A Stochastic Approach to Phoneme and Accent Estimation

Tohru NAGANO, Shinsuke MORI, and Masafumi NISHIMURA

IBM Research, Tokyo Reseach Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato-shi, 242-8502, Japan
{tohru3,forest,nisimura}@jp.ibm.com

Abstract

We present a new stochastic approach to estimate accurately phonemes and accents for Japanese TTS (Text-to-Speech) systems. Front-end process of TTS system assigns phonemes and accents to an input plain text, which is critical for creating intelligible and natural speech. Rule-based approaches that build hierarchical structures are widely used for this purpose. However, considering scalability and the ease of domain adaptation, rule-based approaches have well-known limitations. In this paper, we present a stochastic method based on an n-gram model for phonemes and accents estimation. The proposed method estimates not only phonemes and accents but word segmentation and part-of-speech (POS) simultaneously. We implemented a system for Japanese which solves tokenization, linguistic annotation, text-to-phonemes conversion, homograph disambiguation, and accents generation at the same time, and observed promising results.

1. Introduction

A TTS front-end process which is a prior step to speech synthesis (a TTS back-end) plays an important role for creating intelligible and natural speech. The role of the front-end system is to analyze an input text and to assign some features that facilitate synthesizing a natural voice. In particular, estimating phonemes and accents are critical to system performance. Above all, these errors spoil naturalness. Errors at assigning phonemes and accents occasionally change a meaning of the word and a semantic structure of a text, and impede the correct understanding of the spoken utterance.

Hierarchical prosodic structure generation, which employs dynamic programming, decision trees, and various kinds of statistical methods, is widely used to solve these problems [2]. These methods generate syntactic and prosodic structure step-by-step, such as lexical words with part-of-speech (POS), prosodic words, prosodic phrases, and intonation phrases. However, in some languages some of these problems are inseparable. In English and many other European languages most of the phonetically or prosodically ambiguous graphemes have more than one possible grammatical or semantic category, depending on its phonemes or prosody. For example, there are two prosodies for *"project"* corresponding to on a noun and a verb in English.

Therefore, we will integrate as many disambiguation modules as possible into a single module based on stochastic methods, instead of solving each problem independently.

In this paper, we propose a method which enables the model to estimate phonemes and accents simultaneously. The system solves word segmentation [1], text-to-phonemes conversion, homograph disambiguation, and accent generation at the same time. In the experimental evaluation, we compared the performance of estimating phonemes and accents with a rule-based system that incorporates a stochastic POS tagger, a rule-based prosodic phrase generator, and a rule-based accent generator. As a result of experiments, accuracy of estimating phonemes and accents by our method was higher than the rule-based approach.

[1] Word segmentation, which segments a text into words, is the most fundamental process of Japanese, Chinese, etc.

2. Phonemes and Accents in Japanese

The set of problems to be solved by a front-end is segmenting a input text into words and estimating phonemes and accents for each word. In this section, we explain the characteristics of Japanese accentuation. And a prior art for accentuation method for Japanese is briefly described.

2.1. Word Accents in Japanese

The pitch accents of Japanese can take a binary value high (H) or low (L), and is assigned to each mora. For example, a word *"Kyouto" (Kyoto)* that has three morae (kyo,:,to) has three pitch accents (H,L,L).

Table 1: *Phonemes and Accents in the dictionary*

surface w	POS t	phonemes $\boldsymbol{s}$	accents $\boldsymbol{a}$
Kyouto (Kyoto)	prop	kyo : to	H L L
tawa- (tower)	noun	ta wa :	H L L
hoteru (hotel)	noun	ho te ru	H L L

