KANJI-TO-HIRAGANA CONVERSION BASED ON A LANGUAGE MODEL

Wei-Bin Chang

Philips Research East Asia – Taipei
24F, 66, Sec. 1, Chung Hsiao W. Rd., Taipei 100, Taiwan
weibin.chang@philips.com

ABSTRACT

In speech recognition systems, a common problem is transcription of new additions to the recognition lexicon into their phonetic symbols. Specific to the Japanese language, such a problem can be dealt with in two steps. In this paper, we focus on the first step, in which the new lexical entry is converted into a set of hiragana syllabaries, which is almost a phonetic transcription. We propose a conversion scheme which yields the most likely hiragana syllabaries, based on a language model. Results from our evaluations on three test sets are also reported. Although the study is conducted on Japanese only, our approach has applications to Chinese.

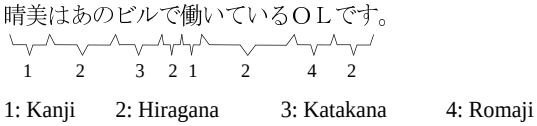
1. INTRODUCTION

Most continuous speech recognition systems require phonetic transcriptions of words in their recognition lexica. It is desirable to have a system that can automatically generate phonetic transcriptions of lexical entries. Since not every user is good at working with phonetic symbols, such a feature is even more important for commercial speech recognition products, where users constantly have the need to add new words to the recognition lexica. This paper addresses one facet of the phonetic transcription problem that is specific to the Japanese language.

1.1. Japanese Orthography

Conventional Japanese orthography is a mixture of four types of character, namely, *kanji*, *hiragana*, *katakana*, and *romaji*. An example of Japanese orthography is shown in Fig. 1(a). Kanji are ideographic characters, borrowed from the Chinese writing system, which in most cases have meanings. Hiragana and katakana, collectively referred to as *kana*, are themselves syllabaries which represent sounds but no meanings. Romaji, consisting of the Roman alphabet, are mostly used to reproduce words of Western languages. Unlike English, Japanese does not use white spaces to delineate words in written text and so the definition of a word in Japanese is ambiguous. Therefore, in the following, we shall refer to the basic unit of a lexicon as a lexical entry, instead of a word, to avoid controversies.

(a) Conventional Japanese orthography:



(b) Hiragana transliteration:

はるみはあのびるではたらいているおおえるです。
晴　美はあのビルで働　いているＯ　Ｌ　です。

Fig. 1. Example showing (a) a text in conventional Japanese orthography and (b) its transliteration in hiragana syllabaries. The sentence means “Harumi is an office lady working in that building.” Note that the reading of a kanji character may consist of more than one hiragana syllabary.

In Japanese, written text can be transliterated into kana syllabaries (in this paper we use hiragana) without loss of information about its reading. In other words, kana provide an almost phonetic transcription of the text. As an example, the hiragana transliteration of the sentence in Fig. 1(a) is shown in Fig. 1(b). In this view, phonetic transcription of a Japanese lexical entry can be dealt with in two steps. In the first step, the lexical entry is converted into kana characters representing its reading. Then the kana-based reading is mapped to phonetic symbols using some mapping rules in the second step. This two-step approach has the following advantages. First, many Japanese dictionaries provide kana readings and hence knowledge sources are abundant. Second, pronunciation variations, due to factors such as accents, co-articulation, and so on, are more easily tackled at the kana level since they are syllabaries.

1.2. Kanji-to-hiragana Conversion

In this paper, we focus on the problem of converting new kanji lexical entries into their hiragana-based readings. Conversion of romaji entries is not considered here since it involves knowledge of not only Japanese but also the languages from where they originate. Kanji-to-hiragana conversion has been addressed in [7]; however, this paper is focused on converting a full sentence into its hiragana-like translation and only considers a very restricted set of kanji characters. We are not aware of any other literature written in English that addresses similar problems.

