# 機械翻訳手法に基づいた日本語の読み推定

**羽鳥 潤**
東京大学大学院 情報理工学系研究科
東京都文京区本郷 7-3-1
hatori{at}is.s.u-tokyo.ac.jp

**鈴木 久美**
東京大学大学院 情報理工学系研究科
東京都文京区本郷 7-3-1
hatori{at}is.s.u-tokyo.ac.jp

## Abstract

本稿では、日本語の漢字仮名交じり文における読み付けの問題に対して、新しい手法を提案する。特に、句に基づいた機械翻訳手法を用いて、単語分割・読み付けの問題を同時に解き、。書記素－音素変換タスクや翻字タスクに提案された部分文字列ベースの手法を用いて、ウェブ文書等から教師無し学習により辞書を構築する方法、そして、それに対して従来から用いられてきた辞書ベースの手法を包括的に扱う事を目標とする。特に、単語・部分文字列から平仮名への変換操作の合成や、結合 $n$-gram 等の手法を取り入れ、全体を生成混成モデルとして解く事により、大規模コーパスからの学習と容易な分野適応を可能にしている。また、最終的なモデルを様々な分野のコーパスで評価し、識別的な文脈素性を用いた既存手法との比較を、実際の出力例を交えながら議論する。

## 1 緒論

本稿では、日本語の漢字仮名交じり文における読み付けの問題を取り上げる。This paper explores the problem of assigning pronunciation to Japanese text, which consists of a mixture of ideographic and phonetic characters. The task is naturally important for the text-to-speech application (Schroeter et al., 2002), and has been researched in that context as letter-to-phoneme conversion, which converts an orthographic character sequence into phonemes. In addition to speech applications, the task is also crucial for those languages that require pronunciation-to-character conversion to input text, such as Chinese and Japanese, where users generally type in the pronunciations of words, which are then converted into the desired character string via the software application called pinyin-to-character or kana-to-kanji conversion (e.g. Gao et al. (2002a); Gao et al. (2002b)).

Predicting the pronunciation of Japanese text is particularly challenging because the pronunciation of Japanese characters and words is highly ambiguous. Japanese orthography employs four sets of characters: *hiragana* and *katakana* (called generally as *kana*), which are syllabary systems and thus phonemic; *kanji*, which is ideographic and consists of several thousand characters; and Roman alphabet. Out of these, kanji characters typically have multiple possible pronunciations[1]; especially those in frequent use tend to have many (5–10, sometimes as many as 20). This yields an exponential number of pronunciation possibilities when multiple kanji characters are combined in a word. The pronunciation of a Japanese word is also often idiosyncratic, just like the pronunciation of English words is: you cannot straightforwardly pronounce a word unless you know the word.

This idiosyncratic property of the word pronunciation naturally motivates us to take a dictionary-based approach. Traditionally, most of the approaches to Japanese pronunciation prediction have regarded the problem as a word pronunciation disambiguation task, in which, followed by a word segmentation step, a word-level pronunciation among those defined in a dictionary is selected (Nagano et al., 2006; Mori and Neubig, 2010a). For example, given a word "人気", a traditional method tries to select the most appropriate pronunciation out of the three dictionary entries: *ninki* (popularity), *hitoke* (sign of life) and *jinki* (people's atmosphere), depending on the context.

However, since the dictionary-based approach is inapplicable to those words that do not exist in the dictionary, there needs to be a mechanism for handling out-of-vocabulary (OOV) words. Nonetheless, the problem with OOV words has received little attention to date. The state-of-the-art system called KyTea (Mori and Neubig, 2010a) uses a simple OOV model based on a noisy channel model. More recently, our recent work (to appear) proposed a substring-based pronunciation prediction model. However, the applicability of this approach is limited to noun phrases because the model learns only from instances extracted from Wikipedia.

In this paper, we extend our previous approach to build a pronunciation prediction model that can deal with the pronunciation of full sentences consisting of both dictionary words and OOV words. By synthesizing substring- and dictionary-based pronunciations, our

[1] In UniDic (Den et al., 2007), the average number of pronunciations of each kanji is 2.3.

