Named Entity Extraction from Japanese Broadcast News

Akio Kobayashi†, Franz J. Och‡ and Hermann Ney*

†NHK Science & Technical Research Laboratories, Japan
‡University of Southern California/Information Science Institute, USA
*Lehrstuhl für Informatik IV, Computer Science Department
RWTH Aachen - University of Technology, Germany
kobayashi-a.fs@nhk.or.jp, och@isi.edu, ney@informatik.rwth-aachen.de

Abstract

This paper describes a method for named entity extraction from Japanese broadcast news. Our proposed named entity tagger gives entity categories for every character in order to deal with unknown words and entities correctly. This character-based tagger has models designed by maximum entropy modeling. We discuss the efficiency of the proposed tagger by comparison with a conventional word-based tagger. The results indicate that the capability of the taggers depends on the entity categories. Therefore, the features derived from both character and word contexts are required to obtain high performance of named entity extraction.

1. Introduction

Recent progress of natural language processing (NLP) has enabled a variety of practical applications. For example, simple n-gram language models in a speech recognition system[1] provide real-time captioning services in broadcast news for hearing impaired people. However, such real-time applications raise a serious issue of maintenance. The systems must register new words observed in the latest news into their lexicons in order to cover the contents of upcoming events. Names of persons, companies and industrial products account for most of these new words. Named Entity (NE) extraction, an information retrieval technique, provides the solution since it can extract such meaningful expressions automatically from a large quantity of documents.

This paper describes a method for NE extraction from Japanese broadcast news. We propose a NE tagger which determines NE categories by evaluating every character. Maximum Entropy (ME) modeling, which is a very powerful method of representing properties of natural language, is applied for the models of the tagger to be realized.

In Section 2, we explain NE extraction and the categories of entities. The motivation for our character-based NE tagger is presented. In Section 3, conditional ME models for NE extraction and their features are briefly introduced. In Section 4, we discuss the efficiency of our proposed character-based NE tagger by comparison with a conventional word-based tagger, which determines the categories for every word.

2. Named Entity Extraction

2.1. Named Entity Extraction

NE extraction is for obtaining names of persons, countries, organizations and artifacts from documents. This area has been researched through contests such as MUC-7 (Message Understanding Conference) as part of information retrieval. A similar contest targeting Japanese NE tasks (IREX, Information Retrieval EXercise) was held in Japan[10]. Figure 1 illustrates a Japanese sentence including NEs (The meaning of the sentence is "Prime Minister Koizumi visited China this morning"). In the figure, "Koizumi" is a person’s name and "China" is the name of a location.

Table 1: Categories

<table>
<thead>
<tr>
<th>category</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>artifact</td>
<td>industrial product, etc.</td>
</tr>
<tr>
<td>organization</td>
<td>organization / company</td>
</tr>
<tr>
<td>location</td>
<td>country / place</td>
</tr>
<tr>
<td>name</td>
<td>person</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>money</td>
<td>money</td>
</tr>
<tr>
<td>percent</td>
<td>percentage, ratio, etc.</td>
</tr>
<tr>
<td>none</td>
<td>non-NE</td>
</tr>
</tbody>
</table>

Figure 1: Example of a Japanese sentence

2.2. Categories

We defined eight unique NE categories according to the definition by the IREX (Information Retrieval EXercise) Committee[5][10](Table 1). The distinction between the IREX definition and ours is that a category is not tied with a word but with a character in our definition.

Each tag is extended by four markers; start, middle, end and unique positional markers are introduced so that entity boundaries can be marked. A start marker is attached to the beginning character of an entity. An end marker is also given to the last character of an entity with its category. If an NE consists of only one character, a unique marker is attached. A middle marker is given to all characters without any markers above.
2.3. Characteristics of Japanese

We confirm the motivation of our proposed character-based NE tagger by viewing characteristics in the Japanese language. One of the characteristics of Japanese is linguistic agglutination, which means that there is no delimiter in Japanese sentences\(^1\). This fact shows that basic statistics (e.g., word frequencies) cannot be obtained without morphological analysis\(^2\). Most popular Japanese morphological analyzers such as “Cha-Sen”[6] and “JUMAN”[7] produce a morpheme sequence from a sentence using morpheme and part-of-speech (POS) n-grams or rules written by hand. However, these analyzers cannot deal with unknown morphemes correctly because they yield morpheme sequences according to their morpheme-based lexicon. Analyzed morpheme sequences inevitably include incorrect morpheme boundaries.

