Word Pronunciation Disambiguation using the Web

Eiichiro Sumita[1, 2]
[1] NiCT
[2] ATR SLC
Kyoto 619-0288, JAPAN
eiichiro.sumita@atr.jp

Fumiaki Sugaya[3]
[3] KDDI R&D Labs

Saitama 356-8502, JAPAN
fsugaya@kddilabs.jp

Abstract

This paper proposes an automatic method of reading proper names with multiple pronunciations. First, the method obtains Web pages that include both the proper name and its pronunciation. Second, the method feeds them to the learner for classification. The current accuracy is around 90% for open data.

1 Introduction

Within text-to-speech programs, it is very important to deal with heteronyms, that is, words that are spelt the same but that have different readings, e.g. "bow" (a ribbon) and "bow" (of a ship). Reportedly, Japanese text-to-speech programs read sentences incorrectly more than 10 percent of the time. This problem is mainly caused by heteronyms and three studies have attempted to solve it (Yarowsky, 1996; Li and Takeuchi, 1997; and Umemura and Shimizu, 2000).

They assumed that the pronunciation of a word corresponded directly to the sense tag or part-of-speech of that word. In other words, sense tagging and part-of-speech tagging can determine the reading of a word. However, proper names have the same sense tag, for example, “location” for landmarks and the same part-of-speech, the “noun.” Clearly then, reading proper names is outside the scope of previous studies. Also, the proper names of locations, people, organizations, and others are dominant sources of heteronyms. Here, we focus on proper names. Our proposal is similar to previous studies in that both use machine learning. However, previous methods used expensive resources, e.g., a corpus in which words are manually tagged according to their pronunciation. Instead, we propose a method that automatically builds a pronunciation-tagged corpus using the Web as a source of training data for word pronunciation disambiguation.

This paper is arranged as follows. Section 2 proposes solutions, and Sections 3 and 4 report experimental results. We offer our discussion in Section 5 and conclusions in Section 6.

2 The Proposed Methods

It is crucial to correctly read proper names in open-domain text-to-speech programs, for example, applications that read Web pages or newspaper articles. To the best of our knowledge, no other studies have approached this problem. In this paper, we focus on the Japanese language. In this section, we first explain the Japanese writing system (Sections 2.1), followed by our proposal, the basic method (Section 2.2), and the improved method (Section 2.3).

2.1 The Japanese writing system

First, we should briefly explain the modern Japanese writing system. The Japanese language is represented by three scripts:

[i] Kanji, which are characters of Chinese origin;
[ii] Hiragana, a syllabary (reading); and
[iii] Katakana, also a syllabary (reading).

Script	Sample
KANJI	大平
HIRAGANA (reading)	おおだいら
KATAKANA (reading)	オオダイラ

Table 1 Three writings of a single word

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 165–168,
New York, June 2006. ©2006 Association for Computational Linguistics

As exemplified in **Table 1**, there are three writings for the word "大平." The lower two samples are representations of the same pronunciation of "oo daira."

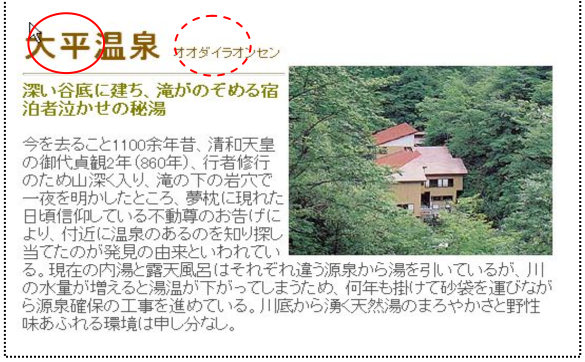
Listing possible readings can be done by consulting a dictionary (see Section 3.1 for the experiment). Therefore, in this paper, we assume that listing is performed prior to disambiguation.

2.2 The basic method based on page hits

The idea is based on the observation that proper names in Kanji often co-occur with their pronunciation in Hiragana (or Katakana) within a single Web page, as shown **Figure 1**. In the figure, the name "大平" in Kanji is indicated with an oval, and its pronunciation in Katakana, "オオダイラ," is high-lighted with the dotted oval.

According to Google, there are 464 pages in which "大平" and "オオダイラ" co-occur.

In this sense, the co-occurrence frequency suggests to us the most common pronunciation.