Several distinguishing features of Japanese make kanji-to-hiragana conversion a challenging problem. First of all, nearly all kanji characters have readings in two categories, the native Japanese reading (known as *kun*) and the one with a Chinese origin (known as *on*) [6]. Moreover, due to constant evolution of the Japanese language, many kanji have more than one kun-reading and also more than one on-reading. The correct reading of a kanji character can only be determined by examining the context in which it appears. An additional complication is that about six thousand kanji characters are used in Japanese, which are two orders of magnitude larger than the size of the Roman alphabet.

We propose a statistical approach to kanji-to-hiragana conversion, which is based on a language model. We first formulate this problem in mathematical terms in Section 2. Section 3 explains the conversion system we propose. Results from our evaluations are presented in Section 4. Finally, we summarize our findings and outline the future work in Section 5.

2. MATHEMATICAL FORMULATION

Taking a statistical approach, we formulate the problem of kanji-to-hiragana conversion as follows. Given a kanji lexical entry consisting of n kanji characters $k_1,\dots,k_n$, the converter outputs $r_1^*,\dots,r_n^*$ which are the solution to the following equation

$$\begin{aligned} r_1^*,\dots,r_n^* &= \arg\max_{r_1,\dots,r_n} Pr(R_1=r_1,\dots,R_n=r_n| \\ &\qquad K_1=k_1,\dots,K_n=k_n), \end{aligned}$$

where r_i is a hiragana-based reading for the ith kanji character k_i. We emphasize here that each r_i may consist of more than one hiragana character (cf. Fig. 1(b)). Using the formula for conditional probabilities and dropping the term independent of $r_1,\dots,r_n$, we can rewrite the above equation as

$$\begin{aligned} r_1^*,\dots,r_n^* &= \arg\max_{r_1,\dots,r_n} Pr(K_1=k_1,\dots,K_n=k_n, \\ &\qquad R_1=r_1,\dots,R_n=r_n). \end{aligned}$$

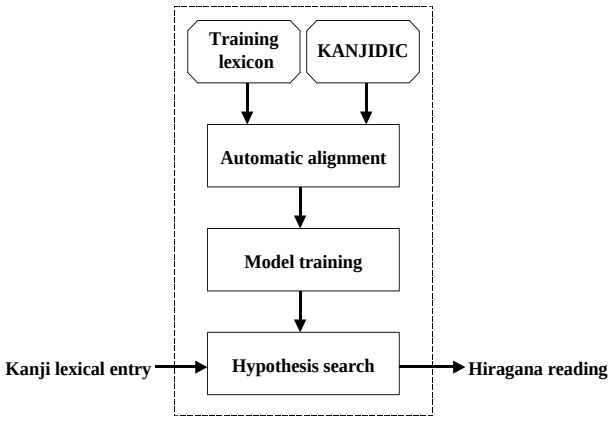


Fig. 2. Architecture of the kanji-to-kana conversion system.

If one applied the analogy to speech recognition problems, one would write the probability in the last equation as

$$\begin{aligned} &Pr(K_1=k_1,\dots,K_n=k_n,R_1=r_1,\dots,R_n=r_n) \\ &\quad = Pr(K_1=k_1,\dots,K_n=k_n|R_1=r_1,\dots,R_n=r_n) \\ &\qquad \times Pr(R_1=r_1,\dots,R_n=r_n). \end{aligned}$$

However, the probability $Pr(K_1,\dots,K_n|R_1,\dots,R_n)$ is difficult to work with. Instead, we write

$$\begin{aligned} &Pr(K_1=k_1,\dots,K_n=k_n,R_1=r_1,\dots,R_n=r_n) \\ &\quad = Pr\big((K_1,R_1)=(k_1,r_1),\dots,(K_n,R_n)=(k_n,r_n)\big) \\ &\quad =: p\big((k_1,r_1),\dots,(k_n,r_n)\big) \end{aligned} \tag{1}$$

in order to circumvent the difficulty. Note that the term $p\big((k_1,r_1),\dots,(k_n,r_n)\big)$ can be thought of as the language model for a "language" whose alphabet consists of kanji-reading pairs (k_i,r_i). Thus, the kanji-to-hiragana converter should decide in favor of the sequence of readings $r_1^*,\dots,r_n^*$ satisfying

$$r_1^*,\dots,r_n^* = \arg\max_{r_1,\dots,r_n} p\big((k_1,r_1),\dots,(k_n,r_n)\big). \tag{2}$$

