Exploiting Wikipedia as External Knowledge for Named Entity Recognition

Jun'ichi Kazama and Kentaro Torisawa
Japan Advanced Institute of Science and Technology (JAIST)
Asahidai 1-1, Nomi, Ishikawa, 923-1292 Japan
{kazama, torisawa}@jaist.ac.jp

Abstract

We explore the use of Wikipedia as external knowledge to improve named entity recognition (NER). Our method retrieves the corresponding Wikipedia entry for each candidate word sequence and extracts a category label from the first sentence of the entry, which can be thought of as a definition part. These category labels are used as features in a CRF-based NE tagger. We demonstrate using the CoNLL 2003 dataset that the Wikipedia category labels extracted by such a simple method actually improve the accuracy of NER.

1 Introduction

It has been known that *Gazetteers*, or entity dictionaries, are important for improving the performance of named entity recognition. However, building and maintaining high-quality gazetteers is very time consuming. Many methods have been proposed for solving this problem by automatically extracting gazetteers from large amounts of texts (Riloff and Jones, 1999; Thelen and Riloff, 2002; Etzioni et al., 2005; Shinzato et al., 2006; Talukdar et al., 2006; Nadeau et al., 2006). However, these methods require complicated induction of patterns or statistical methods to extract high-quality gazetteers.

We have recently seen a rapid and successful growth of Wikipedia (http://www.wikipedia.org), which is an open, collaborative encyclopedia on the Web. Wikipedia has now more than 1,700,000 articles on the English version (March 2007) and the number is still increasing. Since Wikipedia aims to be an encyclopedia, most articles are about named entities and they are more structured than raw texts. Although it cannot be used as gazetteers directly since it is not intended as a machine readable resource, extracting knowledge such as gazetteers from Wikipedia will be much easier than from raw texts or from usual Web texts because of its structure. It is also important that Wikipedia is updated every day and therefore new named entities are added constantly. We think that extracting knowledge from Wikipedia for natural language processing is one of the promising ways towards enabling large-scale, real-life applications. In fact, many studies that try to exploit Wikipedia as a knowledge source have recently emerged (Bunescu and Paşca, 2006; Toral and Muñoz, 2006; Ruiz-Casado et al., 2006; Ponzetto and Strube, 2006; Strube and Ponzetto, 2006; Zesch et al., 2007).

As a first step towards such approach, we demonstrate in this paper that category labels extracted from the first sentence of a Wikipedia article, which can be thought of as the *definition* of the entity described in the article, are really useful to improve the accuracy of NER. For example, "Franz Fischler" has the article with the first sentence, "Franz Fischler (born September 23, 1946) is an Austrian politician." We extract "politician" from this sentence as the category label for "Franz Fischler". We use such category labels as well as matching information as features of a CRF-based NE tagger. In our experiments using the CoNLL 2003 NER dataset (Tjong et al., 2003), we demonstrate that we can improve performance by using the Wikipedia features by 1.58 points in F-measure from the baseline, and by 1.21 points from the model that only uses the gazetteers provided in the CoNLL 2003 dataset. Our final model incorporating all features achieved 88.02 in F-measure, which means a 3.03 point improvement over the baseline, which does not use any

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 698–707, Prague, June 2007. ©2007 Association for Computational Linguistics

gazetteer-type feature.

The studies most relevant to ours are Bunescu and Paşca (2006) and Toral and Muñoz (2006).

Bunescu and Paşca (2006) presented a method of disambiguating ambiguous entities exploiting internal links in Wikipedia as training examples. The difference however is that our method tries to use Wikipedia features for NER, not for disambiguation which assumes that entity regions are already found. They also did not focus on the first sentence of an article. Also, our method does not disambiguate ambiguous entities, since accurate disambiguation is difficult and possibly introduces noise. There are two popular ways for presenting ambiguous entities in Wikipedia. The first is to redirect users to a disambiguation page, and the second is to redirect users to one of the articles. We only focused on the second case and did not utilize disambiguation pages in this study. This method is simple but works well because the article presented in the second case represents in many cases the major meaning of the ambiguous entities and therefore that meaning frequently appears in a corpus.

Toral and Muñoz (2006) tried to extract gazetteers from Wikipedia by focusing on the first sentences. However, their way of using the first sentence is slightly different. We focus on the first noun phrase after *be* in the first sentence, while they used all the nouns in the sentence. By using these nouns and WordNet, they tried to map Wikipedia entities to abstract categories (e.g., LOC, PER ORG, MISC) used in usual NER datasets. We on the other hand use the obtained category labels directly as features, since we think the mapping performed automatically by a CRF model is more precise than the mapping by heuristic methods. Finally, they did not demonstrate the usefulness of the extracted gazetteers in actual NER systems.

