Japanese Pronunciation Prediction as Phrasal Statistical Machine Translation

Jun Hatori[1] **Hisami Suzuki**[2]
[1]Department of Computer Science, University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo 113-0033, Japan
[2]Microsoft Research / One Microsoft Way, Redmond, WA 98052, USA
hatori@is.s.u-tokyo.ac.jp hisamis@microsoft.com

Abstract

This paper addresses the problem of predicting the pronunciation of Japanese text. The difficulty of this task lies in the high degree of ambiguity in the pronunciation of Japanese characters and words. Previous approaches have either considered the task as a word-level classification problem based on a dictionary, which does not fare well in handling out-of-vocabulary (OOV) words; or solely focused on the pronunciation prediction of OOV words without considering the contextual disambiguation of word pronunciations in text. In this paper, we propose a unified approach within the framework of phrasal statistical machine translation (SMT) that combines the strengths of the dictionary-based and substring-based approaches. Our approach is novel in that we combine word- and character-based pronunciations from a dictionary within an SMT framework: the former captures the idiosyncratic properties of word pronunciation, while the latter provides the flexibility to predict the pronunciation of OOV words. We show that based on an extensive evaluation on various test sets, our model significantly outperforms the previous state-of-the-art systems, achieving around 90% accuracy in most domains.

1 Introduction

This paper[1] 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 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 input methods (e.g. Gao et al. (2002a); Gao et al. (2002b)).

Predicting the pronunciation of Japanese text is particularly challenging because the word and character pronunciations are 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[2]; especially those in frequent use tend to have many — between 5 and 10, sometimes as many as 20. This yields an exponential number of pronunciation possibilities when multiple kanji characters are combined in a word. Also, the pronunciation of a word is frequently idiosyncratic.

This idiosyncratic property of the word pronunciation naturally motivates us to take a dictionary-based approach. Traditionally, most approaches to Japanese pronunciation prediction have regarded the problem as a word pronunciation disambiguation task. Since there are no white spaces between words in Japanese text, these approaches first segment an input sentence/phrase into words, and then select a word-level pronunciation among those defined in a dictionary (Nagano et al., 2006; Neubig and Mori, 2010). For example, given a word “人気”, these methods try 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, in these approaches, seg-

[1] This work was conducted during the first author’s internship at Microsoft Research.

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

mentation errors tend to result in the failure of the following step of pronunciation prediction. Moreover, since the dictionary-based approach is inapplicable to those words that are not in the dictionary, there needs to be a separate mechanism for handling out-of-vocabulary (OOV) words.

Nonetheless, the problem of OOV words has received little attention to date. Traditional systems either bypass this problem completely and assign no pronunciation to OOV words, as Mecab (Kudo et al., 2004), a Japanese morphological analyzer, does; or use a simple model to cover them (e.g. Neubig and Mori (2010) uses a noisy-channel model with a character bigram language model). Our previous work (Hatori and Suzuki, 2011) explicitly addresses the problem of predicting the pronunciation of OOV words, but focuses solely on predicting the pronunciation of nouns that are found in Wikipedia in isolation, and does not address the contextual disambiguation of pronunciation at the sentence level.

In this paper, we propose a unified approach based on the framework of phrasal statistical machine translation (SMT), addressing the whole sentence pronunciation assignment while integrating the OOV pronunciation prediction as part of the whole task. The novelty of our approach lies in using word and single-character pronunciations from a dictionary within the SMT framework: the former captures the idiosyncratic properties of word pronunciation, while the latter provides the flexibility to predict the pronunciation of OOV words based on the sequence of pronunciations at the substring level.

In addressing the pronunciation disambiguation problem within the framework of phrasal SMT, we extend the use of composed operations, which were applied in a limited manner in Hatori and Suzuki (2011). Within our dictionary-based model, the composed operations are able to incorporate the composition of dictionary words (i.e. phrases) as well as substrings of the character sequence (i.e. (partial) words). In this sense, our approach is more like a standard monotone phrasal SMT, rather than the substring-based string transduction. We also propose to use the joint n-gram model as a feature function, which has been proven to be effective in the letter-to-phoneme conversion task (Bisani and Ney, 2008; Jiampojamarn et al., 2010). In the context of our current task, this feature not only incorporates smoothed contextual information for the purpose of pronunciation disambiguation, but also captures the dependency between single-kanji pronunciations, which is effective for predicting the pronunciation of OOV words.