Secondly, Japanese has many characters. One Japanese major machine-readable character set includes over 6,000 unique characters. Nagao and Morii[8] reported that character-based n-grams were important to determine morpheme boundaries. According to their survey, there are less frequent character pairs enclosing a morpheme boundary than pairs that do not enclose any boundaries. Nagao[9] studied a morphological analyzer using statistics of characters (i.e., character n-grams). He demonstrated that these character-based statistics should be expected for efficient morphological analyzing. In particular, they are useful determining morpheme boundaries surrounding unknown morphemes.

These surveys indicate that characters are essential features of NE extraction. In the following section, we give a detailed description of the NE tagger which makes decisions of NE categories for every character using statistics of characters.

3. NE Extraction Using Maximum Entropy Modeling

3.1. Formulation of NE Extraction

NE extraction is defined as an optimization problem:

\[
\mathbf{t}^* = \text{arg max}_t \prod_j P(t_j|t_{j-1}, C_{j-m}^{j+m}) \tag{1}
\]

The problem is to find the optimum category sequence \( \mathbf{t}^* \) using a statistical model \( P(t_j|t_{j-1}, C_{j-m}^{j+m}) \). Here, \( t_j \) denotes a category, and \( C_{j-m}^{j+m} \) denotes a certain context associated with \( t_j \). In this paper, \( C_{j-m}^{j+m} \) represents either a sequence of words or a sequence of characters.

3.2. Maximum Entropy Modeling

We briefly introduce maximum entropy modeling\(^2\) to realize the statistical models in Equation (1). Let a pair \((x, y)\) be a co-occurrence observed in training data. \( y \) is a symbol derived from an observation and \( x \) is a sequence of observations. Define \( f_i(x, y) \) (\( i = 1, \ldots, I \)) as a binary “feature” function representing characteristics of the event \((x, y)\).

\[
\mathcal{F} = \{ f_i : (x, y) \mapsto \{0, 1\}, i \in \{1, \ldots, I\}\} \tag{2}
\]

where \( f_i(x, y) \) returns 1 only if \((x, y)\) matches an adequate condition assigned to \( f_i(x, y) \). Given a set of features and training data, the solution of the conditional ME modeling is

\[
P_{ME}(y|x) = \frac{\exp \sum_i \lambda_i f_i(x, y)}{\sum_{y'} \exp \sum_j \lambda_j f_j(x, y')} \tag{3}
\]

where \( \lambda_i \) is a model parameter determined by the Generalized Iterative Scaling algorithm\(^4\) under the following constraints:

\[
E_{P_{ME}}[f_i] = E_{\hat{P}}[f_i] \tag{4}
\]

\( E_{P_{ME}}[f_i] \) denotes the expectation of \( f_i \) on \( P_{ME} \). \( \hat{P} \) represents the empirical distribution over given training data.

3.3. Features

3.3.1. Character Context Features

Initially, character context features of ME models are introduced.

Let \( c_j \) be a character with a category \( t_j \) to be predicted. For an event \((C_j, t_j)\), \( C_j \) is a sequence of characters \{\(c_{j-n}, \ldots, c_j, \ldots, c_{j+n}\}\}. The character context feature is defined as follows:

\[
\lambda(C_j, t_j) = \delta(t_k, t_{k'}) \times \delta(C_j, C_{j'}) \tag{5}
\]

where \( \delta(x, x') \) denotes Kronecker’s delta function and it returns 1 only if \( x = x' \) while it returns 0 otherwise. A character context feature has either a sequence of \( c_{j-n}, \ldots, c_j \) as a left context or a sequence of \( c_j, \ldots, c_{j+n} \) as a right context. A feature has \( n(n = 1, 2, \ldots) \) characters, therefore, features with various context length can be activated at once.

3.3.2. Word Context Features

Word context features are analogously introduced into ME modeling. The definition of the word context feature is

\[
\lambda(C^{m}, t_k) = \delta(t_k, t_{k'}) \times \delta(C^{m}, C_{k'}) \tag{6}
\]

where \( C^{m} \) is a context including a word sequence \{\(w_{l-m}, \ldots, w_l, \ldots, w_{l+m}\}\}(\(m = 0, 1, \ldots\)). In the sequence, \( w_l \) is a word including the character \( c_j \). Then, \( w_{l-1} \) is a predecessor word of \( w_l \), and \( w_{l+1} \) represents a successor. Note that each word context feature establishes a link between a word sequence and a category. Word contexts contain information on word boundaries as a prior knowledge while character contexts never include them. The drawback of using word contexts is that unknown words are not treated correctly.

3.4. Word-based Tagger

The word-based taggers are commonly used in Japanese NE extraction. Borthwick[3] and Uchimoto[11] performed experiments on word-based NE taggers using ME models for IREX tasks. In these studies, the features of the ME models were based on word contexts and several lexical properties (parts of speech, etc.). These properties were induced from word sequences produced by the “JUMAN” morphological analyzer. We construct a word-based tagger using an ME model with word context features as a baseline for comparison with our character-based tagger.