The sequence of pitch accents for the word that has a combination of features $\langle w, t, \boldsymbol{s} \rangle$ (where w is the spelling of a word, t is POS, $\boldsymbol{s}$ is the phoneme sequence), is determined statically by a dictionary. The sequence of accents in the dictionary shows the accents where the word appears individually. However, the sequence of accents varies according to its context. The sequence of accents (H,L,L) of *"Kyouto"* is changed into (L,H,H) in a compound word *"Kyouto tawa-"* (*Kyoto tower* in English), (see Table 2). Furthermore, for a compound word that consists of three words (see Table 3), *"Kyouto tawa-hoteru" (Kyoto tower hotel)*, the accents of the *"tawa-" (tower)* is (H,H,H) is different from both the dictionary's and the sequence of accents in a compound word *"Kyouto tawa-"*. The incorrect accents lead a listener to misunderstand context and syntactic structure of the text. For example, the listener may understand "Kyouto tawa-" with a sequence of accents (H, L, L)(H, L, L) as "Kyou to tawa-" that means *today and tower*.

As described above, accents of the word varies according to its context. The front-end process has to estimate correctly the phoneme sequence and the accents sequence considering

Table 2: *Accents of "Kyouto" and "tawa-" in a compound word "Kyouto tawa-"*

surface	w	*Kyouto (Kyoto)*			*tawa- (tower)*		
POS	t	prop			noun		
phonemes	$\boldsymbol{s}$	kyo	:	to	ta	wa	:
accents	$\boldsymbol{a}$	L	H	H	H	L	L

Table 3: *Accents of "Kyouto", "tawa-" and "hoteru" in a compound word "Kyouto tawa- hoteru"*

surface	w	*Kyouto (Kyoto)*			*tawa- (tower)*			*hoteru (hotel)*		
POS	t	prop			noun			noun		
phonemes	$\boldsymbol{s}$	kyo	:	to	ta	wa	:	ho	te	ru
accents	$\boldsymbol{a}$	L	H	H	H	H	H	H	L	L

its context. It means, the problem to be solved by the system is estimating correct triplets $\langle w, \boldsymbol{s}, \boldsymbol{a}\rangle$ from a given character sequence $\boldsymbol{x}$. If estimating phoneme sequence or accents sequence is incorrect, the listener would fail to disambiguating homonyms depending on its phonemes or prosody. And it causes misunderstanding the meaning of the spoken utterance. Thus, estimating phonemes and accents is an important task.

2.2. Prior Art

There was an attempt at solving this problem by rule sets. Sagisaka [3] gives a detailed rule set for accentuation in Japanese. The model regards a sentence as a sequence of accents phrases $\boldsymbol{v} = (v_1 v_2 \cdots v_l)$. A procedure for accents generation is as follows:

1. Determine $\langle w, t, s\rangle$ for each word by using a POS tagger and a phoneme annotator.
2. Segment the sequence of words $\boldsymbol{w} = (w_1 w_2 \cdots w_h)$ in the sentence into one or more than one accents phrases $\boldsymbol{w}_1^h \mapsto \boldsymbol{v}_1^l$ by considering POS.
3. *Shift* the accents in each accent phrase $\boldsymbol{v}_i (1 \leq i \leq l)$. A shift function for each word in $\boldsymbol{v}_i$ can be acquired by looking up a dictionary.

A weak point of the rule-based method is that the rule-based method costs the maintenance of rules and dictionaries. It is a time-consuming work to keep consistency of rules with avoiding side effects. Furthermore, it requires many detailed rules. And it depends on accuracy of the POS tagger and a POS system of the POS tagger. This approach is highly language-dependent.

3. Fully Stochatic Approach

As described above, the front-end system of a TTS system for Japanese has to solve many problems. In this section, we propose a fully stochastic approach to solve four major problems in TTS: tokenization, POS-tagging, phoneme annotation, and accent annotation.

