Long Vowel Detection for Letter-to-sound Conversion for Japanese Sourced Words Transliterated into the Alphabet

Hisako Asano, Hideharu Nakajima, Hideyuki Mizuno, Masahiro Oku

NTT Cyber Space Laboratories, NTT Corporation
{asano.hisako, nakajima.hideharu, mizuno.hideyuki, oku.masahiro}@lab.ntt.co.jp

Abstract
Modern Japanese texts often include Western sourced words written in the Roman alphabet. Even Japanese sourced words are sometimes transliterated into the Roman alphabet. As most of them are very new and idiosyncratic proper nouns, it is impractical to assume all those alphabetic words can be registered in the word dictionary of a text-to-speech system; their pronunciation must be derived automatically. As long vowel expressions are the same with short vowel or diphthong expressions in Japanese words transliterated into the alphabet, long vowel detection is necessary for generating highly accurate pronunciation. This paper proposes a method to detect long vowel positions using the Support Vector Machine. The cost of making the learning model for the detection is very low because the training data can be generated automatically from the text-to-speech word dictionary. 93.2% word accuracy was achieved which is almost the maximum accuracy possible from just spelling information.

1. Introduction
Japanese is a non-Roman alphabet language, but modern Japanese texts often include Western sourced words written in the Roman alphabet (e.g., “OUTLET 横浜”), rather than in Japanese phonograms for loan words (i.e. Katakana). Even Japanese sourced words are sometimes transliterated into the Roman alphabet for style reasons, such as “イタリア料理店 Ristrante Tokyo” (means “the Italian restaurant, Ristrante Tokyo”; “Tokyo” is a Japanese place name) to give it an Italian taste. For example, a shopping directory in a popular regional web portal covering major cities in Japan, lists more than 8,000 shops, and includes a total of 6,400 alphabetic words (2,900 distinctive words). As most of them are very new and idiosyncratic proper nouns, it is impractical to assume that all those alphabetic words will be registered in the word dictionary of a text-to-speech (TTS) system, so their pronunciations must be derived automatically.

These alphabetic words consist of various language sourced words (e.g., “OUTLET”: English, “Ristrante”: Italian, “Tokyo”: Japanese). Our approach for letter-to-sound conversion of unknown alphabetic words consists of two steps; the first step classifies an unknown word into a say-as class [1], which means the class of the letter-to-sound conversion (e.g., English, spell-out or Japanese) and the second step applies the letter-to-sound conversion for the classified say-as class. This paper focuses on the letter-to-sound conversion for alphabetic transliteration of Japanese, which composes a part of the second step.

Alphabetic transliteration of Japanese generally represents the pronunciation itself but there are no letters to directly represent long vowels; long vowels are omitted or replaced with diphthongs. There is a very common letter-to-sound conversion method using a letter-to-sound mapping table. However, it can not generate long vowel pronunciation at all because it can not distinguish long vowel expressions from short vowel or diphthong expressions. To solve this problem, this paper proposes a new letter-to-sound conversion method for the alphabetic transliteration of Japanese using the Support Vector Machine (SVM) [2].

2. Letter-to-sound conversion for alphabetic transliteration of Japanese

2.1. Problems
Alphabetic transliteration of Japanese represents the pronunciation itself and there is no ambiguity in letter-to-sound conversion except for two exceptions.

The first exception is long vowels. As there are no letters to directly represent long vowels, there are two kinds of indirect notations for long vowels;

• Long vowel omission: A long vowel is omitted and it becomes the same expression with short vowel (e.g., long vowel /u:/ is represented as “u” and short vowel /u/ is also represented as “ui”).

• Replacement with diphthong: A long vowel is replaced with a diphthong (e.g., /u/ is replaced with “uu”).

The pronunciation /yu:ko:/ (a Japanese person name) is written as “Yuko” (long vowel omission) or “Yuuko” (replacement with diphthong) in the alphabetic transliteration. Long vowel omission is generally preferred to replacement with diphthong, but the pronunciation /ke:ko/ (a Japanese person name) is transliterated as “Keiko”; “Keko” is impossible. The two types of notation are sometimes mixed in a word (e.g., “Keio”: “ei” = replacement with diphthong, “o” = long vowel omission, pronunciation = /ke:ko:/).