We collected an extensive evaluation set for the task, including newswire articles, search query logs, person names, and Wikipedia-derived instances. Using these test sets, we show that our model significantly outperforms the previous state-of-the-art systems, achieving around 90% accuracy in most test domains, which is the best known result on the task of Japanese pronunciation prediction to date. We also give a detailed analysis of the comparison of the proposed model with an SVM-based model, KyTea (Neubig and Mori, 2010), through which we hope to shed light on the remaining issues in solving this task.

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 to how the string is pronounced. For example, given a Japanese sentence “東京都美術館の狩野探幽展に行った。” (“I went to the Exhibition of Tanyu Kano at the Tokyo Metropolitan Art Museum.”), the system is expected to output a sequence of hiragana, “とうきょうとびじゅつかんのかのうたんゆうてんにいった。”, pronounced as *tookyoo to bijutsukan no kanoo tanyuu ten ni itta*. The task involves two sub-problems: (a) contextual disambiguation of a word pronunciation, e.g., 行った can be pronounced either as いった *itta* “went” or おこなった *okonatta* “did” depending on the context; (b) pronunciation prediction of OOV words, e.g., in the above example, 狩野探幽展 (“the Exhibition of Tanyu Kano”) is not likely to be in the dictionary, so the pronunciation must be reasonably guessed based on the possible pronunciations of individual characters.

2.2 Related Work

Our research on pronunciation prediction is inspired by previous research on string transduction. The most directly relevant is the work on letter-to-phoneme conversion. Previous approaches to this task include joint n-gram models (e.g., Bisani and Ney (2002); Chen (2003); Bisani and Ney (2008)) and discriminatively trained substring-based models (e.g., Jiampojamarn et al. (2007); Jiampojamarn et al. (2008)). This task is typically evaluated at the word level, and therefore does not include contextual disambiguation.

Similar techniques to the letter-to-phoneme task have also been widely applied to the transliteration task (Knight and Graehl (1998)). The most relevant to the current task include an approach based on substring operations in the SMT framework (e.g., Sherif and Kondrak (2007), Cherry and Suzuki (2009)), and those that use joint n-gram estimation method for the task of transliteration (e.g., Li et al. (2004); Jiampojamarn et al. (2010)). However, similarly to the letter-to-phoneme task, the contextual disambiguation of the words has not received much attention.

The task of Japanese pronunciation prediction itself has been a topic of investigation. 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) discuss the methods of disambiguating new word pronunciation candidates using speech data. Nagano et al. (2006) and Mori et al. (2010b) investigated the use of the joint n-gram estimation to this task.

More recently, Neubig and Mori (2010) proposed a classifier-based system called KyTea, which is one of the current state-of-the-art systems for the task of Japanese pronunciation prediction. As we use this system as one of our baseline systems, we describe this work in some detail here. KyTea exploits an SVM-based two-step approach, which performs a word segmentation step, followed by a pronunciation disambiguation step for each word segment. In the pronunciation prediction step, if the word in question exists in the dictionary, KyTea uses character and character-type n-grams within a window as features for the SVM classifier. For OOV words, a simple OOV model based on a noisy channel model with a character bigram language model is used. While KyTea uses the discriminative indicator features, our model instead uses character/joint n-gram language models and composed operations (to be explained in Section 3.3.2) to capture the context for the purpose of pronunciation disambiguation. The use of the indicator features essentially requires probabilistic optimization of a large number of weights, making the training less scalable than our model, which only requires frequencies of operations and phrases in the training data.

In our previous work (Hatori and Suzuki, 2011), we addressed the pronunciation prediction of Japanese words in a semi-supervised, substring-based framework, using word-pronunciation pairs automatically extracted from Wikipedia. Though we obtained more than 70% accuracy on Wikipedia data, the model is quite specific to handling the noun phrases in Wikipedia, and it is not clear if the approach can handle the pronunciation assignment of a general text, which includes the pronunciation prediction and disambiguation of the words of all types at the sentence level. Since our current work is an extension of this approach, we also adopt our previous work as one of our baseline models in Section 4.4.