The rest of the paper is organized as follows. We first explain the structure of Wikipedia in Section 2. Next, we introduce our method of extracting and using category labels in Section 3. We then show the experimental results on the CoNLL 2003 NER dataset in Section 4. Finally, we discuss the possibility of further improvement and future work in Section 5.

2 Wikipedia

2.1 Basic structure

An article in Wikipedia is identified by a unique name, which can be obtained by concatenating the words in the article title with underscore "_". For example, the unique name for the article, "David Beckham", is David_Beckham. We call these unique names "entity names" in this paper.

Wikipedia articles have many useful structures for knowledge extraction such as headings, lists, internal links, categories, and tables. These are marked up by using the Wikipedia syntax in source files, which authors edit. See the Wikipedia entry identified by How_to_edit_a_page for the details of the markup language.

We describe two important structures, redirections and disabiguation pages, in the following sections.

2.2 Redirection

Some entity names in Wikipedia do not have a substantive article and are only redirected to an article with another entity name. This mechanism is called "redirection". Redirections are marked up as "#REDIRECT [[A B C]]" in source files, where "[[...]]" is a syntax for a link to another article in Wikipedia (internal links). If the source file has such a description, users are automatically redirected to the article specified by the entity name in the brackes (A_B_C for the above example). Redirections are used for several purposes regarding ambiguity. For example, they are used for spelling resolution such as from "Apples" to "Apple" and abbreviation resolution such as from "MIT" to "Massachusetts Institute of Technology". They are also used in the context of more difficult disambiguations described in the next section.

2.3 Disambiguation pages

Some authors make a "disambiguation" page for an ambiguous entity name.[1] A disambiguation page typically enumerates possible articles for that name. For example, the page for "Beckham" enumerates "David Beckham (English footballer)", "Victoria

[1]We mean by "ambiguous" the case where a name can be used to refer to several difference entities (i.e., articles in Wikipedia).

Beckham (English celebrity and wife of David)", "Brice Beckham (American actor)", and so on. Most, but not all, disambiguation pages have a name like Beckham_(disambiguation) and are sometimes used with redirection. For example, Beckham is redirected to Beckham_(disambiguation) in the above example. However, it is also possible that Beckham redirects to one of the articles (e.g, David_Beckham). As we mentioned, we did not utilize the disambiguation pages and relied on the above case in this study.

2.4 Data

Snapshots of the entire contents of Wikipedia are provided in XML format for each language version. We used the English version at the point of February 2007, which includes 4,030,604 pages.[2] We imported the data into a text search engine[3] and used it for the research.

3 Method

In this section, we describe our method of extracting category labels from Wikipedia and how to use those labels in a CRF-based NER model.

3.1 Generating search candidates

Our purpose here is to find the corresponding entity in Wikipedia for each word sequence in a sentence. For example, given the sentence, "Rare Jimi Hendrix song draft sells for almost $17,000", we would like to know that "Jimi Hendrix" is described in Wikipedia and extract the category label, "musician", from the article. However, considering all possible word sequences is costly. We thus restricted the candidates to be searched to the word sequences of no more than eight words that start with a word containing at least one capitalized letter.[4]

3.2 Finding category labels

We converted a candidate word sequence to a Wikipedia entity name by concatenating the words with underscore. For example, a word sequence "Jimi Hendrix" is converted to Jimi_Hendrix. Next, we retrieved the article corresponding to the entity name.[5] If the page for the entity name is a redirection page, we followed redirection until we find a non-redirection page.

Although there is no strict formatting rule in Wikipedia, the convention is to start an article with a short sentence defining the entity the article describes. For example, the article for Jimi_Hendrix starts with the sentence, "Jimi Hendrix (November 27, 1942, Seattle, Washington - September 18, 1970, London, England) was an American guitarist, singer and songwriter." Most of the time, the head noun of the noun phrase just after *be* is a good category label. We thus tried to extract such head nouns from the articles.