大平温泉 オオダイラオンセン

深い谷底に建ち、滝がのぞめる宿泊者泣かせの秘湯

今を去ること1100余年昔、清和天皇の御代貞観2年(860年)、行者修行のため山深く入り、滝の下の岩穴で一夜を明かしたところ、夢枕に現れた日頃信仰している不動尊のお告げにより、付近に温泉のあるのを知り探し当てたのが発見の由来といわれている。現在の内湯と露天風呂はそれぞれ違う源泉から湯を引いているが、川の水量が増えると湯温が下がってしまうため、何年も掛けて砂袋を運びながら源泉確保の工事を進めている。川底から湧く天然湯のまろやかさと野性味あふれる環境は申し分なし。

Figure 1 On the Web, words written in Kanji often co-occur with the pronunciation written in Katakana [1]

Our simple proposal to pick up the most frequent pronunciation achieves surprisingly high accuracy for open data, as Section 4 will later show.

2.3 The improved method using a classifier

The basic method mentioned above merely selects the most frequent pronunciation and neglects all others. This is not disambiguation at all.

The improved method is similar to standard word-sense disambiguation. The hit pages can provide us with training data for reading a particular word. We feed the downloaded data into the learner of a classifier. We do not stick to a certain method of machine learning; any state-of-the-art method will work. The features used in classification will be explained in the latter half of this subsection.

Collecting training data from the Web

Our input is a particular word, W, and the set of its readings, $\{R_k \mid k=1\sim K\}$.

```
For all k =1~K:
  i)   search the Web using the query "W AND $R_k$."
  ii)  obtain the set of snippets, $\{S_l(W, R_k) | l=1\sim L\}$.
  iii) separate $R_k$ from $S_l$ and obtain the set of training data, $\{(T_l(W), R_k) | l=1\sim L\}$.
  end
```

In the experiments for this report, L is set to 1,000. Thus, for each reading R_k of W, we have, at most 1,000 training data $T_l(W)$.

Training the classifier

From the training data $T_l(W)$, we make feature vectors that are fed into the learner of the decision tree with the correct reading R_k for the word in question, W.

Here, we write $T_l(W)$ as $W_{-m}\ W_{-(m-1)}\ \ldots\ W_{-2}\ W_{-1}\ W\ W_1\ W_2\ \ldots\ W_{m-1}\ W_m$, where m is from 2 to M, which hereafter is called the window size.

We use two kinds of features:

- The part-of-speech of $W_{-2}\ W_{-1}$ and $W_1\ W_2$
- Keywords within the snippet. In this experiment, keywords are defined as the top N frequent words, but for W in the bag consisting of all words in $\{T_l(W)\}$.

In this paper, N is set to 100. These features ground the pronunciation disambiguation task to the real world through the Web. In other words, they give us knowledge about the problem at hand, i.e., how to read proper names in a real-world context.

3 Experimental Data

We conducted the experiments using proper location names.

[1] http://oyudokoro.mimo.com/area/C/cd/tng/000370/index.html

3.1 Ambiguous name lists

Japan Post openly provides postal address lists associated with pronunciations .

From that list, we extracted 79,861 pairs of proper location names and their pronunciations. As the breakdown of **Table 2** shows, 5.7% of proper location names have multiple pronunciations, while 94.3% have a single pronunciation. The average ambiguity is 2.26 for ambiguous types. Next, we took into consideration the frequency of each proper name on the Web. Frequency is surrogated by the page count when the query of a word itself is searched for using a search engine. About one quarter of the occurrences were found to be ambiguous.

Number of readings	type	%
1	70,232	94.3
2	3,443	**5.7**
3	599	
4	150	
5	45	
6	11	
7	4	
8	2	
11	1	
total	74,487	100.0

Table 2 Pronunciation ambiguities in Japanese location names

Our proposal depends on co-occurrences on a Web page. If the pairing of a word W and its reading R do not occur on the Web, the proposal will not work. We checked this, and found that there was only one pair missing out of the 79,861 on our list. In this sense, the coverage is almost 100%.

3.2 Open Data

We tested the performance of our proposed methods on openly available data.

Open data were obtained from the EDR corpus, which consists of sentences from Japanese newspapers. Every word is tagged with part-of-speech and pronunciation.

We extracted sentences that include location heteronyms, that is, those that contain Kanji that can be found in the above-mentioned list of location heteronyms within the postal address data. There were 268 occurrences in total. There were 72 types of heteronyms.

4 Experiment Results

We conducted two experiments: (1) an open test; and (2) a study on the degree of ambiguity.

4.1 Open test

We evaluated our proposals, i.e., the basic method and the improved method with the open data explained in Section 3.1. Both methods achieved a high rate of accuracy.

Basic method performance

In the basic method, the most common pronunciation on the Web is selected. The frequency is estimated by the page count of the query for the pairing of the word W and its pronunciation, R_i.

