# IMPROVING PHONEME AND ACCENT ESTIMATION BY LEVERAGING A DICTIONARY FOR A STOCHASTIC TTS FRONT-END

*Tohru NAGANO, Ryuki TACHIBANA, Nobuyasu ITOH, and Masafumi NISHIMURA*

Tokyo Research Laboratory, IBM Research
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken, 242-8502, Japan
{tohru3,ryuki,iton,nisimura}@jp.ibm.com

## ABSTRACT

Determining the correct phonemes and pitch accents is important for creating natural Japanese speech. We implemented a TTS front-end system based on an $n$-gram model. However, the vocabulary of the word $n$-gram model is limited to the list of the words found in the training corpus, and collecting a very large training corpus is not an easy task. In this paper, we propose using an additional class $n$-gram model to incorporate not only the words found in the training corpus, but the words found in the dictionary to further improve the accuracy. In our experiments, our proposed model relatively improves the accuracy for estimating accents by 16.9% and the accuracy for estimating phonemes by 21.6% compared to the word $n$-gram model.

***Index Terms—*** Interpolated LM, Japanese accent, Word clustering, TTS front-end, Speech synthesis

## 1. INTRODUCTION

The front-end modules of TTS systems assign linguistic and phonetic information to input plain texts, which is critical for creating intelligible and natural speech. For Japanese, the front-end process consists of five sub-processes, word segmentation, part-of-speech tagging, grapheme-to-phoneme conversion, pitch accent generation, and prosodic boundary detection. We proposed a stochastic front-end based on an $n$-gram model [1] to support the first four sub-processes as an extension of the statistical morphological analyzer. This paper focuses on improving the grapheme-to-phoneme and pitch accent generation of our front-end.

A common approach is for the front-end modules to use a TTS dictionary to perform the sub-processes. The TTS dictionary generally contains the spellings, the part-of-speech labels, the phonemes, and the base accents for each word. The base accent of a word is the accent that is used when the word is spoken in isolation. The accent can be changed by the context. We call the accent in a specific context a *context accent*. Hence, the base accent is merely one of the possible accents of the word. Since there are several possible combinations of phonemes and accents, choosing the correct combination for each word depending on the local context is a problem for the front-end modules. A rule-based approach [2] was proposed to handle pitch accent generation in Japanese. The rule-based approach determines the context accent for each word in the context by modifying the base accent of the word applying an appropriate rule chosen from a detailed rule set. A strong point of this method is that the types of pitch accents for words can be represented by a small number of rules. However the maintenance of the rules and the dictionaries is time-consuming, since it is necessary to maintain the consistency of the rules while avoiding side effects. In addition, it requires many exceptions to the rules. In order to overcome this weakness of the rule-based approach, we proposed a stochastic method that can simultaneously handle word segmentation, part-of-speech tagging, grapheme-to-phoneme conversion, and pitch accent generation. The stochastic method has advantages including scalability and ease of domain adaptation compared with rule-based approaches. The accuracies of grapheme-to-phoneme conversion and pitch accent generation with our method were higher than with the rule-based approach in our experiments. If a large enough training corpus is available, the method should be capable of calculating the phonemes and accents of the words depending on their contexts. However, it is essentially impossible to include in the training corpus all of the possible contexts of all of the possible words. When there is a word in the input sentence that is not in the training corpus, the stochastic model uses the dictionary to look up the phonemes and the accents of the word. However the dictionary gives only the base accent, which can be different from the correct accent in that context.