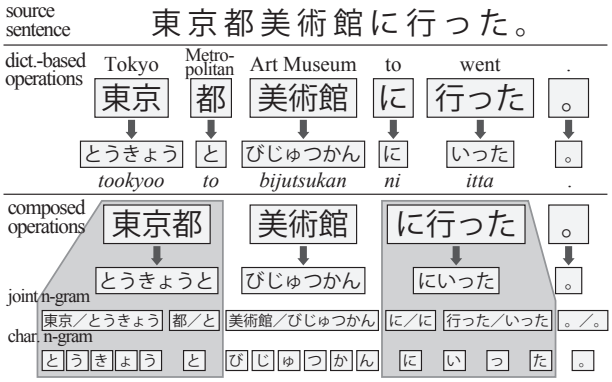


図 1: Overview of the model.

new model seamlessly deals with the tasks of *word pronunciation disambiguation* and *pronunciation prediction* for OOV words within a single model. In order to capture context for the purpose of disambiguation and to incorporate dependency among pronunciations for OOV words, we use composed operations, which were successfully applied in a limited manner in our previous work, and a joint $n$-gram model as a feature function, which incorporates smoothed context for both the source and target sides.

We conduct an extensive evaluation on various datasets, including newswire articles, search query logs, person names, and Wikipedia-derived instances. We train our models with newswire texts, in addition to the word–pronunciation pairs that are automatically extracted from Wikipedia. Our final model outperforms existing systems, achieving around 90% accuracy in most of the domains. The use of composed operations and a joint $n$-gram language model are both shown to significantly improve the accuracy. We also give a detailed analysis of the comparison of the proposed model with other systems, KyTea in particular. In addition to achieving the best known results on the task of Japanese pronunciation prediction, we believe that our work, being based on the SMT framework, has implications to the task of MT generally.

The rest of the paper is organized as follows. In Section 2, we describe the task setting and related work. In Section 3, we describe our model in detail. In Section 4, we present our experimental setting, result, and discussion of the result. Finally, we state our concluding remarks in Section 5.

## 2 Background

### 2.1 Pronunciation Prediction: Task Setting

We define the task of pronunciation prediction as converting a string of orthographic characters representing a sentence (or a word or phrase) into a sequence of hiragana, which corresponds how the string is pronounced. For example, given a Japanese sentence “東京都美術館に行った。” (“I went to the Tokyo Metropolitan Art Museum.”) in 図**??**, the system is expected to output a sequence of phonetic characters, “とうきょうとびじゅつかんにいった。”, pronounced as *tookyoo to bijutsukan ni itta*. The substrings “に” and “った” in the original sentence are hiragana sequences, which respectively correspond to a particle (”to”) and an ending of an inflecting verb. Although hiragana characters do not have ambiguity in pronunciation, this kind of common usages of hiragana provide useful context for disambiguating neighboring kanji pronunciation.

### 2.2 Related Work

The task of pronunciation prediction is inspired by previous research on string transduction. The most directly relevant one is the work on letter-to-phoneme conversion. The methods include joint $n$-gram models (e.g., Bisani and Ney (2002); Chen (2003); Bisani and Ney (2008)), discriminatively trained substring-based models (e.g., Jiampojamarn et al. (2007); Jiampojamarn et al. (2008)) which are influenced by the phrasal SMT models (Koehn et al., 2003), and minimum description length-based methods (Reddy and Goldsmith, 2010).

Similar techniques to the letter-to-phoneme task have also been applied to the transliteration task (e.g. Knight and Graehl (1998), Sherif and Kondrak (2007)). Recently, Cherry and Suzuki (2009) proposed a hybrid model, which incorporates the generative model by Sherif and Kondrak (2007) into an SMT-style discriminative framework. The joint $n$-gram estimation method has also been applied to the task of transliteration (e.g., Li et al. (2004); Jiampojamarn et al. (2010)) and machine translation (e.g., Mari 単 o et al. (2006)). In SMT, some researchers (e.g., Och and Ney (2004)) have also used dictionary-based features to incorporate dictionary knowledge.

In Japanese pronunciation prediction, Sumita and Sugaya (2006) proposed a method to use the web for assigning word pronunciation, but their focus is limited to the pronunciation disambiguation of known proper nouns. Kurata et al. (2007) and Sasada et al. (2009) discussed the methods of disambiguating new word pronunciation candidates using speech data. The joint $n$-gram estimation has also been applied (Nagano et al., 2006; Mori et al., 2010b).

As described in 第**??**節, this work is an extension of our recent work (to appear), which addresses the pronunciation prediction of Japanese words using a semi-supervised approach, with a focus on building a classifier to harvest kanji–pronunciation pairs from Wikipedia. However, the applicability of this approach is limited to nouns phrases, and the evaluation is performed based solely on Wikipedia instances. The system is described

as one of the baseline systems in 第**??**節.

