# Phoneme-to-Text Transcription System with an Infinite Vocabulary

**Shinsuke Mori  Daisuke Takuma  Gakuto Kurata**
IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma Yamato-shi, 242-8502, Japan
mori@fw.ipsj.or.jp

## Abstract

The noisy channel model approach is successfully applied to various natural language processing tasks. Currently the main research focus of this approach is adaptation methods, how to capture characteristics of words and expressions in a target domain given example sentences in that domain. As a solution we describe a method enlarging the vocabulary of a language model to an almost infinite size and capturing their context information. Especially the new method is suitable for languages in which words are not delimited by whitespace. We applied our method to a phoneme-to-text transcription task in Japanese and reduced about 10% of the errors in the results of an existing method.

## 1 Introduction

The noisy channel model approach is being successfully applied to various natural language processing (NLP) tasks, such as speech recognition (Jelinek, 1985), spelling correction (Kernighan et al., 1990), machine translation (Brown et al., 1990), etc. In this approach an NLP system is composed of two modules: one is a task-dependent part (an acoustic model for speech recognition) which describes a relationship between an input signal sequence and a word, the other is a language model (LM) which measures the likelihood of a sequence of words as a sentence in the language. Since the LM is a common part, its improvement augments the accuracies of all NLP systems based on a noisy channel model.

Recently the main research focus of LM is shifting to the adaptation method, how to capture the characteristics of words and expressions in a target domain. The standard adaptation method is to prepare a corpus in the application domain, count the frequencies of words and word sequences, and manually annotate new words with their input signal sequences to be added to the vocabulary. It is now easy to gather machine-readable sentences in various domains because of the ease of publication and access via the Web (Kilgarriff and Grefenstette, 2003). In addition, traditional machine-readable forms of medical reports or business reports are also available. When we need to develop an NLP system in various domains, there is a huge but unannotated corpus.

For languages, such as Japanese and Chinese, in which the words are not delimited by whitespace, one encounters a word identification problem before counting the frequencies of words and word sequences. To solve this problem one must have a good word segmenter in the domain of the corpus. The only robust and reliable word segmenter in the domain is, however, a word segmenter based on the statistics of the lexicons in the domain! Thus we are obliged to pay a high cost for the manual annotation of a corpus for each new subject domain.

In this paper, we propose a novel framework for building an NLP system based on a noisy channel model with an almost infinite vocabulary. In our method, first we estimate the probability of a word boundary existing between two characters at each point of a raw corpus in the target domain. Using these probabilities we regard the corpus as a stochastically segmented corpus (SSC). We then estimate word $n$-gram probabilities from the SSC. Then we build an NLP system, the phoneme-to-text transcription system in this paper. To describe the stochastic relationship between a character sequence and its phoneme sequence, we also propose a character-based unknown word model. With this unknown word model and a word $n$-gram model estimated from the SSC, the vocabulary of our LM, a set of known words with their context information, is expanded from words in a



*Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 729–736,
Sydney, July 2006. 

small annotated corpus to an almost infinite size, including all substrings appearing in the large corpus in the target domain. In experiments, we estimated LMs from a relatively small annotated corpus in the general domain and a large raw corpus in the target domain. A phoneme-to-text transcription system based on our LM and unknown word model eliminated about 10% of the errors in the results of an existing method.

## 2 Task Complexity

In this section we explain the phoneme-to-text transcription task which our new framework is applied to.

### 2.1 Phoneme-to-text Transcription

To input a sentence in a language using a device with fewer keys than the alphabet we need some kind of transcription system. In French stenotypy, for example, a special keyboard with 21 keys is used to input French letters with accents (Derouault and Merialdo, 1986). A similar problem arises when we write an e-mail in any language with a mobile phone or a PDA. For languages with a much larger character set, such as Chinese, Japanese, and Korean, a transcription system called an input method is indispensable for writing on a computer (Lunde, 1998).

The task we chose for the evaluation of our method is phoneme-to-text transcription in Japanese, which can also be regarded as a pseudo-speech recognition in which the acoustic model is perfect. In order to input Japanese to a computer, the user types phoneme sequences and the computer offers possible transcription candidates in the descending order of their estimated similarities to the characters the user wants to input.[1] Then the user chooses the proper one.

### 2.2 Ambiguities

A phoneme sequence in Japanese (written in sans-serif font in this paper) is highly ambiguous for a computer. There are many possible word sequences with similar pronunciations. These ambiguities are mainly due to three factors:

- **Homonyms:** There are many words sharing the same phoneme sequences. In the spoken language, they are less ambiguous since they are pronounced with different intonations. Intonational signals are, however, omitted in the input of phoneme-to-text transcription.
- **Lack of word boundaries:** A word of a long sequence of phonemes can be split into several shorter words, such as frequent content words, particles, etc. (ex. a-ri-ga-to-u/thanks vs. a-ri/ant ga/is to-u/ten).
- **Variations in writing:** Some words have more than one acceptable spellings. For example, 振り込み/fu-ri-ko-mi/bank-transfer is often written as 振込/fu-ri-ko-mi omitting two verbal endings, especially in business writing.

Most of these ambiguities are not difficult to resolve for a native speaker who is familiar with the domain. So the transcription system should offer the candidate word sequences for each context and domain.