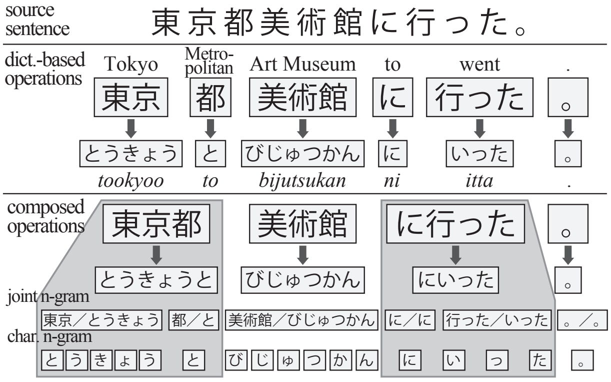

Figure 1: Overview of the model.

3 Pronunciation Prediction Model

This section describes our phrasal SMT-based approach to pronunciation prediction, which is an extension of our previous work (Hatori and Suzuki, 2011). We assume that the task of translating a Japanese orthography string to a hiragana string is basically monotone and without insertion or deletion. The overview of our model is given in Figure 1. The components of the model will be explained below.

3.1 Training and Decoding

As is widely used in SMT research (Och, 2003), we adopt a discriminative learning framework that uses component generative models as real-valued features (Cherry and Suzuki, 2009). 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 $\mathbf{f}(s, t)$.

For the training of model parameters, we use the averaged perceptron (Collins and Roark, 2004): given a training corpus of transduction derivations, each of which describes a word/substring operation sequence converting s into t, the perceptron iteratively updates the weight vector every time it encounters an instance for which the model outputs a wrong sequence. For decoding, we use a

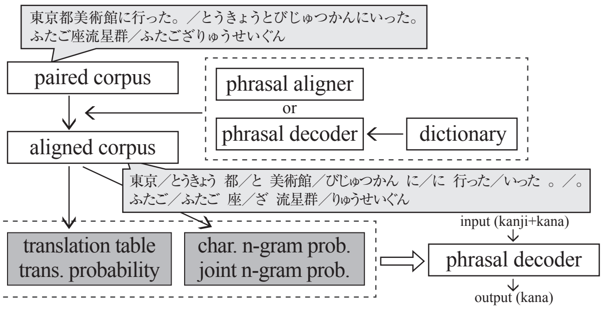
東京都美術館に行った。／とうきょうとびじゅつかんにいった。
ふたご座流星群／ふたござりゅうせいぐん
paired corpus
phrasal aligner
or
phrasal decoder
dictionary
aligned corpus
東京／とうきょう 都／と 美術館／びじゅつかん に／に 行った／いった 。／。
ふたご／ふたご 座／ざ 流星群／りゅうせいぐん
translation table
trans. probability
char. n-gram prob.
joint n-gram prob.
input (kanji+kana)
phrasal decoder
output (kana)

Figure 2: Overview of the training.

stack decoder (Zens and Ney, 2004).

3.2 Features

For our baseline model features, we first use those from Hatori and Suzuki (2011): the bidirectional translation probabilities, $P(t|s)$ and $P(s|t)$, the target character n-gram probability, $P(t)$, the target character count, and the phrase count. In addition, we incorporate the joint n-gram probability, $P(s,t)$, as a feature (described in Section 3.2.1). The estimation of the translation and joint/character n-gram probabilities requires a set of training corpus with source and target alignment at the word/substring level. Once these probabilities have been estimated by using the frequency of (the sequences of) operations in the training set, we only need a small tuning set to adjust the feature weights of the model. This makes online training and domain adaptation easy, and makes our model more scalable compared to fully discriminative systems with indicator features, such as KyTea.

3.2.1 Joint n-gram Language Model Feature

Motivated by the success in the transliteration task (Jiampojamarn et al., 2010), we incorporate the joint n-gram language model into our SMT-based framework. The joint n-gram sequence is the sequence of operations used in the transduction: for example, when a paired sentence "床屋に行く／とこやにいく" is decomposed into three operations "床屋→とこや, に→に, 行く→いく", the corresponding joint n-gram sequence is "⟨ 床屋, とこや ⟩ ⟨ に, に ⟩ ⟨ 行く, いく ⟩". The effectiveness of this feature is confirmed in our experiments in Section 5.2.