4. Experiments

4.1. Training & Test Data

All training and test data were taken from manuscripts of NHK broadcast news. Sentences in both sets of data were partitioned
into words by “Cha-Sen” morphological analyzer. The training
data had 24,433 sentences (1.25M words and 2.0M characters).
There were 14,914 unique words observed more than two times
in the training data. The data had 2,500 unique characters ob-
served more than once. The test data had 996 sentences (48k
words and 77k characters). There were 2,525 NEs in total. The
remaining characters or words were treated as an unknown sym-
bol in the ME models.

4.2. Tagger and Models

We constructed the following taggers for experiments.

- **Baseline**: Word-based tagger using the ME model with
  word context features. A feature has one, two or three
  words in its left/right contexts.

- **Tagger1**: Character-based tagger using the ME model
  with character context features. A feature has one, two,
  three or four characters in its contexts.

- **Tagger2**: Character-based tagger using the ME model
  with the word context features. A word feature has one
  or two words in its contexts.

- **Tagger3**: Character-based tagger using the ME model
  with both character and word context features above.

In GIS training, frequencies of events in the training data
were simply discounted with a constant value of 0.1 to obtain
smoothed ME models. The number of iteration in GIS train-
ing was fixed at 200 times. All parameters associated with the
context length were determined by preliminary experiments.

4.3. F-measure Evaluation

Recall is defined as a measure of the numbers of correct NEs
from the correct documents and calculated by

\[
\text{Recall} : R = \frac{\text{Number of correctly marked up NEs}}{\text{Number of NEs in the reference}}
\]  
(7)

Precision is defined as a measure of the numbers of correct NEs
from all NEs and calculated by

\[
\text{Precision} : P = \frac{\text{Number of correctly marked up NEs}}{\text{Number of marked up NEs}}
\]  
(8)

F-measure is defined as a harmonic mean between recall and
precision:

\[
F = \frac{2PR}{P + R}
\]  
(9)

In this paper, F-measure is used as a criterion for results.

4.4. Results

4.4.1. Overall Results

Overall results of the extraction experiments are shown in Ta-
ble 2. **Tagger1**, which used the ME model with character
context features, did not achieve better performance than the
word-based tagger **Baseline**. It is clear that the features derived
from word contexts are more powerful than those from char-
acter contexts since the word contexts include information on
word boundaries explicitly. **Tagger1**, nevertheless, gave almost
the same recall score as **Baseline**. This result suggests that the
character context features tend to overgenerate NE candidates.
**Tagger3**, which used both character and word context features,
exceeded the performance of **Baseline**. Consequently, it is rea-
sonable for an efficient NE extraction technique to generate sur-
plus NE candidates using character context features while the
word context features suppress excess candidates.

Detailed results are shown in Table 3. **Baseline** gave the
best performance of F-measure on tagging the NEs in location
and artifact categories. In contrast, **Baseline** did not exceed
**Tagger1** in the performance where organization names were
tagged. It is possible that entity structures affect the difference
in performance. For example, the names in the location cate-
gory are typically expressed as concatenations of proper nouns
associated with prefectures, cities and streets. The word con-
text features naturally detect the linkages between these words
more powerfully than the character context features do. In con-
trast, names in the organization category, in particular company
names include many invented words and special notations.
For such entities, the character context features are more efficient
than the word context features since the ME models with the
character context features are designed to express how entities
are formed from characters. Hence, features that reflect the
characteristics of the entities are needed for efficient NE extrac-
tion.

4.4.2. Extraction of Unseen NEs

We compared the performance tagging unseen NEs. There were
a total of 624 unseen NEs (397 unique entities) in the test data.
Recall was regarded as a measure of the capability of the tag-
gers obtaining the unseen NEs since it is difficult to identify the
extracted NEs as the unseen NEs in the reference. Overall re-
results shown in Table 4 shows that the performance of **Tagger1**
was inferior to that of **Baseline** whereas character-based taggers
were expected to deal with the unseen NEs successfully because
of character-based entity tagging. Table 5 shows that all the
classifier-based taggers gave disappointingly poor performance
tagging the NEs in the artifact category. The degraded perform-
ances probably resulted from the length of NEs. The average
length of NEs included in the artifact category was 5.82 char-
acters per entity, which was longer than the average length of
NEs in all categories (4.82).

**Tagger3** improved the performance of **Tagger1** for tagging
the NEs in organization, location and artifact categories. The
result indicates that it is useful for the ME model to take word-
based and character-based features together for most categories.