In our work we address this problem by leveraging the advantages of our stochastic model and other rule-based approaches. We propose using an additional stochastic model that serves as a knowledge source similar to the rules and the available information in the dictionary used by the rule-based approaches. The new model is a class $n$-gram model of new classes, each of which is a class of words with the same accentual feature. Words with the same accentual feature are grouped into a class. Not only the words found in the training corpus are grouped, but we also group into these classes the additional words found in the dictionary. With this procedure, the coverage of the model can be made as large as the dictionary, while the coverage of the original stochastic model was limited to the list of words found in the corpus, which is smaller than the dictionary. Therefore, the class $n$-gram model can now be used to predict the accent changes of the word in contexts not found in the training corpus, while the original stochastic model still supports accurate accent estimation for the contexts that are included in the corpus. We provide experimental results for the phoneme and accent estimation task. In the experiments, we compare the interpolated model with the word-based $n$-gram model (baseline) and the accent class-based $n$-gram model.

## 2. BUILDING AN INTERPOLATED LANGUAGE MODEL USING ACCENT CLASS

In this section, we explain the characteristics of Japanese accentuation. And we propose the accent class $n$-gram model by utilizing partial information in the dictionary.

1-4244-1484-9/08/$25.00 ©2008 IEEE

### 2.1. Phonemes and Accents in Japanese

**Front-end task**

The set of tasks to be handled by a front-end for Japanese is segmenting an input text into words and estimating parts-of-speech, phonemes, and accents for each word to facilitate synthesizing a natural voice. The pitch accents of Japanese can take a binary value high `(H)` or low `(L)`, and one value is assigned to each mora. For example, the word “京都” *(Kyoto)* with three morae /kyo,u,to/ has three pitch accents `(H,L,L)`. The input is the concatenated Japanese character string. The output of this process is the combination of the spellings $w$, parts-of-speech $t$, phonemes $\boldsymbol{s}$, and accents $\boldsymbol{a}$, as shown in Table 1. In this paper, a set of items $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$ is defined as a “word”.

**Rule-based Method and Information in TTS dictionary**

In addition to that information, the base accents $\acute{\boldsymbol{a}}$ and additional flags $g$ (bottom of Table.1) can be given in the dictionaries for the rule-based method. The flag for each word in the dictionary is used for choosing an appropriate rule. The selected rule changes the base accent to the context accent. We denote the flag a *base-to-context (BtC) flag g*. The possible variations of the accent can be calculated from the base accent and the BtC flag. Therefore it is difficult to directly estimate the context accent. In the rule-based method, the accent of the word is roughly determined by the combination of the features of the word and the previous words:

$$\boldsymbol{a}_i = rule(\langle t, \acute{\boldsymbol{a}}, g\rangle_i, \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle_{i-h_b}, \cdots, \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle_{i-1}), \qquad (1)$$

where $\langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle_{i-h_b}, \cdots, \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle_{i-1}$ is the history. We call the set of items $f_{acc} = \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle$ an *accentual feature*. Usually $h_b$ is the position of the first prosodic boundary before $\langle t, \acute{\boldsymbol{a}}, g\rangle_i$. The accents are sequentially determined from the head of the word sequence. This is done after part-of-speech tagging.

**Table 1**. Elements of a part of an input sentence “今日京都タワーホテルに泊まる。” (*“Today, I will stay at the Kyoto Tower Hotel.”* in English)

| word index | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| spelling $w$ | 今日<br>*(today)* | 京都<br>*(kyoto)* | タワー<br>*(tower)* | ホテル<br>*(hotel)* | に<br>*(at)* |
| part-of-speech $t$ | NN | PN | NN | NN | IN |
| phonemes $\boldsymbol{s}$ | kyo,u | kyo,u,to | ta,wa,a | ho,te,ru | ni |
| context accents $\boldsymbol{a}$ | `H,L` | `L,H,H` | `H,H,H` | `H,L,L` | `L` |
| base accents $\acute{\boldsymbol{a}}$ | `H,L` | `H,L,L` | `H,L,L` | `H,L,L` | `L` |
| BtC flags $g$ | C0 | C1 | C1 | C1 | I1 |

### 2.2. Building Accent Class $N$-gram Model

Unknown words can be categorized into two classes. The first class is the *dictionary words $\mathcal{V}_d$* that are in the dictionary but not in the training corpus. From the viewpoint of accents, partial information, such as base accents and BtC flags, is available. The second class is the *out-of-vocabulary words $\mathcal{V}_o$* that are in neither the dictionary nor the training corpus. There is no information about accentuation for the words in this class. Known words is defined as the *in-corpus words $\mathcal{V}_k$* that are in the training corpus.