3. SYSTEM DESIGN

The conversion system we are proposing is based on the optimization equation (2). The system, which is shown in Fig. 2, consists of the following three modules:

1. Automatic alignment;
2. Model training;
3. Hypothesis search.

3.1. Automatic Alignment

The language model (1) can be trained on data "written" in the alphabet of kanji-reading pairs (k_i, r_i). Such training data can be created from a lexicon by aligning each kanji character in an entry with its corresponding hiragana reading. To illustrate, suppose the full hiragana-based reading of a lexical entry with n kanji characters $k_1, \ldots, k_n$, consists of m hiragana characters $h_1, \ldots, h_m$. We consistently associate each k_i with the reading $r_i = (h_{i_1}, \ldots, h_{i_l})$ such that $(r_1, \ldots, r_n) = (h_1, \ldots, h_m)$. The result is a sequence of alignments $(k_1, r_1), \ldots, (k_n, r_n)$. Since manual alignment is labor-intensive, we developed an automatic alignment algorithm.

Our alignment algorithm is based on pattern matching, using a machine-readable kanji dictionary, the KANJIDIC [2], as the reference. The KANJIDIC contains 6,355 kanji characters with their various readings. Starting from the first character in the lexical entry to be aligned, the algorithm looks up in the KANJIDIC for a reading and outputs an alignment such that the concatenation of the chosen readings exactly matches the hiragana reading of the whole entry provided by the lexicon.

Since the KANJIDIC does not include every reading of every kanji character and since some kanji compounds have readings that bear no resemblance to either Chinese or Japanese readings of the composing kanji characters, the simple pattern-matching scheme sometimes fails to yield successful alignments. In fact, in the latter case, even native Japanese speakers would have some difficulties deciding how to align. To make the best use of the training data, if an entry cannot be successfully aligned by our algorithm, we associate a "super" alignment $(\tilde{k}, \tilde{r})$ with this entry, where $\tilde{k} = (k_1, \ldots, k_n)$ and $\tilde{r} = (h_1, \ldots, h_m)$.

3.2. Model Training

In our system, the language model (1) is realized as the linear interpolation of a backing-off word trigram model [4] and a backing-off class trigram model [3], both trained on the training data created according to Section 3.1. Note that a "word" here refers to a proper alignment (k_i, r_i) or a super alignment $(\tilde{k}, \tilde{r})$. In other words,

$$\begin{aligned} & p\big((k_1, r_1), \ldots, (k_n, r_n)\big) \\ & \quad =: \; p(a_1, \ldots, a_n) \\ & \quad \approx \; (1-\alpha) \prod_{i=1}^{n} p(a_i | a_{i-2}, a_{i-1}) + \\ & \qquad \alpha \prod_{i=1}^{n} p\big(g(a_i) | g(a_{i-2}), g(a_{i-1})\big) p\big(a_i | g(a_i)\big), \end{aligned} \quad (3)$$

where $g(a_i)$ is the equivalent class to which the alignment $a_i := (k_i, r_i)$ belongs and $0 < \alpha < 1$.

3.3. Hypothesis Search

The hypothesis search module is the only module in the system that actually handles conversion. Starting with the first kanji character in the input entry, this module looks up all possible kanji-reading pairs (k_i, r_i) for the ith kanji character k_i, and appends them to the current hypotheses to form the new hypotheses. This procedure is repeated for the $(i+1)$th kanji character until the end of the input entry is reached. All hypotheses are then evaluated using the language model (3). Finally, a dynamic programming algorithm selects the optimal sequence of hiragana readings in sense of (2) as the output.

4. EVALUATIONS

4.1. Setup

For our evaluations, we used an extensive lexicon database that we had collected from various sources as our knowledge source. We designated 306,592 entries randomly selected from this lexicon as the training set and the remaining 31,550 entries as the cross-validation set. Both sets were then aligned by our alignment algorithm. The word trigram model and class trigram model were both trained on the training set while the cross-validation set was used to compute the interpolation weight α in (3), which was found to be 0.483.