Because long vowels are frequent in Japanese, it is necessary to detect long vowel positions to improve pronunciation accuracy. In the notation of long vowel omission, each short vowel expression may represent a long vowel pronunciation. It is necessary to detect the long vowel positions in short vowel expressions and add the long vowel marks /:/, but this detection is very difficult and no method

\[1\] The mapping table can be found on http://homepage1.nifty.com/shorinji/gloss/jchar.htm.

\[2\] :/ indicates a long vowel in this paper.
exists. In the notation of replacement with diphthong, diphthong expressions (“aa”, “ii”, “uu”, “ee”, “oo”, “ci” and “ou”) are the candidates for replacement with long vowel. However, the diphthong expression is just diphthong pronunciation if it lies across the boundary of Kanji character3 (Japanese ideogram). For example, the Kanji expression of the alphabetic transliteration “Tateisi” (a Japanese person name) is “ち." (tate: /ti/ (is)) and the pronunciation is not /tate:si/ but /tate isi/. It is necessary to detect the long vowel positions in diphthong expressions and replace diphthongs with long vowels without Kanji information because there is no Kanji information in alphabetic transliterations.

The second exception is /N/ or /n/ ambiguity. The pronunciation of “n + a vowel” is ambiguous between “a mora nasal /N/ + a vowel mora" (two morae) and “a mora beginning with a consonant /n/” (one mora). For example, the pronunciation of alphabetic transliteration “kano” is ambiguous between /ka no/ and /ka no/. However, “a mora beginning with a consonant /n/” (/ka no/) is common in Japanese nouns, so we do not treat this problem in this paper.

### 2.2. Approach

Our approach to the long vowel problem is to classify each mora in Japanese words written alphabetically into three long vowel classes: long vowel addition, long vowel replacement, or no change. A mora is the minimum unit of long vowel addition for the notation of long vowel omission and long vowel replacement for the notation of replacement with diphthong. The classification uses only spelling information because unknown alphabetic words have no other information. Statistical methods are fit for this problem, since general Japanese TTS word dictionaries have word pronunciations with long vowel information and alphabetic transliterations can be easily generated from them; training data can be made easily. We use SVM as the statistical classifier because it can handle a large number of feature sets without suffering the sparse data problem. SVM was recently applied to variety of NLP tasks and was shown to be very effective.

#### 2.3. Flow of letter-to-sound conversion

Figure 1 shows the flow of letter-to-sound conversion for Japanese sourced words transliterated into the alphabet.

First, letters are converted into sounds and divided into morae using a letter-to-sound mapping table (e.g., “kyo” = /k/ “chi” and “ti” = /i/). This mapping table is very general and the conversion is easy. The “n + a vowel” expression is treated as “a mora beginning with a consonant /n/” because this pronunciation is common.

Next, each mora is classified into a long vowel class. This is our new proposal.

Finally, the pronunciation is modified by the classification result and the final pronunciation is generated.

#### 3. Long vowel classification using SVM

As spelling information, we consider the three SVM features shown below.

- **Mora:** The mora sound representation itself.
- **Consonant:** The consonant in the mora sound representation. The dummy value /a/ is given for morae consisting of just a vowel (/a/, /i/, /u/, /e/ or /o/).
- **Vowel:** The vowel in the mora sound representation. The checked sound /Q/ and mora nasal /N/ have no vowel, so dummy values /Q/ and /N/ are given, respectively.

For example, in the mora /si/, the mora feature is /si/, the consonant feature is /s/ and the vowel feature is /i/. The best feature set for long vowel classification is investigated in Section 4.3.2.

The context is the mora range using their features for classification. We consider the classification target mora and the n (n=1, 2, 3) prior and following morae as the context. The best context length n is investigated in Section 4.3.3.

Classification accuracy generally depends on the size of training data. The size of training data needed is confirmed in Section 4.3.4.

#### 4. Experiments

We used YamCha6 in our experiments, which is a general purpose SVM classifier tool. In all experiments, the SVM parameters for learning were second degree of polynomial kernel and 1 slack variable, and the pair-wise method was used for multi-class extension.

### 4.1. Corpora

#### 4.1.1. Person name data

In Japanese texts, most Japanese words transliterated into the alphabet are proper nouns. Therefore, we extracted 26,523 Japanese person name entries from the word dictionary for TTS system. As the entries have the pronunciation with long vowel information, the alphabetic transliteration was automatically generated from the pronunciation. All spelling variations were expanded. For example, the long vowel was written in long vowel omission and replacement with diphthong (e.g., the pronunciation /o:yama/ was transliterated

---

3 All Japanese sourced nouns can be written in Kanji characters.

4 A mora usually consists of a consonant and a vowel (e.g., /ka/) or only a vowel (e.g., /a/). Special morae are choked sound /Q/ and mora nasal /N/. A long vowel is counted as two morae.

5 Space ‘ ‘ represents a mora boundary in this sentence pronunciation examples.

6 http://cl.aist-nara.ac.jp/~taku-ku/software/yamcha
as “oyama” and “ooyama”). This yielded 49,031 alphabetic words.

4.1.2. Web data

As the real data, we collected 254 distinctive Japanese words written in the alphabet from a shopping directory site on the web.