The main idea to improve the prediction of accents and phonemes is to reduce the number of unknown words by utilizing the information in the dictionary. In order to address this problem, we make two language models allowing the model to use as much partial information as possible:

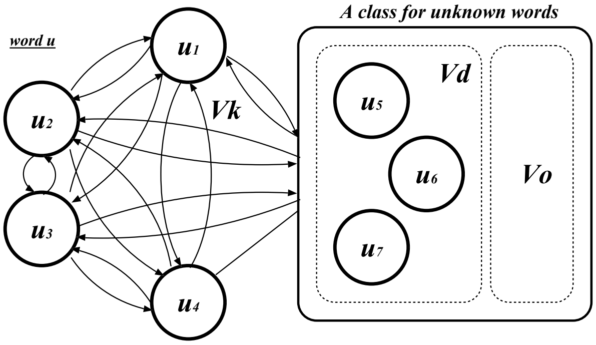

**Fig. 1**. Word $n$-gram model (bigram)

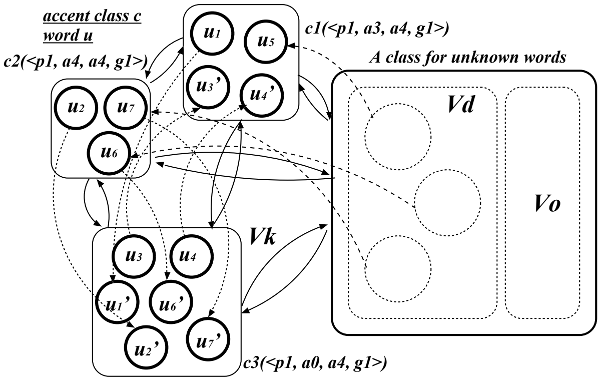

**Fig. 2**. Accent class $n$-gram model (bigram)

- Word $n$-gram model
  This model is the word-based stochastic model (Fig.1) we originally proposed in [1].
- Accent class $n$-gram model
  This model predicts the contextual accent changes of words. Words with the same accentual feature $f_{acc} = \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g\rangle$ are grouped into a class. Each word of both the in-corpus words $\mathcal{V}_k$ and the dictionary words $\mathcal{V}_d$ is grouped into a class. This is done in the following steps:

  1. Prepare a class for each combination of the accentual feature $f_{acc}$ ($c_1$, $c_2$, and $c_3$ in Fig.2).
  2. Each of the in-corpus words is grouped into a class according to the accentual feature ($u_1$, $u_2$, $u_3$, and $u_4$ in Fig.2).
  3. For the dictionary words, assuming the context accents are same as the base accents, the words are grouped into the classes ($u_5$, $u_6$, and $u_7$ in Fig.2).
  4. Both for the in-corpus words and the dictionary words, assuming contextual accent changes, multiple copies of each word are generated with different context accents. And the generated copies are grouped into classes ($u'_1$, $u'_2$, $u'_3$, $u'_4$, $u'_6$, and $u'_7$). The possible classes can be calculated from the base accent and the BtC flag.
  5. Count the class uni-grams and bi-grams using the word-class map built by these procedures.

6. Calculate the word probabilities for each class. Nonzero probabilities are assigned to the copied words.

## 2.3. Estimating correct word sequence

### 2.3.1. Word $N$-gram Model

Our original method outputs the sequence of words with the highest probability under the constraint that the concatenation of the spellings is equal to the input sentence $\boldsymbol{x} = x_1 x_2 \cdots x_l = \boldsymbol{w}$:

$$\hat{\boldsymbol{u}} = \mathbf{argmax}\, P(u_1 u_2 \cdots u_h | x_1 x_2 \cdots x_l), \tag{2}$$

The probability of the word sequence in Equation (2) is calculated from the training corpus based on the word $n$-gram model:

$$P_u(u_1 u_2 \cdots u_h) = \prod_{i=1}^{h+1} P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}), \tag{3}$$

where $k = n-1$, $u_{h+1}$ is the special symbol indicating the end of the sentence.