We evaluated our conversion system on three different test sets, which consisted of lexical entries respectively extracted from the following text databases:

1. JNAS: Selected articles from the Mainichi Newspaper issued between 1991 and 1994 [1].
2. MITI: Whiter papers published by Japan's Ministry of International Trade and Industry from 1993 to 1995 [5].
3. JEIDA: A survey report on the trend of natural language processing from Japan Electronics Industry Development Agency's annual report [5].

Domains of these databases vary. The domain of JNAS is the most general while that of JEIDA is the most specific.

Since we focused on open-set tests, we excluded those entries which were found either in the training set or in the cross-validation set. We also removed entries consisting of only a single character. Some statistics about these test sets are given in Table 1.

4.2. Results

For each test set, we measured the entry error rate (EER), which is the ratio of the number of incorrectly converted entries to the total number of entries in the test set, and the

Test set	Number of entries	Number of kanji characters
JNAS	1779	5450
MITI	762	2416
JEIDA	336	972

Table 1. Statistics of test sets.

Test set	EER	HCER
JNAS	21.64%	10.28%
MITI	20.85%	8.75%
JEIDA	19.35%	10.89%

Table 2. Evaluation results.

hiragana character error rate (HCER), which is the ratio of the sum of deleted, inserted, and substituted hiragana characters to the total number of hiragana characters. The EER is an indication how often the input entry is incorrectly converted while the HCER is a measure of the effort that the user would have to spend on correcting. These error rates are summarized in Table 2.

After examining the errors, we found that most errors are attributed to kanji-reading combinations that are legitimate but unseen in the training data. Among such errors, many occur at inflections of a root form consisting of a single kanji character. In Japanese, inflections of a root usually differ only in the suffixes, which normally consist of one or more hiragana characters. If the input entry is an inflection of a root form with a single kanji character and if this inflection is not seen in the train data, then, at the hypothesis search stage, our language model backs off to the unigram (or zerogram) of the root-reading pair. If the correct reading of this root kanji character is not the most frequent one in the training data, then an error usually results. This is the weakness of our statistical approach.

5. CONCLUSIONS

We have presented a kanji-to-hiragana conversion system based on a statistical approach. For the three test sets in our open-set evaluations, the system yields correct conversions for 78% to 80% of input entries and accuracies of output hiragana characters range from 88% to 91%.

The system we proposed can be easily extended by augmenting the training lexicon and the skill required for the augmentation is quite low. Although our investigation is done on Japanese, the overall approach is also applicable to Chinese. In fact, for Chinese, the automatic alignment module is almost trivial since each Chinese character is monosyllabic.

However, our system suffers from the coverage problem. In particular, improvement on treatment of inflections is most needed. To this end, we will consider incorporating rule-based processing in the system as many inflections can be well dealt with by rules. The research then will be to decide how much rule-based processing should be incorporated since it will cause maintainability problems and increase the system complexity.

6. ACKNOWLEDGEMENTS

The author would like to thank Sachiko Morishita and Yumiko Sasaki for their kind assistance.

7. REFERENCES

[1] *ASJ Continuous Speech Corpus: Japanese Newspaper Article Sentences (JNAS)*. CD-ROM. Acoustical Society of Japan, Tokyo, Japan, 1997.

[2] kanjidic.gz, Feb. 3, 1999, <ftp.cc.monash.edu.au/pub/nihongo/kanjidic.gz> (Jan. 12, 2000).

[3] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modelling," *Proc. EUROSPEECH*, Berlin, Germany, Sept. 1993, pp. 973–976.

[4] R. Kneser and H. Ney, "Improved backing–off for m-gram language modeling," *Proc. ICASSP*, Detroit, MI, May 1995, vol. 1, pp. 181–184.

[5] *RWC Text Database Ver. 2*. CD-ROM. Real World Computing Partnership, Japan, 1998.

[6] M. Shibatani, *The Languages of Japan*. Cambridge University Press, Cambridge, UK, 1990.

[7] J. Picone, T. Staples, K. Kondo, and N. Arai, "Kanji-to-hiragana conversion based on a length-constrained n-gram analysis," *IEEE Trans. Speech and Audio Processing*, vol. 7, no. 6, pp. 685–696, 1999.