3.3 Translation Table

The corpora we use are a collection of pairs of a Japanese sentence and its hiragana sequence, as described as "paired corpus" in Figure 2. These are just like bilingual corpora if we regard the hiragana sequence as monotonically translated from Japanese text. Since the original corpora do not have any word segmentation or word/substring alignments, we first need to obtain them to construct the translation table for the decoder. In previous work, KyTea used a corpus that is manually aligned using words as a unit of alignment, while Hatori and Suzuki (2011) used an unsupervised substring-based alignment. The former is not scalable easily, while the latter cannot take advantage of existing dictionaries. In this work, we use a novel application of dictionary-based phrasal decoder in order to create an aligned corpus, which allows us to use dictionary information while learning substring-based alignments for handling OOV pronunciation prediction.

3.3.1 Dictionary-based model

In the dictionary-based model we propose, alignments are obtained using a phrasal decoder which is based on a dictionary. This essentially treats the dictionary entries as the minimal unit of substring operations, instead of using single-kanji pronunciations estimated from training corpora as in the case of the substring-based model (Hatori and Suzuki, 2011). We first build a simple dictionary-based decoder with only two features: the forward translation probability and the phrase count; and then use it to decode a paired corpus to obtain the alignments between the source and target strings. In this process, instances including any operation that is not defined in the dictionary are discarded; this is a major difference with the substring-based model of Hatori and Suzuki (2011), which uses all instances of training data.

Since Japanese dictionaries typically include single-kanji entries as well as word entries[3], dictionary-based substring operations actually consist of both single-kanji (that is not a word per se) and word pronunciations. This is why our dictionary-based model is still able to handle OOV words. We show in Section 5 that the benefit of removing noisy training samples by this process outweighs the risk of discarding infrequent or non-standard pronunciations that do not exist in the dictionary.

3.3.2 Composed operations

Our previous work (Hatori and Suzuki, 2011) exploits composed operations in order to include local contextual information in the substring-based model. Given a paired corpus, they use an aligner to obtain single-character alignments, which maps

[3] This is because each kanji character is a morpheme representing a meaning, and is worth an entry in dictionaries.

one kanji to one or more kana characters, which are then composed into larger operations. This procedure makes it possible to obtain longer alignments with limited memory, rather than using the source phrase length larger than one. In the current work, we extend the use of composed operations so that they work properly with the joint n-gram estimation.

The composed operations are beneficial for capturing contextual information. For example, the 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 assume that the correct pronunciation is most likely *itta*, because the pronunciation *ni okonatta* is unusual (行った *okonatta* is seldom preceded by に *ni*). 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), 食器-棚 “plate rack” is pronounced as *shokki-dana*, while the components of this word are individually pronounced as *shokki* (“plate”) and *tana* (“rack”). By considering the compositions of operations, we can capture the pronunciation in the context of a compound word. Our phrasal decoder considers all (i.e. composed and non-composed) operations during the decoding, but longer (composed) operations are generally preferred when available because the phrase count feature usually receives a negative weight.

However, the simultaneous use of these operations of different size may cause a problem when the joint n-gram estimation is applied: because composed operations include multiple non-composed operations, they break the independence assumption of n-gram occurrences in the language model. For example, given a parallel phrase “展覧会に行った／てんらんかいにいった” (went to an exhibition), which is decomposed into “展覧会／てんらんかい, に／に, 行った／いった” by dictionary-based alignments, the joint n-gram language model expects that the occurrence of “に／に ” (non-composed operation) is independent of that of “に-行った／に-いった” (composed operation), but this is not the case. To avoid this, we let the model retain the original operations even after they are composed. As shown in Figure 1, even after the two operations “に→に” and “行った→いった” are merged into a composed operation “に-行った→に-いった”, the joint n-gram probability is still estimated based on the original (non-composed) operations. For efficiency purposes, we only retain the decomposition of the first appearance of each composed operation even if multiple different decompositions are possible.

4 Experiments

4.1 Dictionary

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 a 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 into 325k surface forms after considering *okurigana* (verb inflectional ending) variants. The in-house dictionary consists of a total of 226k words and single-kanji pronunciations. After removing duplicates, the combined dictionary consists of 770k entries. Note that these dictionaries are also used as part of training data.

4.2 Training and Test Data

As described in Section 3, we need word/substring-aligned parallel corpora to train the models. We used three different sources of training data in our experiments. First, following Hatori and Suzuki (2011), we used Wikipedia: following the heuristics described in the paper, we extracted about 460k noisy word-pronunciation pairs from Japanese Wikipedia articles as of January 24, 2010. Of these pairs, we set aside 3k instances for use in development and evaluation, and used the rest for training (referred to as “Wiki-Train”). Secondly, since word-pronunciation pairs extracted from Wikipedia are noisy[4] and mostly consist of noun phrases, we also used a newspaper corpus, which is comprised of 1.4m sentence pairs, referred to as “News-Train”. Finally, for the comparison with KyTea, we use a publicly available corpus, the Balanced Corpus of Contemporary Written Japanese (Maekawa (2008)). Specifically, we use the 2009 Core Data of this corpus, which consists of 37k sentences annotated with pronunciations (referred to as “BCCWJ”).