Another closely related work is KyTea (Mori and Neubig, 2010a), which is one of the state-of-the-art systems for the task of Japanese pronunciation prediction. This system exploits an SVM-based two-step approach, which performs a word segmentation step, followed by a pronunciation disambiguation step for each word segment. This approach contrasts with ours in that our approach is a joint model that solves the two tasks simultaneously using joint probability distribution. In the pronunciation prediction step, if the word in question exists in the dictionary, the KyTea uses character and character-type[2] $n$-grams within a window as features; for OOV words, a separate OOV model based on a noisy channel model, which uses target (hiragana) character bigram and target-to-source (hiragana-to-kanji) transduction probability, is used. Whereas the KyTea uses the discriminative features in the pronunciation disambiguation model, our model instead uses composed operations and character/joint and $n$-gram language models. The training of KyTea essentially requires probabilistic optimization over all the appearances of the same word, making the training less scalable than our model, which only requires frequencies of operations and phrases from the training data.

# 3 読み推定モデル

この節では、読み推定に対する提案手法を紹介する。この手法は、統計的機械翻訳手法に基づいているが、アライメントは単調（順序が前後しない）で、挿入・削除は起きないと仮定する。[3]

## 3.1 デコーダ

統計的機械翻訳においてよく使われている (Och, 2003) ように、識別的な句ベースのデコーダを利用する。生成混成モデルでは、複数の生成確率が実数値素性として識別モデルにおいて用いられる[4]。このデコーダは、入力列 $s$ と出力列 $t$ のペアに対して、

Given the source sequence $s$ and the target character sequence $t$, we define real-valued features over $s$ and $t$, $f_i(s,t)$ for $i \in \{1,\ldots,n\}$. The score of a sequence pair $\langle s,t \rangle$ is given by the inner product of the weight vector $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_n)$ and the feature vector $\mathbf{f}(s,t)$.

素性としては、(1) 双方向翻訳確率 $P(t|s), P(s|t)$、(2) 出力文字 $n$ グラム確率 $P(t)$、(3) 結合 $n$ グラム確率 $P(s,t)$、出力文字数、句の数を利用する。ここで、結合 $n$ グラム確率は、入力単語と出力単語のペアの列（「⟨ 床屋, とこや ⟩ ⟨ に, に ⟩ ⟨ 行く, いく ⟩」）に対する $n$ グラム言語モデル確立である。また、部分文字列モデルでは、これに加えて辞書語素性（辞書語にマッチした句の長さの合計）を用いる。

[2]The type is either kanji, hiragana, katakana, Roman, numeral, or none of these.

[3]実際には、不忍池（しのばずいけ）のように厳密には単調でない例、一関（きちのせき）のような挿入が起こる事もある。

[4]

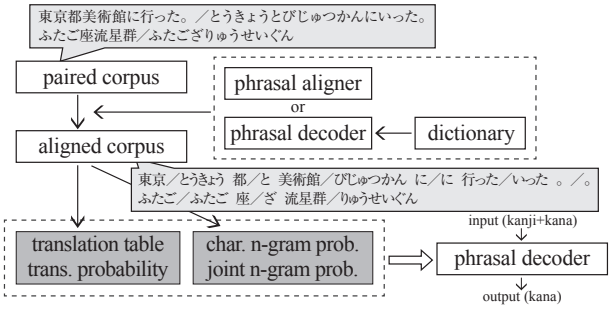


図 2: Overview of the training.

訓練には、平均化パーセプトロンを用いた。デコーディングには、Zens and Ney (2004) で用いられているようなスタック・デコーダを用いた。

## 3.2 翻訳表

The corpora we use are a collection of pairs of a Japanese sentence and its kana sequence. For obtaining these alignments, we use either a phrasal aligner, as in our previous work, or a dictionary-based phrasal decoder, which is novel in this work, as shown in 図**??**. Accordingly, we investigate two frameworks, namely the substring- and dictionary-based methods.