First, we eliminated unnecessary markup such as italics, bold face, and internal links from the article. We also converted the markup for internal links like `[[Jimi Hendrix|Hendrix]]` to `Hendrix`, since the part after `|`, if it exists, represents the form to be displayed in the page. We also eliminated template markup, which is enclosed by `{{` and `}}`, because template markup sometimes comes at the beginning of the article and makes the extraction of the first sentence impossible.[6] We then divided the article into lines according to the new line code, `\n`, `<br>` HTML tags, and a very simple sentence segmentation rule for period (.). Next, we removed lines that match regular expression `/^\s*:/` to eliminate the lines such as:

> *This article is about the tree and its fruit.*
> *For the consumer electronics corporation, see Apple Inc.*

These sentences are not the content of the article but often placed at the beginning of an article. Fortunately, they are usually marked up using `:`, which is for indentation.

After the preprocessing described above, we extracted the first line in the remaining lines as the first sentence from which we extract a category label.

[2]The number of article pages is 2,954,255 including redirection pages

[3]We used HyperEstraier available at http://hyperestraier.sourceforge.net/index.html

[4]Words such as "It" and "He" are not considered as capitalized words here (we made a small list of stop words).

[5]There are pages for other than usual articles in the Wikipedia data. They are distinguished by a *namespace* attribute. To retrieve articles, we only searched in namespace 0, which is for usual articles.

[6]Templates are used for example to generate profile tables for persons.

We then performed POS tagging and phrase chunking. TagChunk (Daumé III and Marcu, 2005)[7] was used as a POS/chunk tagger. Next, we extracted the first noun phrase after the first "is", "was", "are", or "were" in the sentence. Basically, we extracted the last word in the noun phrase as the category label. However, we used the second noun phrase when the first noun phrase ended with "one", "kind", "sort", or "type", or it ended with "name" followed by "of". These rules were for treating examples like:

Jazz is $[\text{a kind}]_{\text{NP}}$ $[\text{of}]_{\text{PP}}$ $[\text{music}]_{\text{NP}}$ characterized by swung and blue notes.

In these cases, we would like to extract the head noun of the noun phrase after "of" (e.g., "music" in instead of "kind" for the above example). However, we would like to extract "name" itself when the sentence was like "Ichiro is a Japanese given name".

We did not utilize Wikipedia's "Category" sections in this study, since a Wikipedia article can have more than one category, and many of them are not clean hypernyms of the entity as far as we observed. We will need to select an appropriate category from the listed categories in order to utilize the Category section. We left this task for future research.

3.3 Using category labels as features

If we could find the category label for the candidate word sequence, we annotated it using IOB2 tags in the same way as we represent named entities. In IOB2 tagging, we use "B-X", "I-X", and "O" tags, where "B", "I", and "O" means the beginning of an entity, the inside of an entity, and the outside of entities respectively. Suffix X represents the category of an entity.[8] In this case, we used the extracted category label as the suffix. For example, if we found that "Jimi Hendrix" was in Wikipedia and extracted "guitarist" as the category label, we annotated the sentence, "Rare Jimi Hendrix song draft sells for almost $17,000", as:

Rare_{O} $\text{Jimi}_{\text{B-guitarist}}$ $\text{Hendrix}_{\text{I-guitarist}}$ song_{O} draft_{O} for_{O} almost_{O} $\$17{,}000_{\text{O}}$ $._{\text{O}}$

Note that we adopted the leftmost longest match if there were several possible matchings. These IOB2 tags were used in the same way as other features in our NE tagger using Conditional Random Fields (CRFs) (Lafferty et al., 2001). For example, we used a feature such as "the Wikipedia tag is B-guitarist and the NE tag is B-PER".

[7] http://www.cs.utah.edu/~hal/TagChunk/

[8] We use bare "B", "I", and "O" tags if we want to represent only the matching information.

4 Experiments

In this section, we demonstrate the usefulness of the extracted category labels for NER.

4.1 Data and setting

We used the English dataset of the CoNLL 2003 shared task (Tjong et al., 2003). It is a corpus of English newspaper articles, where four entity categories, PER, LOC, ORG, and MISC are annotated. It consists of training, development, and testing sets (14,987, 3,466, and 3,684 sentences, respectively). We concatenated the sentences in the same document according to the document boundary markers provided in the dataset.[9] This generated 964 documents for the training set, 216 documents for the development set, and 231 documents for the testing set. Although automatically assigned POS and chunk tags are also provided in the dataset, we used TagChunk (Daumé III and Marcu, 2005)[10] to assign POS and chunk tags, since we observed that accuracy could be improved, presumably due to the quality of the tags.[11]

We used the features summarized in Table 1 as the baseline feature set. These are similar to those used in other studies on NER. We omitted features whose surface part described in Table 1 occurred less than twice in the training corpus.