Our test data consist of six datasets from various domains. Table 1 shows the statistics of these corpora, with the OOV rate estimated using KyTea[5]

[4] We have found that roughly 10% of these instances are invalid word-pronunciation pairs.

[5] We ran KyTea 0.13 with the built-in default model. For

Test set	#Instance	Avg. len.	OOV rate
News-1 (N1)	867	51.8	0.3%
News-2 (N2)	739	44.9	0.3%
Query-1 (Q1)	1,049	3.8	3.5%
Query-2 (Q2)	3,078	5.7	12.7%
Name (PN)	9,170	3.0	23.4%
Wiki (WP)	2,000	4.1	13.7%

Table 1: Statistics of test sets, where ”Avg. len.” is the average length of an instance in the number of characters.

- **News-1**(N1) and **News-2**(N2): collections of newswire articles available as Microsoft Research IME Corpus (Suzuki and Gao, 2005). These articles are from different newspapers from the news corpus we used in training. In preparing these test sets, instances including Arabic and kanji numerals (0,1,···,9, 〇, 一,···, 九), or Roman alphabets are excluded[6].
- **Query-1**(Q1) and **Query-2**(Q2): 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 difficult-to-pronounce words, mostly consisting of person names.
- **Wiki**(WP): manually-cleaned word-pronunciation pairs from Wikipedia, which consists mostly of proper nouns including names of people and locations as well as terms that are difficult to pronounce.

For the tuning of the weights of the 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 Experimental settings

We use our original implementation of the phrasal aligner and decoder, which is also used as our implementation of the substring-based model of Hatori and Suzuki (2011). An ITG-based aligner with EM algorithm (Zhang et al., 2008) is used with monotonic setting; we set the source (kanji) and target (kana) phrase length limits to 1 and 4, and prohibit alignments to a null symbol in either source or target side. The decoder runs with the beam size of 20. The maximum number of composed operations is 4 for the substring-based model of Hatori and Suzuki (2011), and 3 for the proposed 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. We did not use the infrequent operation cut-off. All of these parameters and settings are set based on the preliminary experiments. 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: instances correspond to sentences in News-1/2, and to words or phrases in all other test domains. The statistical significance of the results is given using McNemar’s test.

4.4 Baseline Models

We describe three baseline models that we use as reference in our experiment.

- **Mecab**: *Mecab* version 0.98[7], which is the state-of-the-art morphological analyzer for Japanese that also outputs pronunciations of words (Kudo et al., 2004), with the off-the-shelf IPA Dictionary containing 392k word entries provided at the author’s page.
- **KyTea**: *KyTea* version 0.13[8], which is described in Section 2.2. In our comparison experiment, we run KyTea version 0.13 both as is (using their pre-trained model), and as trained by us to allow the comparison of the framework using the same publicly available training data.
- **HS11**: *HS11* is our reimplementation of the substring-based model by Hatori and Suzuki (2011), which was shown to outperform the substring-based joint trigram model on a Wikipedia test set.

5 Results and Discussion

5.1 Main Results

Table 2 shows the performance of the proposed model along with various baseline models. The first two lines are the result of the off-the-shelf, pre-trained systems. Mecab achieves around or above 80% accuracy on five out of six test sets, although the result on Wiki is below 60% because

News-1/2, the OOV rate in the table is the OOV word rate based on the KyTea’s output. For the other test sets, the figures show the rate of the instances (words or phrases) that contain any OOV word, again based on the KyTea’s output

[6] This is because there exist different standards in how to pronounce them. For example, 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.