### 2.3.2. Accent Class $N$-gram model

With the class $n$-gram model, the probability of a word sequence in Equation (2) is calculated by multiplication of the class $n$-gram probability and the probability of each word in the class.

$$\begin{aligned} & P_c(u_1 u_2 \cdots u_h) \qquad (4) \\ = & \prod_{i=1}^{h+1} P(u_i | c(u_i)) P(c(u_i) | c(u_{i-k}) \cdots c(u_{i-2}) c(u_{i-1})), \end{aligned}$$

where $c(u)$ is a class that contains a set of word $u$. The probability of $u$ in $c$ is calculated by counting words $u$ in the training corpus:

$$\begin{aligned} & P(u|c(u)) \qquad (5) \\ = & \begin{cases} \alpha \dfrac{N(u, c(u))}{\sum_{u', N(u', c(u')) \neq 0} N(u', c(u'))}, & \text{if } N(u, c(u)) \neq 0 \\ (1-\alpha) \dfrac{1}{\sum_{u', N(u', c(u'))=0} 1} & \text{otherwise} \end{cases} \end{aligned}$$

where $0 \leq \alpha \leq 1$. In this equation, the probability for each word $u$ that is found in the corpus is calculated based on the count $N(u, c(u))$ which is the number of times the word is found in the training corpus. Meanwhile, a small value is given for the probabilities of the words not found in the corpus. Those words are the words of the dictionary words and the words generated by assuming context accents. The parameter $\alpha$ is a predefined coefficient to spare low probabilities for the words not found in the corpus.

### 2.3.3. Interpolated Model

To use the accurate accent estimation of the word $n$-gram model and the wide coverage of the class $n$-gram model, an interpolation technique is used. In this method, the probability of the word sequence in Equation (2) is calculated by:

$$P(u_1 u_2 \cdots u_h) = \lambda_u P_u(u_1 u_2 \cdots u_h) + \lambda_c P_c(u_1 u_2 \cdots u_h).$$

where $0 \leq \{\lambda_u, \lambda_c\} \leq 1$, $\lambda_u + \lambda_c = 1$. The interpolation coefficients $\lambda_u$ and $\lambda_c$ are estimated by the deleted interpolation method, and optimized by using the EM algorithm [3]. The algorithm renews $\lambda$ according to its likelihood on the test corpus.

## 3. EXPERIMENTAL EVALUATION

We conducted experiments to evaluate the performance of the interpolated models explained in Section 2. In this section, we describe the conditions and the results of the experiments and evaluate our new method.

### 3.1. Conditions

The corpus we used in the experiments contains Japanese sentences extracted from newspaper articles, TV news, telephone conversation, and so on. Each sentence in the corpus was segmented into words and each word $w$ was annotated with its part-of-speech $t$, its phoneme sequence $\boldsymbol{s}$, and its accent sequence $\boldsymbol{a}$. The total number of sentences was about 60,000 sentences. 1/10 of the learning corpus was used for parameter estimation, and the rest of it was used for building the language model. The total size of the test corpus was 200 sentences. The average length of the Japanese words was 1.68 characters for the entire corpus. The dictionary we used in the experiments contains 160K words.

**Table 2**. corpus

| | #sentences | #words | #chars |
|---|---|---|---|
| learning | 59,351 | 1,045,803 | 1,755,004 |
| test | 200 | 5,519 | 8,349 |

### 3.2. Details of the Models

In the experiments we compared the accuracy of the word $n$-gram model and the accent class $n$-gram model, and the interpolated model that contains both the word $n$-gram and accent class $n$-gram models.