### 3.2.1 部分文字列モデル

部分文字列モデルでは、Zhang et al. (2008) の用いたような句ベースのアライナを用いて、教師データを全く用いずにアライメントを取る。まず、1 つの漢字が長さ 1 以上の読み列に対応しているとしてアライメントをとり、後に翻訳操作の合成（第**??**節参照）を行う事でより大きな単位のアライメントを取れるようにしている。

### 3.2.2 辞書語モデル

辞書語モデルでは、辞書語に基づいた句デコーダを用いてアライメントを取る。このデコーダでは、翻訳操作は辞書語単位のものとなる。

In the *dictionary-based model*, we obtain the alignments using a phrasal decoder which is trained on a dictionary. This essentially treats the dictionary words as the minimal unit of substring operations. Once a dictionary-based model is built, we use it to decode a paired corpus to obtain the alignments between the source and target strings. This dictionary-based decoder is basically the same as the one described in 第**??**節, with the exception that we only use two simple features: the forward translation probability and the phrase count. In this process, instances including any operation that is not defined in the dictionary are discarded; this is the major difference with the substring-based model described above, which uses all instances of training data. Since Japanese pronunciation dictionaries typically include single-kanji pronunciations as well as word-level pronunciations, the

dictionary-based model is still able to handle OOV words that are compositionally comprised of single-kanji pronunciations. This can be considered as a back-off process, implicitly incorporating word–character hybrid aspect into the model, with the preference of longer phrases captured by the phrase count feature.

It is also conceivable to mix the substring- and dictionary-based operations obtained from the same corpus. However, in our preliminary experiments, we have found that this significantly deteriorates the performance of the model, probably because the mixture of inconsistent alignments breaks the independent nature of a word's occurrence in the joint $n$-gram language model.[5]

#### 3.2.3 翻訳操作の合成

We extend the composed operations, which were also used in our previous work (to appear), so that they can handle dictionary-based operations with joint $n$-gram estimation. The composed operations are quite effective for capturing the local context: for example, a phrase "行った" can be pronounced in two ways: *itta* "went" and *okonatta* "did", which cannot be distinguished without any context. However, if this phrase is preceded by a hiragana particle に *ni* "to", we can easily tell that the correct pronunciation is quite likely to be *itta*, because the pronunciation *ni okonatta* is unusual. The composed operations are also useful in capturing the pronunciation of compound nouns: for example, due to the phonological process called *rendaku* (sequential voicing; Vance (1987)), 神棚 "altar" is pronounced as *kami-dana*, while the components of this word are individually pronounced as *kami* ("god") and *tana* ("shelf"). In the substring-based model, the composed operations are also beneficial in recovering noncompositional pronunciation of words, as the initial one-kanji-to-many-kana alignment stage arbitrarily split the kana sequence to align to kanji characters.

If we simply adopt the composed operations with the joint $n$-gram estimation, it may cause the same problem of inconsistent alignments as described in 第**??**節. To avoid this problem, we let the model retain the original operations even after they are composed. As shown in 図 **??**, even after the two operations "東京→とうきょう" and "都→と" are composed into "東京都→とうきょうと", the joint $n$-gram probability is estimated based on the original (non-composed) operations. For efficiency purposes, we only retain the composition of the first appearance of each composed operation even if multiple compositions are possible.

[5]For example, given a phrase "夏の星座" (summer constellation), the substring- and dictionary-based models decompose it into "夏/の/星/座" and "夏/の/星座", respectively. If these two inconsistent alignments both exist in the training data, the $n$-gram language model expect that the occurrence of "星座" is independent of that of "星" given the $n$-gram context "夏/の", but this is not the case.

## 4 実験

### 4.1 辞書

In the dictionary-based framework, we need a dictionary based on which we obtain the alignments. We use a combination of three dictionaries: UniDic (Den et al., 2007), Iwanami Dictionary, and an in-house dictionary that was available to us of unknown origin. UniDic is a dictionary resource available for research purposes, which is updated on the regular basis and includes 625k word forms as of the version 1.3.12 release (July 2009). Iwanami Dictionary consists of 107k words, which expands to 325k surface forms after considering *okurigana* (verb inflectional ending) variations (as in "[掛:か, か] る" → {"掛る", "掛かる", "かかる"}). The in-house dictionary consists of a total of 226k words and single-kanji pronunciations. After removing duplicates, the combined dictionary consists of 770k word/kanji–pronunciation pairs. Note that the dictionary is also used for the dictionary feature in the substring-based model.