[7] http://mecab.sourceforge.net/

[8] http://www.phontron.com/kytea/

Model	N1	N2	Q1	Q2	PN	WP
Mecab	78.8	79.7	88.0	79.8	79.8	55.9
KyTea	83.6	85.9	92.9	85.6	52.9	62.9
HS11	23.3	31.8	87.7	73.3	83.9	64.5
HS11+	37.6	31.8	93.3	82.7	90.5	**72.9**
Proposed	**89.7**	**88.6**	**95.5**	**87.8**	**92.9**	70.2

Table 2: Instance-level accuracy (in %) of pronunciation prediction models. The upper two models use the off-the-shelf models; the lower three models are trained using the same resources: Wiki-Train, News-Train, and the combined dictionary.

the system does not have a mechanism to handle OOV words. The second row shows the result of KyTea using the off-the-shelf "full SVM model"[9], which is trained on several resources including BCCWJ and UniDic. It generally does better than Mecab, but the accuracies on the high OOV rate domains (i.e. Name and Wiki) are still quite low.

The bottom three models are all trained with the same resources: Wiki-Train and News-Train with all the three dictionaries. "HS11" is the substring-based model proposed by Hatori and Suzuki (2011), while "HS11+" is the model enhanced with two additional features: the joint n-gram feature (as described in Section 3.2), and the dictionary feature, whose value is the total length (in souce characters) of words matching any dictionary entry.[10] By comparing these two models, the effectiveness of these features over the model "HS11" is quite clear. However, the accuracy is below 40% on newswire test sets, where each instance is a full sentence. We assume that this is because the substring-based model cannot capture the contextual information that is broad enough, and also is easily affected by noise in the training data. Our proposed model, corresponding to the last line in the table, overcomes this problem and achives the best accuracy in all but one test domain (Wiki), showing the effectiveness and robustness of the dictionary-based approach. We lags behind "HS11+" on Wiki, probably because the dictionary-based model discards many operations that are uncommon, but are still useful for the pronunciation of OOV words in Wikipedia.

Table 3 shows the direct comparison between KyTea and the proposed model trained[11] with exactly the same datasets: BCCWJ, Wiki-Train, and UniDic, all of which are from publicly available resources. Whereas "KyTea (w/noise)" uses all the instances for training, "KyTea (wo/noise)" uses only the instances that are filtered using dictionary-based operations[12]. Note that this cleaning process is also a novel contribution of our work. As is observed from Table 3, this cleaning process resulted in a large improvement in accuracy, with the exception of the Name and Wiki sets. After inspecting the errors manually, 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*". However, this problems seems negligible when a larger dictionary including common pronunciations for person's names is available. In the comparison in Table 2, where the models use a combination of three dictionaries, the dictionary-based model "Proposed" performs better than the substring-based model "HS11+" even on the Name 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	73.8	75.4	**92.8**†	**84.9**†	62.8	64.3

Table 3: Instance-level accuracy (in %) of the models trained on Wiki-Train and BCCWJ with UniDic. "†" denotes a statistically-significant ($p < 0.01$) difference between "KyTea (wo/noise)" and "Proposed".

Overall, the proposed model outperforms "KyTea (wo/noise)" in four out of six test sets, and the differences in the remaining two sets (News-1/2) are not statistically significant. Considering also that the training data is relatively small in this comparison experiment[13], we can conclude that our model has at least a comparable performance to KyTea for the task of pronunciation disambiguation, while achieving a superior performance on the task of pronunciation prediction for OOV words. A manual analysis of the results also showed that our model indeed has an advantage in outputting phonetically natural pronunciation sequences, partially resolving problems related to *on/kun*[14] and *rendaku*, as in 契約-切れ *keiyaku-*

[9] We could not train KyTea with the same dataset as the proposed model uses due to memory limitation.

[10] The dictionary is also used as the training data.

[11] Our training of KyTea is performed as follows: we first train a segmentation model for KyTea 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.

[12] 27.6% of the instances in Wiki-Train is filtered out. This percentage is larger than the noise rate of 10% in this corpus, which Hatori and Suzuki (2011) reported, because the sole use of UniDic does not cover many single-kanji pronunciations, as mentioned later in this paragraph.

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

[14] Pronunciations of kanji are classified into *on* and *kun* pronunciations (corresponding to their origin, Chinese and

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

Table 4: Feature ablation results for the dictionary-based model trained with Wiki-Train, News-Train and the combined dictionary. All the losses in accuracy were statistically significant ($p < 0.01$).

gire (individually pronounced as *keiyaku* and *kire*; "contract expiration"). Although KyTea wrongly output *keiyaku-kire* to this instance, the proposed model was able to output the correct pronunciation by learning that the pronunciation of 切れ tends to be *gire* after the pronunciation *ku*, from other instances such as 句-切れ *ku-gire* (segments in haiku). On the other hand, KyTea is better at capturing generalized context by using a character-type feature, resolving instances such as "ブランド-米" (katakana + *mai*; "brand rice"), while the proposed model wrongly output the most frequent pronunciation *bei* for 米.