- ***Word***
  Regards a sentence as a sequence of words and estimates the most probable word $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$ sequence based on a bigram model. The model contains of 20,149 classes. (baseline)
- ***AccentClass***
  Regards a sentence as a sequence of classes that consist of words that have same accentual feature $f_{acc} = \langle t, \boldsymbol{a}, \acute{\boldsymbol{a}}, g \rangle$. This model contains 1,062 classes.
- ***Interpolated***(Proposed)
  Interpolated model of the model ***Word*** and the model ***AccentClass***.

In addition to these three models, we compared them with these two models using conventional methods.

- ***Rule***
  A rule-based model. An expert writes rules for word segmentation, grapheme-to-phoneme conversion, and accent annotation.
- ***AutoClass***
  A class-based bigram model. This model uses an automatic clustering method based on mutual information [4]. This model contains 14,803 classes.

All of the models use the same corpus and use the same vocabulary.

### 3.3. Evaluation and Discussion

In order to evaluate our model, we compared the five methods in Section 3. To measure the assignment accuracy, we used the Mora Error Ratio, (MER) which is similar to the Character Error Ratio.

For example, the annotation result of "日本人" (*Japanese person*) can be ⟨ 日本, PN, /ni,ho,n/, (L,H,H) ⟩ ⟨ 人, NN, /ji,n/, (H,L) ⟩. The MER for accents is calculated by counting the correct units of the combinations $\langle s, a\rangle$. The MER for the phonemes is calculated by counting the correct $\langle s\rangle$s. In the first output above, the /ni,ho,n,ji,n/, (L,H,H,H,L) is formatted to the sequence (ni-L, ho-H, n-H, ji-H, n-L) to calculate the accuracy of the accents, and formatted to the sequence (ni, ho, n, ji, n) to calculate the accuracy of the phonemes. The test corpus contains 10,826 units in 200 sentences.

**Table 3**. Summary of predictive powers and accuracy of the models.

| model | entropy | accents | | phonemes | |
|---|---|---|---|---|---|
| | $H_{test}$ | MER(%) | | MER(%) | |
| ***Word*** | 4.8962 | 9.64 | - | 1.20 | - |
| ***AccentClass*** | 5.5656 | 12.17 | (-26.2) | 2.05 | (-70.8) |
| ***Interpolated*** | 4.9281 | **8.01** | (16.9) | **0.94** | (21.6) |
| ***Rule*** | *N/A* | *11.09* | (-15.0) | *1.28* | (-5.00) |
| ***AutoClass*** | *4.8953* | *9.92* | (-2.90) | *1.18* | (1.66) |

The results are shown in Table 3. The $H_{test}$ is the character-based cross-entropy of the test corpus. The cross-entropy of the proposed model ***Interpolated*** is not improved compared with that of the baseline model ***Word***. The accuracy for the accent estimation is shown in the third column. The numbers in parentheses in the third column are the ratios of improvement compared with the baseline model ***Word***. The accuracies for phonemes are shown in the fourth column.

For the proposed model ***Interpolated***, the MER for phonemes decreased from 9.64% to 8.01% compared with the baseline model ***Word***. The MER for accent also decreased from 1.20% to 0.94%. These are equivalent to about 16.9% and 21.6% error reduction, respectively. Most importantly, not only the performance in estimating accents but also the performance in estimating phonemes is improved by interpolating with the accent class $n$-gram model, even though the vocabulary for the phoneme estimation is the same as the baseline model. This means that the correct sequence of accents in the context can provide the correct sequence of phonemes, and vice versa. Compared with the model ***AutoClass*** that uses an automatic clustering method that classifies only the words found in the training corpus, the performance of ***AutoClass*** is almost the same as the baseline model ***Word***. The result shows the automatic clustering method is not effective for this task. The accuracies of accents and phonemes with the proposed model ***Interpolated*** is higher than with the rule-based model ***Rule***.