3.1. Morphological Analyzer Based on an N-gram Model

Initially the morphological analysis (tokenization and POS tagging) was solved by rule-based systems like other natural language processing (NLP) problems. Later inspired by the speech recognition research which used a statistical language model, the POS tagging for English or other European languages, whose words are delimited by white spaces, was solved by HMM-based models. Nagata [7] extended this method to languages that have no apparent word boundaries, such as Japanese, Chinese, etc. His initial method used a POS-based n-gram model, but, the state-of-the-art morphological analyzers use an n-gram model based on pairs of a spelling of a word w and its POS t as follows:

$$P(\langle w_1, t_1\rangle\langle w_2, t_2\rangle \cdots \langle w_h, t_h\rangle) = \prod_{i=1}^{h+1} P(\langle w_i, t_i\rangle | \langle w_{i-k}, t_{i-k}\rangle \cdots \langle w_{i-1}, t_{i-1}\rangle),$$

where $k = n - 1$, $\langle w_i, t_i\rangle (i \leq 0)$ is a special symbol indicating the beginning of the sentence, and $\langle w_{h+1}, t_{h+1}\rangle$ is another special symbol indicating the end of the sentence. The stochastic morphological analyzer outputs the sequence of pairs with the highest probability under the constraint that the concatenation of the spellings is equal to the input sentence $\boldsymbol{x} = x_1 x_2 \cdots x_h = \boldsymbol{w}$:

$$(\langle w_1, t_1\rangle\langle w_2, t_2\rangle \cdots \langle w_h, t_h\rangle) = \textbf{argmax}\, P(\langle w_1, t_1\rangle\langle w_2, t_2\rangle \cdots \langle w_h, t_h\rangle | x_1 x_2 \cdots x_h).$$

3.2. Front-end Based on an N-gram Model

In order to solve the major problems of a TTS front-end simultaneously, we propose an extension of the statistical morphological analyzer. First we define the unit of the n-gram model u as a quadruplet of spelling of a word w, its POS t, its phoneme sequence s, and its accent sequence a, that is $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a}\rangle$. Then we introduce an n-gram model based on this unit. The probability of a unit sequence $\boldsymbol{u}$ by the model M_u is as follows:

$$M_u(u_1 u_2 \cdots u_h) = \prod_{i=1}^{h+1} P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}). \quad (1)$$

Similar to the morphological analyzer, our fully stochastic front-end outputs the sequence of units with the highest probability under the constraint that the concatenation of the spellings is equal to the input sentence $\boldsymbol{x} = x_1 x_2 \cdots x_h = \boldsymbol{w}$:

$$\hat{\boldsymbol{u}} = \textbf{argmax}\, M_u(u_1 u_2 \cdots u_h | x_1 x_2 \cdots x_h). \quad (2)$$

This solution search problem is solved efficiently using dynamic programming method [8]. The computation complexity is linear to the number of characters contained in the input sentence.

3.3. Unknown Word Model

Normally the probability in Equation (1) is estimated from an annotated corpus using maximum likelihood estimation. In this case, the probability of a sentence containing unknown words according to the model M_u is equal to zero and a unit sequence containing unknown words is never selected as the solution by Equation (2), even when the sequence is the correct solution. In real applications, however, input sentences sometimes contain unknown words. Thus the front-end has to be able to recognize unknown words and guess their POS, phoneme sequence, and accent sequence.

In order to cope with the unknown word problem, we introduce a special symbol $\texttt{UNK}_t$ representing all units whose POS is t out of vocabulary $\mathcal{V}$, which is a set of quadruplets. And P in Equation (1) is divided into two cases:

$$P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}) = \begin{cases} P(u_i | u_{i-k} \cdots u_{i-2} u_{i-1}) & \text{if } u_i \in \mathcal{V} \\ P(\texttt{UNK}_{t_i} | u_{i-k} \cdots u_{i-2} u_{i-1}) M_x(u_i | t_i) & \text{if } u_i \notin \mathcal{V}, \end{cases} \quad (3)$$

where M_x is an unknown word model.

An unknown word model must have the following functions:

- generate all possible character strings with a probability greater than zero, [2]
- guess the most probable POS, phoneme sequence, and accent sequence given the unknown words spelling.