### 4.2 データセット

As described in Section 3, we need substring-aligned parallel data of Japanese phrases and their corresponding pronunciation to train the models. Following our previous work (to appear), we first investigate the automatic acquisition of pronunciation pairs from unannotated corpus, specifically taking advantage of the convention of Japanese text that the pronunciation of those words that are difficult or unusual to pronounce are often indicated in parentheses immediately following the word in question.[6] By using simple regular-expression-based heuristics, we extracted 460k word–pronunciation pairs (referred to as "Wiki-Train") from Japanese Wikipedia articles as of January 24, 2010. Since the extracted word–pronunciation pairs are noisy[7] and mostly consist of noun phrases, we also investigate the use of an annotated newspaper corpus, which is comprised of 1.4M sentence pairs (referred to as "News-Train"). Also, in the substring-based model, the above-mentioned dictionaries are additionally used as training instances. For the comparison experiment with KyTea, we use an annotated portion of the Balanced Corpus of Contemporary Written Japanese (BCCWJ; Maekawa (2008)), which consists of texts from books, whitepapers, newspapers, and Yahoo! JAPAN Q&A. We use the 2009 Core Data, which consists of 37K sentences annotated with word pronunciations (referred to as "BCCWJ").

Our test sets consist of six datasets from various domains.

[6]This contrasts with the dictionary, where the pronunciations of all words are found.

[7]Our previous work found that roughly 10% of these instances are invalid word–pronunciation pairs.

- **News-1**(N1) and **News-2**(N2): collections of newswire articles from the Microsoft Research IME Corpus (Suzuki and Gao, 2005). These articles are from different newspapers from the news corpus we used in training. After noisy instances are filtered out[8], each set consists of 867 and 739 sentences with average source length of 51.8 and 44.9 characters, respectively.
- **Query-1**(Q1) and **Query-2**(Q2): consist of 1,049 and 3,078 instances with 3.8 and 5.7 source characters on average, which are from query logs from a search engine (source undisclosed for blind reviewing). These sets consist of various instances ranging from general noun phrases to relatively new proper nouns.
- **Name**(PN): a collection of 9,170 difficult-to-pronounce words, mostly consisting of person names, with an average source length of 3.0.
- **Wiki**(WP): 2,000 manually cleaned word–pronunciation pairs from Wikipedia, which contain no overlap with the training instances. This test set consists mostly of proper nouns including names of people and locations as well as the terms that are difficult to pronounce. The average source length is 4.1 characters.

For the tuning of the weights of the linear model, we used 200 held-out instances for each test domain, except that the development set of Query-1 is also used for the tuning for Query-2, and the set of Wiki is used for the tuning for Name.

### 4.3 実験設定

We use our original implementation of the phrasal aligner and decoder. The aligner runs with the EM algorithm, with the source (kanji) and target (kana) phrase length limits set to 1 and 4; Alignments to a null symbol in either source or target side prohibited. The decoder runs with the beam size of 20. We did not use the frequency threshold for the operation cutoff. The maximum number of composed operations is 4 for the substring-based model, and 3 for the dictionary-based model. In the substring-based model, character 5-gram and joint 4-gram language models with Kneser-Ney smoothing and the BoS (beginning-of-string) and EoS (end-of-string) symbols are used; in the dictionary-based model, character 5-gram and joint 3-gram models with the same settings are used. All of these parameters and settings are set based on the preliminary experiment with performance and memory efficiency considered. As the evaluation measure, we use instance-level accuracy, which is calculated based on the percentage of the outputs that exactly match the gold standard. The statistical significance of results is evaluated using McNemar's test.

[8]In preparing these test sets, instances including Arabic numerals (i.e. 0,1,⋯,9) or kanji numerals (i.e. 〇, 一,⋯, 九) are excluded because there exist various different standards in how to pronounce them. Some systems output the literal pronunciation of numbers (e.g., "*kokono-ka*" for "9 日" (ninth day)), while other systems leave the numbers untouched (e.g., "*9 nichi*" for "9 日"). The best output is application-dependent; the literal pronunciation is preferred for text-to-speech applications, whereas just outputting numerals as such suits better for the training of Japanese input methods. Therefore, the existence of these instances is problematic especially when we need to perform the comparison among different systems.

| Model | N1 | N2 | Q1 | Q2 | PN | WP |
|---|---|---|---|---|---|---|
| KyTea | 83.6 | 85.9 | 92.9 | 85.6 | 52.9 | 62.9 |
| Proposed (U) | 18.1 | 11.5 | 87.9 | 77.6 | 71.1 | 70.9 |
| Baseline (S) | 23.3 | 31.8 | 87.7 | 73.3 | 83.9 | 64.5 |
| Proposed (S) | 37.6 | 31.8 | 93.3 | 82.7 | 90.5 | **72.9** |
| Trigram (D) | 86.4 | 84.7 | 93.0 | 85.7 | 91.1 | 67.2 |
| Proposed (D) | **89.7** | **88.6** | **95.5** | **87.8** | **92.9** | 70.2 |

表 1: Instance-level accuracy (in %) of pronunciation prediction models. All models except KyTea and Proposed (U) are trained using Wiki-Train and News-Train with the combined dictionary.