There are two variations based on the Hiragana and Katakana pronunciation scripts. The average accuracy for the open data was 89.2% for Hiragana and 86.6% for Katakana (**Table 3**). These results are very high, suggesting a strong bias of pronunciation distribution in the open data.

Scripts	Accuracy
HIRAGANA	89.2
KATAKANA	86.6

Table 3 Open test accuracy for the basic method

Performance of the improved method

Table 4 shows the average results for all 268 occurrences. The accuracy of the basic method (**Table 3**) was lower than that of our improved proposal in all window sizes, and it was outperformed at a window size of ten by about 3.5% for both Hiragana and Katakana.

Script	M=2	M=5	M=10
HIRAGANA	89.9	90.3	**92.9**
KATAKANA	89.2	88.4	**89.9**

Table 4 Open test accuracy for the improved method

4.2 Degree of ambiguity

Here, we examine the relationship between the degree of pronunciation ambiguity and pronunciation accuracy using a cross-validation test for training data[2] for the improved method with Hiragana.

Average case

We conducted the first experiment with twenty words [3] that were selected randomly from the Ambiguous Name List (Section 3.1). The average ambiguity was 2.1, indicating the average performance of the improved proposal.

Class	M=2	M=5	M=10	basic
2.1	**89.2 %**	**90.9 %**	**92.3 %**	67.5%

Table 5 Average cases

Table 5 summarizes the ten-fold cross validation, where M in the table is the training data size (window size). The accuracy changes word by word, though the average was high about 90% of the time.

The "basic" column shows the average accuracy of the basic method, i.e., the percentage for the most frequent pronunciation. The improved method achieves much better accuracy than the "basic" one.

The most ambiguous case

Next, we obtained the results (**Table 6**) for the most ambiguous cases, where the degree of ambiguity ranged from six to eleven[4]. The average ambiguity was 7.1.

Class	M=2	M=5	M=10	basic
7.1	**73.9 %**	**77.3 %**	**79.9 %**	57.5%

Table 6 Most ambiguous cases

As we expected, the performances were poorer than the average cases outlined above, although they were still high, i.e., the average ranged from about 70% to about 80 %. Again, the improved method achieved much better accuracy than the "basic" method. [5]

5 Discussion on Transliteration

Transliteration (Knight and Graehl, 1998) is a mapping from one system of writing into another, automation of which has been actively studied between English and other languages such as Arabic, Chinese, Korean, Thai, and Japanese. If there are multiple translation candidates, by incorporating context in a way similar to our proposal, one will be able to disambiguate them.

6 Conclusion

This paper proposed a new method for reading proper names. In our proposed method, using Web pages containing Kanji and Hiragana (or Katakana) representations of the same proper names, we can learn how to read proper names with multiple readings via a state-of-the-art machine learner. Thus, the proposed process requires no human intervention. The current accuracy was around 90% for open data.

References

K. Knight and J. Graehl. 1998 Machine transliteration. Computational Linguistics, 24(4):599-612.

H. Li and J. Takeuchi. 1997. Using Evidence that is both string and Reliable in Japanese Homograph Disambiguation, SIGNL119-9, IPSJ.

Y. Umemura and T. Shimizu. 2000. Japanese homograph disambiguation for speech synthesizers, Toyota Chuo Kenkyujo R&D Review, 35(1):67-74.

D. Yarowsky. 1996. Homograph Disambiguation in Speech Synthesis. In J. van Santen, R. Sproat, J. Olive and J. Hirschberg (eds.), Progress in Speech Synthesis. Springer-Verlag, pp. 159-175.

[2] There is some question as to whether the training data correctly catch all the pronunciations. The experiments in this subsection are independent of this problem, because our intention is to compare the performance of the average case and the most ambiguous case.

[3] 東浜町, 三角町, 宮丸町, 川戸 ,下坂田, 蓬田, 金沢町, 白木町, 神保町, 助谷, 新御堂, 糸原, 駿河町, 百目木, 垣内田町, 杉山町, 百戸, 宝山町, 出来島, 神楽町.

[4] 小谷, 上原町, 上原, 小原, 西原, 上町, 大平, 葛原, 平田, 馬場町, 新田, 土橋町, 大畑町, 上野町, 八幡町, 柚木町, 長田町, 平原.

[5] For some words, the basic accuracy is higher than the cross validation accuracy because the basic method reaches all occurrences on the Web thanks to the search engine, while our improved method limits the number of training data by L in Section 2.3. For example, the most frequent pronunciation of "上原" has 93.7% on the Web, whereas the distribution in the training data is different from such a sharp distribution due to the limitation of L.