Gazetteer files for the four categories, PER (37,831 entries), LOC (10,069 entries), ORG (3,439 entries), and MISC (3,045 entries), are also provided in the dataset. We compiled these files into one gazetteer, where each entry has its entity category, and used it in the same way as the Wikipedia feature described in Section 3.3. We will compare features using this gazetteer with those using Wikipedia in the following experiments.

[9] We used sentence concatenation because we found it improves the accuracy in another study (Kazama and Torisawa, 2007).

[10] http://www.cs.utah.edu/~hal/TagChunk/

[11] This is not because TagChunk overfits the CoNLL 2003 dataset (TagChunk is trained on the Penn Treebank (Wall Street Journal), while the CoNLL 2003 data are taken from the Reuters corpus).

Table 1: Baseline features. The value of a node feature is determined from the current label, y_0, and a surface feature determined only from $\boldsymbol{x}$. The value of an edge feature is determined by the previous label, y_{-1}, the current label, y_0, and a surface feature. Used surface features are the word (w), the downcased word (wl), the POS tag (pos), the chunk tag (chk), the prefix of the word of length n (pn), the suffix (sn), the word form features: 2d - cp (these are based on (Bikel et al., 1999))

Node features: $\{"", \mathrm{x}_{-2}, \mathrm{x}_{-1}, \mathrm{x}_0, \mathrm{x}_{+1}, \mathrm{x}_{+2}\} \times y_0$
x = w, wl, pos, chk, p1, p2, p3, p4, s1, s2, s3, s4, 2d, 4d, d&a, d&-, d&/, d&,, d&., n, ic, ac, l, cp
Edge features: $\{"", \mathrm{x}_{-2}, \mathrm{x}_{-1}, \mathrm{x}_0, \mathrm{x}_{+1}, \mathrm{x}_{+2}\} \times y_{-1} \times y_0$
x = w, wl, pos, chk, p1, p2, p3, p4, s1, s2, s3, s4, 2d, 4d, d&a, d&-, d&/, d&,, d&., n, ic, ac, l, cp
Bigram node features: $\{\mathrm{x}_{-2}\mathrm{x}_{-1}, \mathrm{x}_{-1}\mathrm{x}_0, \mathrm{x}_0\mathrm{x}_{+1}\} \times y_0$
x = wl, pos, chk
Bigram edge features: $\{\mathrm{x}_{-2}\mathrm{x}_{-1}, \mathrm{x}_{-1}\mathrm{x}_0, \mathrm{x}_0\mathrm{x}_{+1}\} \times y_{-1} \times y_0$
x = wl, pos, chk

We used CRF++ (ver. 0.44)[12] as the basis of our implementation of CRFs. We implemented scaling, which is similar to that for HMMs (see for instance (Rabiner, 1989)), in the forward-backward phase of CRF training to deal with long sequences due to sentence concatenation.[13] We used Gaussian regularization to avoid overfitting. The parameter of the Gaussian, σ^2, was tuned using the development set.[14] We stopped training when the relative change in the log-likelihood became less than a pre-defined threshold, 0.0001, for at least three iterations.

4.2 Category label finding

Table 2 summarizes the statistics of category label finding for the training set. Table 3 lists examples of the extracted categories. As can be seen, we could extract more than 1,200 distinct category labels. These category labels seem to be useful, although there is no guarantee that the extracted category label is correct for each candidate.

Table 2: Statistics of category label finding.

search candidates (including duplication)	256,418
candidates having Wikipedia article	39,258
(articles found by redirection)	9,587
first sentence found	38,949
category label extracted	23,885
(skipped "one")	544
(skipped "kind")	14
(skipped "sort")	1
(skipped "type")	41
(skipped "name of")	463
distinct category labels	1,248

Table 3: Examples of category labels (top 20).

category	frequency	# distinct entities
country	2598	152
city	1436	284
name	1270	281
player	578	250
day	564	131
month	554	15
club	537	167
surname	515	185
capital	454	79
state	416	60
term	369	78
form	344	40
town	287	97
cricketer	276	97
adjective	260	6
golfer	229	88
world	221	24
team	220	52
organization	214	38
second	212	1

4.3 Feature comparison

We compared the following features in this experiment.

Gazetteer Match (gaz_m) This feature represents the matching with a gazetteer entry by using "B", "I", and "O" tags. That is, this is the gazetteer version of **wp_m** below.