### 4.4 ベースライン手法

We describe three baseline models that we use as reference in the experiment. First, we use *KyTea* version 0.13, which is described in 第**??**節. Second, we use the *substring-based baseline model* (referred to as "Baseline (S)"), which is a substring-based model that we proposed in our previous work. This model is similar to the substring-based model that we propose in 第**??**節, but differs in that this baseline model does not use the joint $n$-gram model and dictionary features. Finally, as the third baseline model, we use the *joint trigram model* that is augmented with the dictionary-based operations (referred to as "Trigram (D)"). The same decoder as described in 第**??**節 is used.

## 5 結果と議論

表**??** shows the overall performance of the proposed models, with comparison to other state-of-the-art systems. The first row shows the result of KyTea using the off-the-shelf "full SVM model" provided at the author's page[9]. "Proposed (U)" is an unsupervised, substring-based model, which is trained and tuned exclusively with noisy Wikipedia-derived instances, without using a dictionary and development set. Since the Wiki-Train instances include almost no verb instances, the unsupervised model performs poorly for full sentences in News-1 and News-2 with 18.1% and 11.5% sentence-level accuracy, respectively. The bottom four models are all trained with Wiki-Train and News-Train using all the three dictionaries. "(S)" denotes substring-based models, and

[9]http://www.phontron.com/kytea/model.html. This model is trained on several resources including BCCWJ and UniDic. We could not train KyTea with the same dataset as the proposed model uses because of limited memory. See 表**??** for the comparison based on the same training set.

| Model | N1 | N2 | Q1 | Q2 | PN | WP |
|---|---|---|---|---|---|---|
| KyTea (w/noise) | 68.5 | 65.3 | 88.0 | 79.5 | **67.9** | **65.8** |
| KyTea (wo/noise) | **75.3** | **75.5** | 91.5 | 83.4 | 61.7 | 64.1 |
| Proposed (D) | 73.8 | 75.4 | **92.8**$^{\dagger}$ | **84.9**$^{\dagger}$ | 62.8 | 64.3 |

表 2: Instance-level accuracy (in %) of the models trained on BCCWJ with UniDic. "†" denotes a statistically-significant ($p < 0.01$) difference between KyTea (wo/noise) and Proposed (D).

| Model | N1 | N2 | Q1 | Q2 | PN | WP |
|---|---|---|---|---|---|---|
| Proposed (D) | 89.7 | 88.6 | 95.5 | 87.8 | 92.9 | 70.2 |
| - wo/joint $n$-gram | -5.5 | -3.3 | -1.5 | -3.8 | -4.4 | -4.2 |
| - wo/composed op. | -3.9 | -4.0 | -2.6 | -1.2 | -1.8 | -2.9 |

表 3: Feature ablation results for the dictionary-based model trained on Wiki-Train and News-Train with the combined dictionary. All the losses in accuracy were statistically significant ($p < 0.01$).

"(D)" denotes dictionary-based models. The proposed dictionary-based model reported the best results in five out of six test sets, showing the effectiveness and robustness of the model. On Wiki, the dictionary-based model "Proposed (D)" falls behind the substring-based model "Proposed (S)", probably because the dictionary-based model discards many operations that are not common, but useful for the pronunciation of proper nouns and OOV words in Wikipedia.