4.2. Baseline method and evaluation criteria

4.2.1. Long vowel addition

The baseline method of long vowel addition is no long vowel addition for short vowel expressions. This yields errors when short vowel expressions represent long vowel pronunciations.

The evaluation criterion for long vowel addition is:

$$\text{Addition precision rate} = \frac{\text{# of correct pronunciation}}{\text{# of short vowel expression}}$$

(1)

4.2.2. Long vowel replacement

Most diphthong expressions “aa” and “ee” have diphthong pronunciation, and the majority of diphthong expressions “ii”, “uu”, “oo”, “ei” and “ou” have long vowel pronunciation in the person name data generated in 4.1.1. Therefore, the baseline method of long vowel replacement is that the diphthong expressions “ii”, “uu”, “oo”, “ei” and “ou” are replaced with long vowels. This yields errors when “aa” and “ee” do not lie across the boundary of Kanji character or “ii”, “uu”, “oo”, “ei” and “ou” lie across the boundary of Kanji character.

The evaluation criterion for long vowel replacement is:

$$\text{Replacement precision rate} = \frac{\text{# of correct pronunciation}}{\text{# of diphthong expression}}$$

(2)

4.2.3. Word precision

The word precision is used for the total evaluation criterion:

$$\text{Word precision rate} = \frac{\text{# of correct pronunciation word}}{\text{# of word}}$$

(3)

4.3. Trial for identifying optimal parameter set

We performed several experiments using the person name data shown in 4.1.1 to identify the optimal parameter set for long vowel classification. The person name data was divided into ten equal sets. The experiments in Section 4.3.2 and 4.3.3 used the tenfold cross-validation method (training data : test data = 9 : 1) and the evaluation criteria were the average precision rates of the criteria in Section 4.2. The experiment for training data size in Section 4.3.4 did not use cross-validation and used only one data set (maximum training data : test data = 9 : 1). Finally the errors in the optimal parameter set were analyzed.

4.3.1. Baseline method

Table 1 shows the average precision rates of the baseline method described in Section 4.2 for the cross-validation test data.

<table>
<thead>
<tr>
<th>Feature set</th>
<th>Addition</th>
<th>Replacement</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mora, Consonant, Vowel</td>
<td>97.7%</td>
<td>98.2%</td>
<td>91.9%</td>
</tr>
<tr>
<td>Mora</td>
<td>98.0%</td>
<td>98.2%</td>
<td>92.7%</td>
</tr>
<tr>
<td>Consonant, Vowel</td>
<td>98.2%</td>
<td>98.2%</td>
<td>93.2%</td>
</tr>
</tbody>
</table>

4.3.2. Feature set

Table 2 shows the average precision rate of cross-validation for the three feature sets shown below. In this experiment, the context length was $n=2$.

- Mora, Consonant and Vowel
- Mora
- Consonant and Vowel

‘Consonant and Vowel’ feature set is best for long vowel addition. The reasons are (1) ‘Consonant and Vowel’ is more robust than ‘Mora’ because ‘Mora’ is divided into ‘Consonant and Vowel’ and the data coverage of ‘Consonant and Vowel’ is wider than ‘Mora’, (2) the vowel feature is important for long vowel addition (e.g., morae with vowel /a/ are almost no change, while morae with vowel /o/ are comparatively high rate of long vowel addition).

On the other hand, there is no feature set difference in long vowel replacement. Long vowel replacement only occurs in diphthong expressions and the important thing is to find the exceptional conditions of no long vowel replacement. Probably the conditions are not complicated and either of ‘Mora’ or ‘Consonant and Vowel’ can cover the conditions.

<table>
<thead>
<tr>
<th>Feature set</th>
<th>Addition</th>
<th>Replacement</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mora, Consonant, Vowel</td>
<td>97.7%</td>
<td>98.2%</td>
<td>91.9%</td>
</tr>
<tr>
<td>Mora</td>
<td>98.0%</td>
<td>98.2%</td>
<td>92.7%</td>
</tr>
<tr>
<td>Consonant, Vowel</td>
<td>98.2%</td>
<td>98.2%</td>
<td>93.2%</td>
</tr>
</tbody>
</table>

4.3.3. Context length

Table 3 shows the average precision rate of cross-validation for context length $n = 1, 2, 3$. ‘Consonant, Vowel’ feature set (the best feature set) was used in this experiment.

Context length $n = 2$ is the best. It indicates that the context length corresponding to one Kanji character is enough for long vowel classification because almost all Kanji characters have one or two mora pronunciation.