## 4. RELATED WORKS

We proposed an interpolated model of the word $n$-gram model and the accent class $n$-gram model which allows us to expand the coverage of the model by grouping words with the same accentual features into a class. From a point of view of assigning words to classes, Brown et al. [4] discussed several statistical algorithms for assigning words to classes considering the co-occurrence with other words. Mori et al. [5] used an average test set perplexity as an objective function to build a class based $n$-gram model. They improved their language models using the contextual information such as the co-occurrence and the history of the word sequence. Although our attempt also was to use the contexts of the classes for improving the model, our method is aimed at predicting by using the classes the contextual changes of the dictionary words for which contexts are not available, while their research mainly focused on grouping only *in-corpus words* into classes. As for combination of a statistical model and heuristic information, Linares [6] proposed a general hybrid language model defined as an interpolated model of a word $n$-gram model and a word stochastic grammatical model to capture the local relations between words and the global relation. While the Linares' method incorporated grammars, our method incorporates dictionaries with the stochastic model. We consider that the use of the dictionaries is practically important since there are large available dictionaries. To cope with the task for assigning phonemes and accents for Japanese, rule based approaches [2] and a method generating rules by using C4.5 [7] has been proposed. Minematsu et al. [8] has proposed a method for assigning accent using Conditional Random Fields (CRFs). In contrast with these methods, we use a fully stochastic method that can simultaneously handle word segmentation, part-of-speech tagging, grapheme-to-phoneme conversion, and pitch accent generation because in some languages some of these tasks are inseparable. In our experiments, accuracy by our stochastic method was higher than rule-based method.

## 5. CONCLUSION

In this paper, we have presented a new method for improving the accuracy of the estimation of accents and phonemes by combining the word-based $n$-gram model and the accent class-based $n$-gram model. The word-based $n$-gram model reflects the details of the word sequences. However it is hard to cover the vocabulary of the test corpus with the word-based $n$-gram model, since the accents of the words vary with the context. We built an accent class-based $n$-gram model to incorporate the vocabulary of the dictionary, and add the existing classes derived from the vocabulary of the known words. Adding the accent class-based $n$-gram models improves the language model using the same corpus. Our experiments using our proposed model show a 16.9% error reduction for estimating accents and a 21.6% error reduction for estimating phonemes compared to the word $n$-gram model.

## 6. REFERENCES

[1] T. Nagano, S. Mori, and M. Nishimura, "A Stochastic Approach to Phoneme and Accent Estimation," in *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH2005)*, 2005, pp. 3293–3296.

[2] Y. Sagisaka and Y. Sato, "Accentuation Rules for Japanese Word Concatenation," *Transactions of IEICE of Japan*, vol. J66-D, no. 7, pp. 849–856, 1983, (In Japanese).

[3] C. Stanley and J. Goodman, "An Empirical study of smoothing techniques for language modeling," in *Proceedings of the Association for Computational Linguistics 34th Annual Meeting (ACL1996)*, 1996, pp. 310–318.

[4] P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer, "Class-Based N-Gram Models of Natural Language," *Computational Linguistics*, vol. 18, no. 4, pp. 466–479, 1992.

[5] S. Mori, M. Nishimura, and N. Itoh., "Word Clustering for A Word Bi-gram Model," in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP1998)*, 1998, pp. 310–318.

[6] D. Linares, J. Benedí, and Y. Sánchez, "A hybrid language model based on a combination of N-grams and stochastic context-free grammars," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, no. 2, pp. 113–127, 2004.

[7] S. Seto, M. Morita, T. Kagoshima, and M. Akamine, "Automatic rule generation for linguistic features analysis using inductive learning technique: linguistic features analysis in TOS drive TTS system," in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP1998)*, 1998, pp. 1059–1063.

[8] N. Minematsu, R. Kuroiwa, and K. Hirose, "CRF-based statistical learning of Japanese accent sandhi for developing Japanese text-to-speech synthesis systems," in *Proceedings of the 6th ISCA Workshop on Speech Synthesis*, 2007, pp. 148–153.