Gazetteer Category Label (gaz_c) This feature represents the matching with a gazetteer entry and its category by using "B-X", "I-X", and "O" tags, where X is one of "PER", "LOC", "ORG", and "MISC". That is, this is the gazetteer version of **wp_c** below.

Wikipedia Match (wp_m) This feature represents the matching with a Wikipedia entity by using "B", "I", and "O" tags.

[12] http://chasen.org/˜taku/software/CRF++

[13] We also replaced the optimization module in the original package with that used in the Amis maximum entropy estimator (http://www-tsujii.is.s.u-tokyo.ac.jp/amis) since we encountered problems with the provided module in some cases. Although this Amis module implements BLMVM (Benson and Moré, 2001), which supports the bounding of weights, we did not use this feature in this study (i.e., we just used it as the replacement for the L-BFGS optimizer in CRF++).

[14] We tested 15 points: $\{0.01, 0.02, 0.04, \ldots, 163.84, 327.68\}$.

Table 4: Statistics of gazetteer and Wikipedia features. Rows "NEs (%)" show the number of matches that also matched the regions of the named entities in the training data, and the percentage of such named entities (there were 23,499 named entities in total in the training data).

Gazetteer Match (gaz_m)	
matches	12,397
NEs (%)	6,415 (27.30%)
Wikipedia Match (wp_m)	
matches	27,779
NEs (%)	16,600 (70.64%)
Wikipedia Category Label (wp_c)	
matches	18,617
NEs (%)	11,645 (49.56%)
common with gazetteer match	5,664

Wikipedia Category Label (wp_c) This feature represents the matching with a Wikipedia entity and its category in the way described Section in 3.3. Note that this feature only fires when the category label is successfully extracted from the Wikipedia article.

For these **gaz_m**, **gaz_c**, **wp_m**, and **wp_c**, we generate the node features, the edge features, the bigram node features, and the bigram edge features, as described in Table 1.

Table 4 shows how many matches (the leftmost longest matches that were actually output) were found for **gaz_m**, **wp_m**, and **wp_c**. We omitted the numbers for **gaz_c**, since they are same as **gaz_m**. We can see that Wikipedia had more matches than the gazetteer, and covers more named entities (more than 70% of the NEs in the training corpus). The overlap between the gazetteer matches and the Wikipedia matches was moderate as the last row indicates (5,664 out of 18,617 matches). This indicates that Wikipedia has many entities that are not listed in the gazetteer.

We then compared the baseline model (**baseline**), which uses the feature set in Table 1, with the following models to see the effect of the gazetteer features and the Wikipedia features.

(A): + gaz_m This uses **gaz_m** in addition to the features in **baseline**.

(B): + gaz_m, gaz_c This uses **gaz_m** and **gaz_c** in addition to the features in **baseline**.

(C): + wp_m This uses **wp_m** in addition to the features in **baseline**.

(D): + wp_m, wp_c This uses **wp_m** and **wp_c** in addition to the features in **baseline**.

(E): + gaz_m, gaz_c, wp_m, wp_c This uses **gaz_m**, **gaz_c**, **wp_m**, and **wp_c** in addition to the features in **baseline**.

(F): + gaz_m, gaz_c, wp_m, wp_c (word comb.) This model uses the combination of words (wl) and **gaz_m**, **gaz_c**, **wp_m**, or **wp_c**, in addition to the features of model **(E)**. More specifically, these features are the node feature, $\mathrm{wl}_0 \times x_0 \times y_0$, the edge feature, $\mathrm{wl}_0 \times x_0 \times y_{-1} \times y_0$, the bigram node feature, $\mathrm{wl}_{-1} \times \mathrm{wl}_0 \times x_{-1} \times x_0 \times y_0$, and the bigram edge feature, $\mathrm{wl}_{-1} \times \mathrm{wl}_0 \times x_{-1} \times x_0 \times y_{-1} \times y_0$, where x is one of gaz_m, gaz_c, wp_m, and wp_c. We tested this model because we thought these combination features could alleviate the problem by incorrectly extracted categories in some cases, if there is a characteristic correlation between words and incorrectly extracted categories.