<table>
<thead>
<tr>
<th>Context length</th>
<th>Addition</th>
<th>Replacement</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96.7%</td>
<td>92.6%</td>
<td>88.1%</td>
</tr>
<tr>
<td>2</td>
<td>98.2%</td>
<td>98.2%</td>
<td>93.2%</td>
</tr>
<tr>
<td>3</td>
<td>97.5%</td>
<td>95.5%</td>
<td>91.0%</td>
</tr>
</tbody>
</table>

4.3.4. Training data size

Figure 2 shows the precision rate versus the ratio of training data size. Ratio 1 means 44,128 words are used for training while ratio 1/729 indicates 61 words.

The precision rate increases with the training data size in both of long vowel addition and replacement. The full size (ratio 1) prepared for this experiment offers the best accuracy,
and there is some possibility of increasing precision rate by adding more training data.

![Learning curve for long vowel classification](image)

**Figure 2: Learning curve for long vowel classification**

### 4.3.5. Optimal parameter set

Through these experiments, the optimal parameter set is identified as follows:

- **Feature set**: Consonant and Vowel
- **Context length**: $n = 2$
- **Training data size**: About 44,000 words

This set archived average word precision of 93.2%. The baseline method of precision was 80.3%, so the error reduction was 65%.

### 4.4. Error analysis

349 erroneous words in one test data of cross-validation with the optimal parameter set were analyzed. The distribution of error words is shown in Table 4, which was classified by pronunciation ambiguity. Japanese alphabetic words with long vowel omission and short vowel sometimes have the same spelling. For example, the spelling “oyama” is ambiguous between /o-ryama/ and /oyama/ and the correct pronunciation can not be decided from just the spelling because Japanese person names with both pronunciations exist. If the word pronunciation generated by our method exists as a Japanese person name such as /oyama/, it is classified as ‘Word ambiguity’. If the pronunciation does not exist as a Japanese person name but the pronunciation is ambiguous in the context (five morae: the target mora and the two prior and following morae), which means more than three morae are same in the context, it is classified as ‘Partial ambiguity’. For example, the generated pronunciation of “okusu” is /okusu/ (correct: /o-kusu/), and there is no Japanese name with pronunciation /okusu/. However, spelling “okusumi” with pronunciation /okusumi/ exists in the training data. The context for the first mora “o” (-, -, o, ku, su) is the same between “okusu” and “okusumi”. Other cases are classified as ‘No ambiguity’. ‘/N/ or /n/’ in Table 4 means the errors for ‘/N/ or /n/’ ambiguity shown in Section 2.1.

74% of errors are due to word pronunciation ambiguity and 15% to partial pronunciation ambiguity. It means our method achieves almost the maximum accuracy possible with just spelling information. Background knowledge (e.g., Who is Mr. Oyama?) is needed to distinguish the word pronunciation ambiguity.

The ‘/N/ or /n/’ ambiguity errors are few and most errors occur in particular expressions. The false conversion of /nitii/ to /niti/ was more than half of these errors (e.g., The pronunciation of “Junichiro” is not /junitii/ but /juNitii:/). Some exceptional mapping rules (e.g. “nitii” = /Niti/) can improve the accuracy.

### Table 4: Word error distribution (# of words)

<table>
<thead>
<tr>
<th>Pronunciation</th>
<th>Addition</th>
<th>Replacement</th>
<th>/N/ or /n/</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word ambiguity</td>
<td>70.8%</td>
<td>2.9%</td>
<td>0.6%</td>
</tr>
<tr>
<td>Partial ambiguity</td>
<td>14.6%</td>
<td>0.9%</td>
<td>0%</td>
</tr>
<tr>
<td>Word</td>
<td>3.4%</td>
<td>1.1%</td>
<td>5.7%</td>
</tr>
</tbody>
</table>

### 4.5. Evaluation using Web data

Table 5 shows the precision rate with the optimal parameter set for Web data described in Section 4.1.2.

The SVM method achieves almost equal accuracy with person name data. This is not a strict evaluation because Web data is a small corpus and it has a few long vowel pronunciations. However, we consider that the learning model generated from person name data is effective for web data, since it includes many Japanese proper nouns transliterated into the alphabet.

### Table 5: Precision rate for Web data

<table>
<thead>
<tr>
<th>Method</th>
<th>Addition</th>
<th>Replacement</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>94.1%</td>
<td>87.5%</td>
<td>87.4%</td>
</tr>
<tr>
<td>(Proposal)</td>
<td>96.4%</td>
<td>87.5%</td>
<td>92.5%</td>
</tr>
</tbody>
</table>

### 5. Conclusions

We proposed a new method that can detect long vowel position in Japanese words transliterated into the alphabet using SVM. The cost of making the learning model for the detection is very low because the training data can be generated automatically from the TTS word dictionary. It was confirmed that it archived almost the maximum pronunciation accuracy possible from just spelling information.

### 6. References


---

712 words need long vowel replacement (7 of them are diphthong “ei”, which can not be expressed as long vowel omission) and 30 words need long vowel addition in Web data.