discarding low-confidence entries, it is possible to create an ENE dictionary of over one million entities covering 182 ENE types with an estimated accuracy of 89.48%. Our main contributions are as follows: (1) We created the first large-scale ENE dictionary; no work has attempted to classify entities into such fine-grained types. (2) We made clear the features that are useful for the classification into ENE types; we ascertained the effectiveness of the previously proposed features regarding the content and meta data and newly found useful title-related features, such as the semantic categories found in the title and character bigrams.

Our method has a number of limitations. First, the classification accuracy is still low. As shown in Table 5, the accuracy for some ENE types are very poor. We want to devise new features and also find ways to augment our training data. Second, some features, e.g., character-based features, could be dependent on the Japanese language. We need to examine whether our method is applicable to other languages, especially English. Third, we could not cover all ENE types in our dictionary; many ENE types especially those related to numerical expressions were not included in the dictionary. This is mainly because Wikipedia titles do not cover such expressions. We want to investigate other resources to enrich our dictionary with currently scarce ENE types. Fourth, we want to evaluate our dictionary extrinsically, for example by using it in such information extraction tasks as question answering. We also want to use the dictionary to derive gazetteer features for our ENE recognizer that is under development. Finally, we want to extend our dictionary to ontologies so that it can be used for more intelligent tasks. Since ENE types themselves form a hierarchy, we will be able to use this structure to relate the ENEs in our created dictionary.

References

Bhole, A., Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Extracting named entities and relating them over time based on Wikipedia. *Informatica (Slovenia)*, 31(4):463–468.

Brunstein, A. (2002). Annotation guidelines for answer types. LDC2005T33, Linguistic Data Consortium.

Chang, J., Tsai, R. T.-H., and Chang, J. S. (2009). Wikisense: Supersense tagging of Wikipedia named entities based WordNet. In *Proc. PACLIC*, pages 72–81.

Dakka, W. and Cucerzan, S. (2008). Augmenting Wikipedia with named entity tags. In *Proc. IJCNLP*, pages 545–552.

Fuchi, T. and Takagi, S. (1998). Japanese morphological analyzer using word co-occurrence: JTAG. In *Proc. COLING-ACL*, pages 409–413.

Grishman, R. and Sundheim, B. (1996). Message understanding conference - 6: A brief history. In *Proc. COLING*, pages 466–471.

Hashimoto, T., Inui, T., and Murakami, K. (2008). Constructing extended named entity annotated corpora. In *SIG-NL-188, Information Processing Society of Japan*, pages 113–120. (in Japanese).

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., and Hayashi, Y. (1997). *Goi-Taikei – A Japanese Lexicon*. Iwanami Shoten.

Isozaki, H. and Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *Proc. COLING*, pages 1–7.

Kazama, J. and Torisawa, K. (2008). Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proc. ACL-HLT*, pages 407–415.

Mori, T., Masuda, H., Kiyota, Y., and Nakagawa, H. (2009). Wik-IE: A tool to extract structure of Wikipedia entries. In *SIG-SWO-A803-02, The Japanese Society for Artificial Intelligence*. (in Japanese).

Nagata, M., Shibaki, Y., and Yamamoto, K. (2010). Using Goi-Taikei as an upper ontology to build a large-scale japanese ontology from Wikipedia. In *Proc. the 6th Workshop on Ontologies and Lexical Resources*, pages 11–18.

Nothman, J., Curran, J. R., and Murphy, T. (2008). Transforming Wikipedia into named entity training data. In *Proc. the Australasian Language Technology Association Workshop 2008*, pages 124–132.

Ponzetto, S. P. and Strube, M. (2007). Deriving a large scale taxonomy from Wikipedia. In *Proc. AAAI*, pages 1440–1445.

Richman, A. E. and Schone, P. (2008). Mining Wiki resources for multilingual named entity recognition. In *Proc. ACL-HLT*, pages 1–9.

Saito, K. and Nagata, M. (2003). Multi-language named-entity recognition system based on HMM. In *Proc. the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 41–48.

Saleh, I., Darwish, K., and Fahmy, A. (2010). Classifying Wikipedia articles into NE's using SVM's with threshold adjustment. In *Proc. the 2010 Named Entities Workshop*, pages 85–92.

Sekine, S. and Isahara, H. (2000). IREX: IR & IE evaluation project in Japanese. In *Proc. LREC*.

Sekine, S. and Nobata, C. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *Proc. LREC*.

Sekine, S., Sudo, K., and Nobata, C. (2002). Extended named entity hierarchy. In *Proc. LREC*.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

Suzuki, J., McDermott, E., and Isozaki, H. (2006). Training conditional random fields with multivariate evaluation measures. In *Proc. COLING-ACL*, pages 217–224.

Tardif, S., Curran, J. R., and Murphy, T. (2009). Improved text categorisation for Wikipedia named entities. In *Proc. the Australasian Language Technology Association Workshop 2009*, pages 104–108.

Tkatchenko, M., Ulanov, A., and Simanovsky, A. (2011). Classifying Wikipedia entities into fine-grained classes. In *Proc. the 2011 IEEE 27th International Conference on Data Engineering Workshops*, pages 212–217.

Toral, A. and Muoz, R. (2006). Proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *Proc. the EACL 2006 workshop on NEW TEXT Wikis and blogs and other dynamic text sources*, pages 56–61.

Voorhees, E. M. and Dang, H. T. (2005). Overview of the TREC 2005 question answering track. In *Proc. TREC*.

Watanabe, Y., Asahara, M., and Matsumoto, Y. (2007). A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *Proc. EMNLP-CoNLL*, pages 649–657.

Zhang, Z. and Iria, J. (2009). A novel approach to automatic gazetteer generation using Wikipedia. In *Proc. the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 1–9.