Table 5 shows the performance of these models. The results for (A) and (C) indicate that the matching information alone does not improve accuracy. This is because entity regions can be identified fairly correctly if models are trained using a sufficient amount of training data. The category labels, on the other hand, are actually important for improvement as the results for (B) and (D) indicate. The gazetteer model, (B), improved F-measure by 1.47 points from the baseline. The Wikipedia model, (D), improved F-measure by 1.58 points from the baseline. The effect of the gazetteer feature, **gaz_c**, and the Wikipedia features, **wp_c**, did not differ much. However, it is notable that the Wikipedia feature, which is obtained by our very simple method, achieved such an improvement easily.

The results for model (E) show that we can improve accuracy further, by using the gazetteer features and the Wikipedia features together. Model (E) achieved 87.67 in F-measure, which is better than those of (B) and (D). This result coincides with the fact that the overlap between the gazetteer feature

Table 5: Effect of gazetteer and Wikipedia features.

model (best σ^2)	category	dev P	dev R	dev F	eval P	eval R	eval F
baseline (20.48)	PER	90.29	92.89	91.57	87.19	91.34	89.22
	LOC	93.32	92.81	93.07	88.14	88.25	88.20
	ORG	85.36	83.07	84.20	82.25	78.93	80.55
	MISC	92.21	84.71	88.30	79.58	75.50	77.49
	ALL	90.42	89.38	**89.90**	85.17	84.81	**84.99**
(A): + gaz_m (81.92)	PER	90.60	92.56	91.57	87.90	90.72	89.29
	LOC	92.84	93.20	93.02	88.26	88.37	88.32
	ORG	85.54	82.92	84.21	82.37	79.05	80.68
	MISC	92.15	85.25	88.56	78.73	75.93	77.30
	ALL	90.41	89.45	**89.92**	85.33	84.76	**85.04**
(B): + gaz_m, gaz_c (163.84)	PER	92.45	94.41	93.42	90.78	91.96	91.37
	LOC	94.43	94.07	94.25	89.98	89.33	89.65
	ORG	86.68	85.38	86.03	82.43	81.34	81.88
	MISC	92.47	85.25	88.71	79.50	76.78	78.12
	ALL	91.77	90.84	**91.31**	86.74	86.17	**86.46**
(C): + wp_m (163.84)	PER	90.84	92.56	91.69	87.77	90.11	88.92
	LOC	92.63	93.03	92.83	87.23	88.07	87.65
	ORG	86.19	83.74	84.95	81.77	79.65	80.70
	MISC	91.69	84.92	88.18	79.04	75.21	77.08
	ALL	90.49	89.53	**90.01**	84.85	84.58	**84.71**
(D): + wp_m, wp_c (163.84)	PER	91.57	94.41	92.97	90.13	92.02	91.06
	LOC	94.78	93.96	94.37	89.41	89.63	89.52
	ORG	87.36	85.01	86.17	82.70	82.00	82.35
	MISC	91.87	84.60	88.09	81.34	76.35	78.77
	ALL	91.68	90.63	**91.15**	86.71	86.42	**86.57**
(E): + gaz_m, gaz_c, wp_m, wp_c (40.96)	PER	93.32	95.49	94.39	92.28	93.14	92.71
	LOC	94.91	94.39	94.65	90.69	90.47	90.58
	ORG	88.27	86.95	87.60	83.08	83.68	83.38
	MISC	93.14	85.36	89.08	81.33	76.92	79.06
	ALL	92.65	91.65	**92.15**	87.79	87.55	**87.67**
(F): + gaz_m, gaz_c, wp_m, wp_c (word comb.) (5.12)	PER	93.38	95.66	94.50	92.52	93.26	92.89
	LOC	94.88	94.77	94.83	91.25	90.71	90.98
	ORG	88.67	86.95	87.80	83.61	84.17	83.89
	MISC	93.56	85.03	89.09	81.63	77.21	79.36
	ALL	92.82	91.77	**92.29**	88.21	87.84	**88.02**

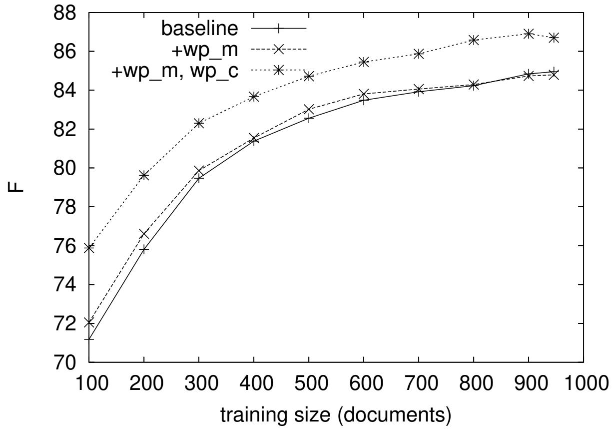

Figure 1: Relation between the training size and the accuracy.

and the Wikipedia feature was not so large. If we consider model (B) a practical baseline, we can say that the Wikipedia features improved the accuracy in F-measure by 1.21 points.