We propose n-gram models for each POS based on pairs of character and phoneme sequence $\langle x, \boldsymbol{s}\rangle$:

$$M_x(\langle x_1, \boldsymbol{s}_1\rangle\langle x_2, \boldsymbol{s}_2\rangle \cdots \langle x_{h'}, \boldsymbol{s}_{h'}\rangle|t) \quad (4)$$
$$= \prod_{i=1}^{h'+1} P(\langle x_i, \boldsymbol{s}_i\rangle|\langle x_{i-k}, \boldsymbol{s}_{i-k}\rangle \cdots \langle x_{i-1}, \boldsymbol{s}_{i-1}\rangle, t).$$

The accent sequence of unknown word is fixed to be `LHH...H` [3] which was the most frequent one (37.28%) [4] in a learning corpus. Finally the unknown word model M_x of our fully statistical front-end returns the following value as the generation probability of quadruplet $u = \langle w, t, \boldsymbol{s}, \boldsymbol{a}\rangle$ given t:

$$M_x(u|t) = \begin{cases} M_x(\langle x_1, \boldsymbol{s}_1\rangle\langle x_2, \boldsymbol{s}_2\rangle \cdots \langle x_{h'}, \boldsymbol{s}_{h'}\rangle|t) \\ \qquad\qquad \text{if } \boldsymbol{a} = \texttt{LHH...} \\ 0 \qquad\qquad \text{otherwise,} \end{cases} \quad (5)$$

where $w = x_1 x_2 \cdots x_{h'}$ and the length of the phoneme sequence is equal to that of the accent sequence ($|\boldsymbol{s}| = |\boldsymbol{a}|$).

Finally our statistical front-end calculates the probabilities of the solution candidates based on Equations (1),(3), and (5) and selects the most probable solution by Equation (2).

3.4. Parameter Estimation

The parameters in Equation (3) are estimated based on maximum likelihood estimation from a corpus whose sentences are segmented into words annotated with their POS, phoneme sequence, and accent sequence. The learning corpus is divided into nine parts and quadruples appearing only in a single partial corpus are replaced with the unknown word symbol $\texttt{UNK}_t$ according to the POS. The n-gram probabilities in Equation (3) are interpolated with lower ones. The interpolation coefficients are estimated by the deleted interpolation technique [9] using nine partial corpora.

The parameters in Equation (4) are estimated from the low frequent quadruplets which are replaced with the unknown word symbol at the step of the quadruplet n-gram probability estimation. From these low frequent quadruplet examples, pairs of character sequence and phoneme sequence are extracted according to the POS to make a learning corpus for each POS. The character sequence and the phoneme sequence of a pair is aligned automatically by looking the word up in a dictionary containing all possible phoneme sequences for each character. The parameters in Equation (4) for each POS t are estimated from uniquely aligned examples [5]. The n-gram probabilities in Equation (4) are also interpolated with lower ones.

[2] This condition guarantees that our front-end outputs a solution for any input sentence.

[3] The first accent is `L`. Following accents are `H`. The length of accents `H` is equal to $|\boldsymbol{a}| - 1$.

[4] The accent sequence can also be guessed by an unknown word model. This may improve the performance of the front-end.

[5] Only a few examples had an alignment ambiguity.

4. Evaluation

Experiments on the Japanese corpus were conducted to evaluate the performance of the fully stochastic method explained in Section 3.

4.1. Corpus

The corpus we used in the experiments contains Japanese sentences extracted from news paper articles, TV news, telephone dialogs, and so on. So the corpus contains various words, expressions, and speech phenomena. Each sentence in the corpus was segmented into words and each word w is annotated with its POS t, its phoneme sequence $\boldsymbol{s}$, and its accent sequence $\boldsymbol{a}$. A sentence contains 21.6 words on average and the average length of the phoneme sequence of a word is 1.91. For training we used $8,800$ sentences and the models were tested on 150 sentences (see Table 4).