表**??** shows the direct comparison between KyTea and the proposed dictionary-based model trained[10] with exactly the same datasets: BCCWJ, Wiki-Train, and UniDic, all of which are from publicly available resources. Whereas "KyTea (w/noise)" uses all instances for training, "KyTea (wo/noise)" uses instances that are filtered using dictionary-based operations, which are a novel contribution of our work. As you can see from 表**??**, this filtering process resulted in a large improvement in accuracy, with the exception that the accuracy decreased a lot on Name (and a bit on Wiki). By manual error analysis, we have found that this is because the UniDic-based operations do not include many single-kanji pronunciations that are commonly used in person's names, such as "美 *mi*" and "人 *to*"[11]. However, as seen from the difference on Name between "Proposed (S)" and "Proposed (D)" in 表**??** (these models use a combination of three dictionaries), this problem seems negligible when a dictionary including common pronunciations for person's names is available. Overall, the proposed model outperforms "KyTea (wo/noise)" in four out of six test sets, showing the effectiveness of the proposed model. Our model lagged a bit behind "KyTea (wo/noise)" on News-1 and News-2, because the problem of word pronunciation disambiguation, for which the discriminative framework is expected to work best, is dominant on the newswire domain. However, considering that the training data is relatively small[12] and the differences were not statistically significant, we can conclude that our model has a comparable performance to KyTea for the task of pronunciation disambiguation, while it has better performance for the task of pronunciation prediction for OOV words. Manual analysis showed that our model indeed has an advantage in outputting phonetically natural pronunciation sequences, partially resolving problems related to *rendaku* and *on–kun* dependencies[13], as in 契約切れ *keiyakugire* (individually pronounced as *keiyaku* and *kire*; "contract expiration"). On the other hand, KyTea is clearly better at capturing generalized context by using the character-type feature, as seen in "ブランド米" (*burando* (in katakana) + *mai*; "brand rice").

表**??** shows the results of the feature ablation experiment for the dictionary-based model. As we mentioned before, the advantage of the joint $n$-gram feature is twofold: incorporating smoothed context into word pronunciation disambiguation (considered to be dominant on News-1/2), as well as incorporating single-kanji pronunciation dependencies into pronunciation prediction for OOV words (considered to be common on Name and Wiki). The improvement observed in all of these domains suggests that the joint $n$-gram probability successfully captured these two aspects. On the other hand, the use of composed operations showed large improvement particularly on News-1/2, suggesting that it is more useful for capturing local context for word pronunciation disambiguation, rather than for pronunciation prediction for OOV words.

図**??** shows the performance of the proposed dictionary-based model with respect to the number of News-Train sentences used for training. At first, the model is trained only with Wiki-Train, and then sentences from News-Train are incrementally added. This can be seen as a process for adapting the OOV model to a fully sentential, disambiguation-capable model. On the test sets other than the newswire domain, the accuracy remains almost unchanged by the addition of newswire sentences, suggesting that the training on the automatically extracted Wikipedia instances is effective enough for the pronunciation prediction for noun phrases in these domains. On the newswire domain, on the contrary, the

[10] Our training of KyTea is performed as follows: We first train a KyTea's segmentation model using BCCWJ and UniDic, and use this model to segment the substring-aligned Wiki-Train instances to obtain a corpus with consistent segmentation, which is then used to train the final model.

[11] Since most of these pronunciations are used only for person's names, some dictionaries (including UniDic) do not cover them. However, for example, the pronunciation "美 *mi*" (meaning "beauty") is quite common in women's names; roughly 3.2% of instances in Name includes this pronunciation.

[12] Since the translation probabilities in our model are based simply on corpus frequency, the model is less powerful with small training data, while it is more scalable.

[13] Pronunciations of kanji are classified into *on* and *kun* pronunciations (corresponding to their origin, Chinese and Japanese), each of which tends to be used consecutively.

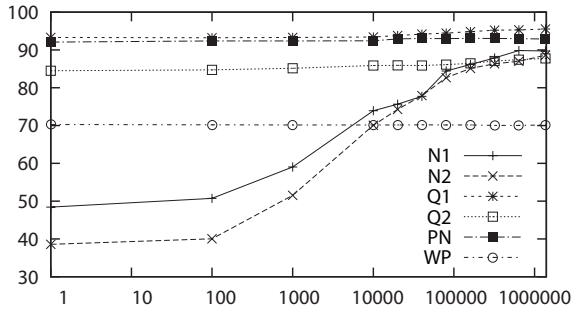


図 3: Performance of the proposed dictionary-based model trained on Wiki-Train, News-Train with respect to the log of the number of the training sentences from News-Train.

accuracy has been largely improved. These results show that our model could successfully combine the tasks of pronunciation disambiguation of dictionary words and pronunciation prediction for OOV words, without deteriorating the performance of the model in any of these domains.