We can also see that the effect of the gazetteer features and the Wikipedia features were consistent irrespective of categories (i.e., PER, LOC, ORG, or MISC) and performance measures (i.e., precision, recall, or F-measure). This indicates that gazetteer-type features are reliable as features for NER.

The final model, (F), achieved 88.02 in F-measure. This is greater than that of the baseline by 3.03 points, showing the usefulness of the gazetteer type features.

4.4 Effect of training size

We observed in the previous experiment that the matching information alone was not useful. However, the situation may change if the size of the training data becomes small. We thus observed the effect of the training size for the Wikipedia features **wp_m** and **wp_c** (we used $\sigma^2 = 10.24$). Figure 1 shows the result. As can be seen, the matching information had a slight positive effect when the size of training data was small. For example, it improved F-measure by 0.8 points from the baseline at 200 documents. However, the superiority of category labels over the matching information did not change. The effect of category labels became greater as the training size became smaller. Its effect compared with the matching information alone was 3.01 points at 200 documents, while 1.91 points at 964 documents (i.e., the whole training data).

Table 6: Breakdown of improvements and errors.

(B) → (E)	num.	$\bar{g} \wedge \bar{w}$	$\bar{g} \wedge w$	$g \wedge \bar{w}$	$g \wedge w$
inc → inc	442	219	123	32	68
inc → cor	102	28	56	3	15
cor → inc	56	28	13	7	8
cor → cor	5,342	1,320	1,662	723	1,637

4.5 Improvement and error analysis

We analyze the improvements and the errors caused by using the Wikipedia features in this section.

We compared the output of (B) and (E) for the development set. There were 5,942 named entities in the development set. We assessed how the labeling for these entities changed between (B) and (E). Note that the labeling for 199 sentences out of total 3,466 sentences was changed. Table 6 shows the breakdown of the improvements and the errors. "**inc**" in the table means that the model could not label the entity correctly, i.e., the model could not find the entity region at all, or it assigned an incorrect category to the entity. "**cor**" means that the model could label the entity correctly. The column, "**inc** → **cor**", for example, has the numbers for the entities that were labeled incorrectly by (B) but labeled correctly by (E). We can see from the column, "num", that the number of improvements by (E) exceeded the number of errors introduced by (E) (102 vs. 56). Table 6 also shows how the gazetteer feature, **gaz_c**, and the Wikipedia feature, **wp_c**, fired in each case. We mean that the gazetteer feature fired by using "g", and that the Wikipedia feature fired by using "w". "$\bar{g}$" and "$\bar{w}$" mean that the feature did not fire. As is the case for other machine learning methods, it is difficult to find a clear reason for each improvement or error. However, we can see that the number of $\bar{g} \wedge w$ exceeded those of other cases in the case of "**inc** → **cor**", meaning that the Wikipedia feature contributed the most.

Finally, we show an example of case **inc** → **cor** in Figure 2. We can see that "Gazzetta dello Sport" in the sentence was correctly labeled as an entity of "ORG" category by model (E), because the Wikipedia feature identified it as a newspaper entity.[15]

[15] Note that the category label, "character", for "Atalanta" in the sentence was not correct in this context, which is an example where disambiguation is required. The final recognition was correct in this case presumably because of the information from **gaz_c** feature.

The	Gazzetta	dello	Sport	said	the	deal	would	cost	Atalanta	around	$	600,000	.	
O	O	O	B-ORG	O	O	O	O	O	B-ORG	O	O	O	O	**- gaz_c**
O	B-newspaper	I-newspaper	I-newspaper	O	O	O	O	O	B-character	O	O	O	O	**- wp_c**
O	B-ORG	I-ORG	I-ORG	O	O	O	O	O	B-ORG	O	O	O	O	**- correct**
O	B-LOC	O	B-ORG	O	O	O	O	O	B-ORG	O	O	O	O	**- (B)**
O	B-ORG	I-ORG	I-ORG	O	O	O	O	O	B-ORG	O	O	O	O	**- (E)**

Figure 2: An example of improvement caused by Wikipedia feature.