Table 4: *Size of corpus*

	#sentences	#words	#chars
learning	8,800	190,318	285,082
test	150	2,130	3,170

4.2. Details of the Models

We built the models according to three approaches as follows:

WTS+A_{rule} Rule-based accent generator

1. Regards a sentence as a sequence of words and estimates the most probable triplet ($\langle w, t, \boldsymbol{s}\rangle$) sequence based on an bi-gram model.
2. Estimate boundaries of the prosodic phrases by using hand-crafted rules. The number of rules is about $1,000$.
3. Assign accents for each word in each prosodic phrase based on the method described in Section 2.2.

WTSA Fully stochastic model

- Regards a sentence as a sequence of words and estimates the most probable quadruplet ($\langle w, t, \boldsymbol{s}, \boldsymbol{a}\rangle$) sequence based on a bi-gram model.

WSA Fully stochastic model (without POS)

- Regards a sentence as a sequence of words estimates the most probable triplet ($\langle w, \boldsymbol{s}, \boldsymbol{a}\rangle$) sequence based on a bi-gram model.

The only difference between **WTSA** and **WSA** is whether the model incorporates POS or not. This is to investigate the necessity of POS annotation to the learning corpus. Note that POS information is not always necessary for a TTS back-end.

4.3. Evaluation

Table 5 shows the performances of estimation of phonemes and accents. **WTSA** attains the highest accuracy 90.26% among these models. Accuracy of **WSA** is 89.72%. It is 0.54% lower than accuracy of **WTSA** . Results conclude that fully stochastic models **WTSA** and **WSA** , perform better than the rule-based model **WTS+**A_{rule} . Part-of-speech is an effective feature for phoneme and accent estimation. The right most column in Table

Table 5: *Model vs. accuracy in word segmentation, POS tagging, phoneme estimation, and accent estimation*

		#dist-words	word seg. $\langle w \rangle$	word seg. & POS $\langle w, t \rangle$	word seg. & phonemes $\langle w, \boldsymbol{s} \rangle$	**word seg. & pho.&acc.** $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$	accents (approx.)& $\langle \boldsymbol{a} \rangle \sim \langle w, \boldsymbol{s}, \boldsymbol{a} \rangle \vert \langle w, \boldsymbol{s} \rangle$
WTS+A_{rule}	$\langle w, t, \boldsymbol{s} \rangle$	15,723	*97.61*	*96.08%*	97.69%	**89.53%**	91.65%
WTSA	$\langle w, t, \boldsymbol{s}, \boldsymbol{a} \rangle$	21,164	*97.87*	*95.79%*	97.45%	**90.26%**	92.63%
WSA	$\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$	19,560	*97.64*	*N/A*	97.08%	**89.72%**	92.42%

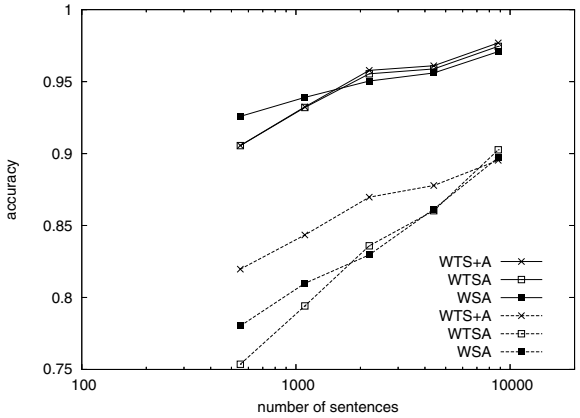

Figure 1: *Learning curve for phonemes and accents. Solid lines represent accuracy of* $\langle w, \boldsymbol{s} \rangle$ *as a reference. Dotted lines represent accuracy of* $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$.