5.2 Feature Ablation Experiments

Table 4 shows the results of the feature ablation experiment of the proposed model. As we mentioned in Section 3.2.1, the advantage of the joint n-gram language model is twofold: incorporating smoothed context into word pronunciation disambiguation (which is the dominant problem in News-1/2), as well as incorporating single-kanji pronunciation dependencies into pronunciation prediction for OOV words (considered to be common in Name and Wiki). The improvement observed in these domains suggests that the joint n-gram probability successfully captured these two aspects. The use of composed operations showed large improvement particularly on News-1/2, proving its utility for the pronunciation disambiguation aspect of this task.

5.3 Data Ablation Experiments

Figure 3 shows the performance of the proposed model with respect to the number of News-Train sentences used for training. In this experiment, the model is first trained only with Wiki-Train; then, sentences from News-Train are incrementally added. This can be seen as a process for adapting a word-based model to a fully sentential, disambiguation-capable model. As expected, the accuracy is consistently improved in the news domain as more sentences are added, while the accuracy remains almost unchanged in the rest of the domains, without showing any negative effect by the additional out-of-domain training data. These results suggest that our model is robust and can adapt to new domains with a simple addition of training data.

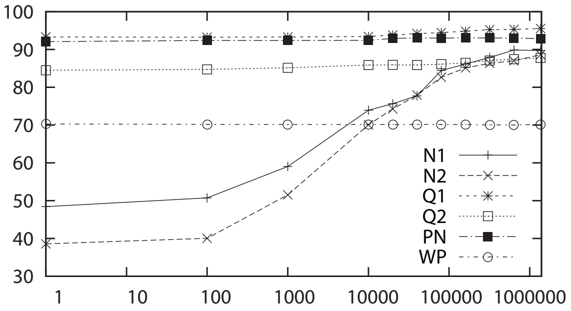

Figure 3: Performance (accuracy in %) of the proposed model with respect to the log of the number of additional training sentences from News-Train.

6 Conclusion

We have presented a unified approach to the task of Japanese pronunciation prediction. Based on the framework of phrasal SMT, our model seamlessly and robustly integrates the task of word pronunciation disambiguation and pronunciation prediction for OOV words. Its basic components are trained in an unsupervised manner, and work in the presence of noise in training data. The model also has potential to adapt to a new domain when additional training data is available. We have performed an extensive evaluation on various test sets, and showed that our model achieves the new state-of-the-art accuracy on the task of Japanese pronunciation prediction.

Looking into the future, we would like to see if the proposed model is effective in a general task of transliteration within a sentential context, which is conceivable as an application of phonetic input (e.g., inputting Arabic using Roman text and converting it automatically into Arabic scripts). On the task of Japanese pronunciation prediction, we are also interested in incorporating class-based features, such as character type information and on/kun dependencies, by using both existing resources and clustering methods.

Acknowledgement

We are grateful to Graham Neubig for providing us with detailed information on KyTea, and to anonymous reviewers for useful comments.

Japanese), each of which tends to be used consecutively.

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*.

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*.

Colin Cherry and Hisami Suzuki. 2009. Discriminative substring decoding for transliteration. In *EMNLP*.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *ACL*.

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 *EMNLP*.

Jun Hatori and Hisami Suzuki. 2011. Predicting word pronunciation in Japanese. In *CICLing 2011, Lecture Notes in Computer Science (6609)*, pages 477–492. Springer.

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

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

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

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

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Appliying conditional random fields to Japanese morphological analysis. In *EMNLP*.

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 *ACL*.

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

Shinsuke Mori, Tetsuro Sasada, and Graham Neubig. 2010b. Language model estimation from a stochastically tagged corpus (in Japanese). *Technical Report, SIG, Information Processing Society of Japan*.

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.

Graham Neubig and Shinsuke Mori. 2010. Word-based partial annotation for efficient corpus construction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*.

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

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*.

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 *ACL*.

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

Hisami Suzuki and Jianfeng Gao. 2005. Microsoft Research IME Corpus. MSR Technical Report No. 2005-168.

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 *HLT-NAACL*.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *ACL*.