5 Discussion and Future Work

We have empirically shown that even category labels extracted from Wikipedia by a simple method such as ours really improves the accuracy of a NER model. The results indicate that structures in Wikipedia are suited for knowledge extraction. However, the results also indicate that there is room for improvement, considering that the effects of **gaz_c** and **wp_c** were similar, while the matching rate was greater for **wp_c**. An issue, which we should treat, is the disambiguation of ambiguous entities. Our method worked well although it was very simple, presumably because of the following reason. (1) If a retrieved page is a disambiguation page, we cannot extract a category label and critical noise is not introduced. (2) If a retrieved page is not a disambiguation page, it will be the page describing the major meaning determined by the agreement of many authors. The extracted categories are useful for improving accuracy because the major meaning will be used frequently in the corpus. However, it is clear that disambiguation techniques are required to achieve further improvements. In addition, if Wikipedia grows at the current rate, it is possible that almost all entities become ambiguous and a retrieved page is a disambiguation page most of the time. We will need a method for finding the most suitable article from the articles listed in a disambiguation page.

An interesting point in our results is that Wikipedia category labels improved accuracy, although they were much more specific (more than 1,200 categories) than the four categories of the CoNLL 2003 dataset. The correlation between a Wikipedia category label and a category label of NER (e.g., "musician" to "PER") was probably learned by a CRF tagger. However, the merit of using such specific Wikipedia labels will be much greater when we aim at developing NER systems for more fine-grained NE categories such as proposed in Sekine et al. (2002) or Shinzato et al. (2006). We thus would like to investigate the effect of the Wikipedia feature for NER with such fine-grained categories as well. Disambiguation techniques will be important again in that case. Although the impact of ambiguity will be small as long as the target categories are abstract and an incorrectly extracted category is in the same abstract category as the correct one (e.g., extracting "footballer" instead of "cricketer"), such mis-categorization is critical if it is necessary to distinguish footballers from cricketers.

6 Conclusion

We tried to exploit Wikipedia as external knowledge to improve NER. We extracted a category label from the first sentence of a Wikipedia article and used it as a feature of a CRF-based NE tagger. The experiments using the CoNLL 2003 NER dataset demonstrated that category labels extracted by such a simple method really improved accuracy. However, disambiguation techniques will become more important as Wikipedia grows or if we aim at more fine-grained NER. We thus would like to incorporate a disambiguation technique into our method in future work. Exploiting Wikipedia structures such as disambiguation pages and link structures will be the key in that case as well.

References

S. J. Benson and J. J. Moré. 2001. A limited memory variable metric method for bound constraint minimization. Technical Report ANL/MCS-P909-0901, Argonne National Laboratory.

D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231.

R. Bunescu and M. Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL 2006*.

H. Daumé III and D. Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *ICML 2005*.

O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web – an experimental study. *Artificial Intelligence Journal*.

J. Kazama and K. Torisawa. 2007. A new perceptron algorithm for sequence labeling with non-local features. In *EMNLP-CoNLL 2007*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289.

D. Nadeau, Peter D. Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *19th Canadian Conference on Artificial Intelligence*.

S. P. Ponzetto and M. Strube. 2006. Exploiting semantic role lebeling, WordNet and Wikipedia for coreference resolution. In *NAACL 2006*.

L. R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *16th National Conference on Artificial Intelligence (AAAI-99)*.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. 2006. From Wikipedia to semantic relationships: a semi-automated annotation approach. In *Third European Semantic Web Conference (ESWC 2006)*.

S. Sekine, K. Sudo, and C. Nobata. 2002. Extended named entity hierarchy. In *LREC '02*.

K. Shinzato, S. Sekine, N. Yoshinaga, and K. Torisawa. 2006. Constructing dictionaries for named entity recognition on specific domains from the Web. In *Web Content Mining with Human Language Technologies Workshop on the 5th International Semantic Web*.

M. Strube and S. P. Ponzetto. 2006. WikiRelate! computing semantic relatedness using Wikipedia. In *AAAI 2006*.

P. P. Talukdar, T. Brants, M. Liberman, and F. Pereira. 2006. A context pattern induction method for named entity extraction. In *CoNLL 2006*.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern context. In *EMNLP 2002*.

E. F. Tjong, K. Sang, and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CoNLL 2003*.

A. Toral and R. Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *EACL 2006*.

T. Zesch, I. Gurevych, and M. Möhlhäuser. 2007. Analyzing and accessing Wikipedia as a lexical semantic resource. In *Biannual Conference of the Society for Computational Linguistics and Language Technology*.