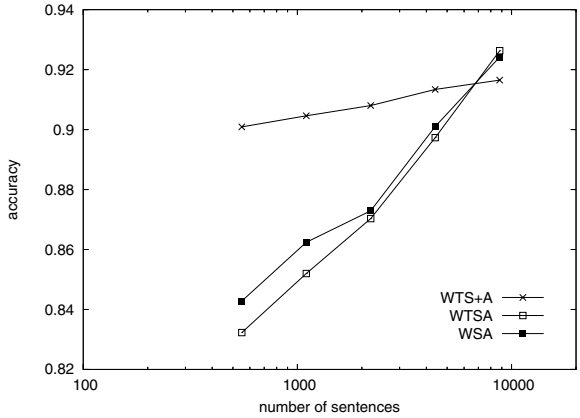

Figure 2: *Learning curve for accents. Solid lines represent approximate accuracy for accents :* $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle \vert \langle w, \boldsymbol{s} \rangle$.

5 represents an approximate value of accuracy of accentuation : $accuracy(\langle a \rangle) \sim accuracy(\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle \vert \langle w, \boldsymbol{s} \rangle)$. **WTSA** attains the highest accuracy 92.63%.

Figures 1 and 2 show the relations between the training corpus size and accuracy. In Table 5, the right most points of each line accord with fifth column $\langle w, \boldsymbol{s} \rangle$ and sixth column $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$, respectively. Learning curves of $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ for **WTSA** and **WSA** intersect a curve for **WTS+**A_{rule} at about $8,000$ sentences. The inclination of leaning curves implies that accuracy of full stochastic models will expand a margin against the rule-based model. From the results displayed in Figure 2, approximate accuracy of accents for **WTSA** and **WSA** are approaching to 1.0 sharply, as contrasted with **WTS+**A_{rule} that performs nearly flat. As is to be expected, accuracy of the phonemes and accents $\langle w, \boldsymbol{s}, \boldsymbol{a} \rangle$ for **WTSA** and **WSA** is getting closer to accuracy of phonemes, according to the size of the corpus.

About POS, small drops (0.37% for phonemes and 0.54% for accents) of accuracy for the model without part-of-speech **WSA** in performances can be seen. Annotating a little large corpus without part-of-speech, and annotating a little small corpus with part-of-speech are trade-off.

5. Conclusion

In this paper, we present a novel method for accurate phoneme and accent estimation. Our n-gram based stochastic model can solve tokenization, linguistic annotation, text-to-phonemes conversion, homograph disambiguation, and accent generation simultaneously. Results from our experimental studies show that the accuracy of a fully stochastic model exceeds the accuracy of a rule-based method. Considering scalability and domain adaptation, the fully stochastic system is an adaptive framework that requires only a corpus.

6. References

[1] Pan, S. and Hirschberg, J., "Modeling local context for pitch accent prediction," Proceedings of ACL, pp. 233-240, 2000.

[2] Shi, Q. and Fischer, V., "A comparison of statistical methods and features for the prediction of prosody prosodic structures," Proceedings of ICSLP, ThA1404p, 2004.

[3] Sagisaka, Y. and Sato, H., "Accentuation Rules for Japanese Word Concatenation," Transaction of IEICE of Japan, Vol. J66-D, No. 7, 1983. (In Japanese)

[4] Beckman, M. and Pierrehumbert, J., "Japanese prosodic phrasing and intonation synthesis," Proceedings of ACL, P86-1025, 1986.

[5] Klein, E., "A constraint-based approach to English prosodic constituents," Proceedings of ACL, pp 217-224, 2000.

[6] Marsi, E., et al, "Learning to predict pitch accents and prosodic boundaries in Dutch," Proceedings of ACL, pp 489-496, 2003.

[7] Nagata, M., "A stochastic Japanese morphological analyzer using a Forward-DP Backward-A* N-Best search algorithm," Proceedings of Coling, pp 201-207, 1994.

[8] Cormen, T., Leiserson, C., and Rivest, R., "Introduction to Algorithms," The MIT Press, 1990.

[9] Jelinek, F., "Self-organized language modeling for speech recognition," Technical report, IBM T. J. Watson Research Center, 1985.