4.4.3. Numbers of Features

Considering numbers of the features in the ME models, the
efficiency of the taggers is discussed. As shown in Table 2, **Baseline**
achieved comparably high performance with a smaller
number of features than others while **Tagger3** gave a small im-

<table>
<thead>
<tr>
<th># features</th>
<th>Precision(%)</th>
<th>Recall(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>335k</td>
<td>85.74</td>
</tr>
<tr>
<td>Tagger1</td>
<td>1.0M</td>
<td>80.06</td>
</tr>
<tr>
<td>Tagger2</td>
<td>345k</td>
<td>76.60</td>
</tr>
<tr>
<td>Tagger3</td>
<td>1.4M</td>
<td>86.42</td>
</tr>
</tbody>
</table>
Table 3: Extraction results (F-measure)

<table>
<thead>
<tr>
<th></th>
<th>org.</th>
<th>pers.</th>
<th>loc.</th>
<th>art.</th>
<th>date</th>
<th>time</th>
<th>money</th>
<th>prcnt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>77.74</td>
<td>83.20</td>
<td><strong>88.36</strong></td>
<td><strong>85.47</strong></td>
<td>92.51</td>
<td>56.90</td>
<td>81.16</td>
<td>87.38</td>
</tr>
<tr>
<td>Tagger1</td>
<td>82.80</td>
<td>79.05</td>
<td>80.44</td>
<td>79.08</td>
<td>92.86</td>
<td>65.98</td>
<td>86.96</td>
<td>89.00</td>
</tr>
<tr>
<td>Tagger2</td>
<td>74.38</td>
<td>78.46</td>
<td>80.65</td>
<td>78.62</td>
<td>92.86</td>
<td>65.31</td>
<td>81.16</td>
<td>81.16</td>
</tr>
<tr>
<td>Tagger3</td>
<td>83.17</td>
<td>87.14</td>
<td>87.60</td>
<td>84.14</td>
<td><strong>93.20</strong></td>
<td>68.04</td>
<td>84.06</td>
<td>87.56</td>
</tr>
</tbody>
</table>

Table 4: Overall extraction results (unseen NEs)

<table>
<thead>
<tr>
<th></th>
<th>Recall(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>38.78</td>
</tr>
<tr>
<td>Tagger1</td>
<td>34.29</td>
</tr>
<tr>
<td>Tagger2</td>
<td>35.26</td>
</tr>
<tr>
<td>Tagger3</td>
<td>45.03</td>
</tr>
</tbody>
</table>

Table 5: Extraction results (unseen NEs, recall)

<table>
<thead>
<tr>
<th></th>
<th>org.</th>
<th>pers.</th>
<th>loc.</th>
<th>art.</th>
<th>date</th>
<th>time</th>
<th>money</th>
<th>prcnt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>7.97</td>
<td>43.27</td>
<td>51.41</td>
<td><strong>14.06</strong></td>
<td>66.67</td>
<td>88.89</td>
<td>64.71</td>
<td>88.24</td>
</tr>
<tr>
<td>Tagger1</td>
<td>17.39</td>
<td>41.52</td>
<td>32.20</td>
<td>1.56</td>
<td>66.67</td>
<td>88.89</td>
<td>76.47</td>
<td>94.12</td>
</tr>
<tr>
<td>Tagger2</td>
<td>10.87</td>
<td>43.86</td>
<td>42.94</td>
<td>6.25</td>
<td>66.67</td>
<td>77.78</td>
<td>64.71</td>
<td>70.59</td>
</tr>
<tr>
<td>Tagger3</td>
<td><strong>20.29</strong></td>
<td><strong>56.14</strong></td>
<td><strong>57.06</strong></td>
<td>3.13</td>
<td>66.67</td>
<td>88.89</td>
<td>70.59</td>
<td><strong>100.00</strong></td>
</tr>
</tbody>
</table>

This suggests that the features of Tagger3 describe redundantly the ME model. Since the number of features has an impact on computational costs of GIS training, an efficient feature selection technique is required.

5. Conclusion

This paper described a method of NE extraction from Japanese broadcast news using maximum entropy modeling. We discussed efficiencies of both word-based and character-based taggers considering features of the ME models. The word-based tagger is powerful for NE extraction, while the character-based tagger is also useful for tagging the unseen NEs in several categories. The character-based tagger using the ME model with both word and character context features obtained the best performance for F-measure.

6. Acknowledgements

The authors would like to thank members of the Lehrstuhl für Informatik IV, at RWTH Aachen for their helpful suggestions and provision of computational equipment and tools.

7. References