## 6 結論

本稿では、日本語の読み付け問題に対する統合的な手法を提案した。句ベースの統計的機械翻訳手法に基づき、提案手法は既知語に対する読み曖昧性解消問題と未知語の読み推定問題を統一的に扱う事が可能になった。提案手法の基本要素は教師無し学習によって学習することができ、ノイズに対しても頑健である事から、新たな分野に簡単に適応する事が出来る。また、様々な分野の評価セットにおいて実験を行った結果、提案手法は既存手法よりも優れており、ほぼ全分野において 90%近い精度を持つ事が分かった。提案した翻訳操作の合成と結合 $n$ グラムの利用は、優位にモデルの精度を向上させる事が分かった。

誤り分析の結果、SVM によるモデルは提案手法とは異なった性質を示す事が分かった事から、今後、他の手法が利用している文字種素性や、音訓読みの依存関係などを利用する事で、更なる精度向上が期待される。

## References


Maximilian Bisani and Hermann Ney. 2002. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proceedings of the International Conference on Spoken Language Processing (Interspeech 2002)*.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50:434–451.

Stanley F. Chen. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech 2003)*.

Colin Cherry and Hisami Suzuki. 2009. Discriminative substring decoding for transliteration. In *Proceedings of the 2009 Conference on Empirical Methods on Natural Language Processing (EMNLP-2009)*.

Yasuharu Den, Toshinobu Ogiso, Hideki Ogura, Atsushi Yamada, Nobuaki Minematsu, Kiyotaka Uchimoto, and Hanae Koiso. 2007. The development of an electronic dictionary for morphological analysis and its application to Japanese corpus linguistics (in Japanese). *Japanese linguistics*, 22:101–122.

Jianfeng Gao, Mingjing Li, Joshua T. Goodman, and Kai-Fu Lee. 2002a. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing*, 1:3–33.

Jianfeng Gao, Hisami Suzuki, and Yang Wen. 2002b. Exploiting headword dependency and predictive clustering for language modeling. In *Proceedings of the 2002 Conference on Empirical Methods on Natural Language Processing (EMNLP-2002)*.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of HLT-NAACL 2007*.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-2008*.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training framework. In *Proceedings of NAACL-2010*.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL-2003*.

Gakuto Kurata, Shinsuke Mori, Nobuyasu Itoh, and Masafumi Nishimura. 2007. Unsupervised lexicon acquisition from speech and text. In *Proceedings of ICASSP-2007*.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of ACL-2004*.

Kikuo Maekawa. 2008. Compilation of the KOTONOHA-BCCWJ corpus (in Japanese). *Nihongo no kenkyu (Studies in Japanese)*, 4:82–95.

Jos 辿 B. Mari 単 o, Rafael E. Banchs, Josep M. Crego, Adri? de Gispert, Patrik Lambert, Jos 辿 A. R. Fonollosa, and Marta R. Costa-juss?. 2006. N-gram-based machine translation. *Computational Linguistics*, 32.

Shinsuke Mori and Graham Neubig. 2010a. Automatically improving language processing accuracy by using kana–kanji conversion logs (in Japanese). In *Proceedings of the 16th Annual Meeting of the Association for Natural Language Processing (NLP-2010)*.

Shinsuke Mori, Tetsuro Sasada, and Graham Neubig. 2010b. Language model estimation from a stochastically tagged corpus (in Japanese).

Tohru Nagano, Shinsuke Mori, and Masafumi Nishimura. 2006. An n-gram-based approach to phoneme and accent estimation for tts (in Japanese). *Transactions of Information Processing Society of Japan*, 47:1793–1801.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL-2003*.

Sravana Reddy and John Goldsmith. 2010. An MDL-based approach to extracting subword units for grapheme-to-phoneme conversion. In *NAACL*.

Tetsuro Sasada, Shinsuke Mori, , and Tatsuya Kawahara. 2009. Domain adaptation of statistical kana–kanji conversion system by automatic acquisition of contextual information with unknown words (in Japanese). In *Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing (NLP-2009)*.

Juergen Schroeter, Alistair Conkie, Ann Syrdal, Mark Beutnagel, Matthias Jilka, Volker Strom, Yeon-Jun Kim, Hong-Goo Kang, , and David Kapilow. 2002. A perspective on the next challenges for TTS research. In *Proceedings of the IEEE 2002 Workshop on Speech Synthesis*.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *Proceedings of ACL-2007*.

Eiichiro Sumita and Fumiaki Sugaya. 2006. Word pronunciation disambiguation using the web. In *Proceedings of NAACL-2006*.

Hisami Suzuki and Jianfeng Gao. 2005. Microsoft Research IME Corpus. Technical Report MSR-TR-2005-168, Microsoft Research.

Timothy J. Vance. 1987. *An Introduction to Japanese Phonology*. State University of New York Press.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT-NAACL 2004*.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. In *Proceedings of ACL-2008*.