### 2.3 Available Resources

Generally speaking, three resources are available for a phoneme-to-text transcription based on the noisy channel model:

- **annotated corpus:**
  a small corpus in the general domain annotated with word boundary information and phoneme sequences for each word
- **single character dictionary:**
  a dictionary containing all possible phoneme sequences for each single character
- **raw corpus in the target domain:**
  a collection of text samples in the target domain extracted from the Web or documents in machine-readable form

## 3 Language Model and its Application

A stochastic LM $M$ is a function from a sequence of characters $\boldsymbol{x} \in \mathcal{X}^*$ to the probability. The summation over all possible sequences of characters must be equal to or less than 1. This probability is used as the likelihood in the NLP system.

### 3.1 Word $N$-gram Model

The most famous LM is an $n$-gram model based on words. In this model, a sentence is regarded as a word sequence $\boldsymbol{w}_1^h$ $(= w_1 w_2 \cdots w_h)$ and words are predicted from beginning to end:

$$M_{w,n}(\boldsymbol{w}) = \prod_{i=1}^{h+1} P(w_i | \boldsymbol{w}_{i-n+1}^{i-1}),$$

[1] Generally one of Japanese phonogram sets is used as phoneme. A phonogram is input by a combination of unambiguous ASCII characters.

where $w_i$ $(i \leq 0)$ and $w_{h+1}$ is a special symbol called a BT (boundary token). Since it is impossible to define the complete vocabulary, we prepare a special token UW for unknown words and an unknown word spelling $\boldsymbol{x}_1^{h'}$ is predicted by the following character-based $n$-gram model after UW is predicted by $M_{w,n}$:

$$M_{x,n}(\boldsymbol{x}_1^{h'}) = \prod_{i=1}^{h'+1} P(x_i|\boldsymbol{x}_{i-n+1}^{i-1}), \qquad (1)$$

where $x_i$ $(i \leq 0)$ and $x_{h'+1}$ is a special symbol BT. Thus, when $w_i$ is outside of the vocabulary $\mathcal{W}$,

$$P(w_i|\boldsymbol{w}_{i-n+1}^{i-1}) = M_{x,n}(w_i)P(\texttt{UW}|\boldsymbol{w}_{i-n+1}^{i-1}).$$

### 3.2 Automatic Word Segmentation

Nagata (1994) proposed a stochastic word segmenter based on a word $n$-gram model to solve the word segmentation problem. According to this method, the word segmenter divides a sentence $\boldsymbol{x}$ into a word sequence with the highest probability

$$\hat{\boldsymbol{w}} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}=\boldsymbol{x}} M_{w,n}(\boldsymbol{w}).$$

Nagata (1994) reported an accuracy of about 97% on a test corpus in the same domain using a learning corpus of 10,945 sentences in Japanese.

### 3.3 Phoneme-to-text Transcription

A phoneme-to-text transcription system based on an LM $T$ (Mori et al., 1999) receives a phoneme sequence $\boldsymbol{y}$ and returns a list of candidate sentences $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots)$ in descending order of the probability $P(\boldsymbol{x}|\boldsymbol{y})$:

$$T(\boldsymbol{y}) = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots),$$
$$\text{where } i \leq j \Leftrightarrow P(\boldsymbol{x}_i|\boldsymbol{y}) \geq P(\boldsymbol{x}_j|\boldsymbol{y}).$$

Similar to speech recognition, the probability is decomposed into two independent parts: a pronunciation model (PM) and an LM.

$$\begin{aligned} & P(\boldsymbol{x}_i|\boldsymbol{y}) \geq P(\boldsymbol{x}_j|\boldsymbol{y}) \\ \Leftrightarrow \quad & \frac{P(\boldsymbol{y}|\boldsymbol{x}_i)P(\boldsymbol{x}_i)}{P(\boldsymbol{y})} \geq \frac{P(\boldsymbol{y}|\boldsymbol{x}_j)P(\boldsymbol{x}_j)}{P(\boldsymbol{y})} \\ \Leftrightarrow \quad & P(\boldsymbol{y}|\boldsymbol{x}_i)P(\boldsymbol{x}_i) \geq P(\boldsymbol{y}|\boldsymbol{x}_j)P(\boldsymbol{x}_j) \qquad (2) \\ & (\because P(\boldsymbol{y}) \text{ is independent of } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j.) \end{aligned}$$

In this formula $P(\boldsymbol{x})$ is an LM representing the likelihood of a sentence $\boldsymbol{x}$. For the LM, we can use a word $n$-gram model we explained above.

The other part in the above formula $P(\boldsymbol{y}|\boldsymbol{x})$ is a PM representing the probability that a given sentence $\boldsymbol{x}$ is pronounced as $\boldsymbol{y}$. Since it is impossible to collect the phoneme sequences $\boldsymbol{y}$ for all possible sentences $\boldsymbol{x}$, the model is decomposed into a word-based model $M_y$ in which the words are pronounced independently

$$M_{y,w}(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i), \qquad (3)$$

where $\boldsymbol{y}_i$ is a phoneme sequence corresponding to the word $w_i$ and the condition $\boldsymbol{y} = \boldsymbol{y}_1^h$ is met.

The probabilities $P(\boldsymbol{y}_i|w_i)$ are estimated from a corpus in which each word is annotated with a phoneme sequence as follows:

$$P(\boldsymbol{y}_i|w_i) = \frac{f(\boldsymbol{y}_i,\ w_i)}{f(w_i)}, \qquad (4)$$

where $f(e)$ stands for the frequency of an event $e$ in the corpus. For unknown words no transcription model has been proposed and the phoneme-to-text transcription system (Mori et al., 1999) simply returns the phoneme sequence itself.[2] This is done by replacing the unknown word model based on the Japanese character set $M_{x,n}(\boldsymbol{x})$ by a model based on the phonemic alphabet $M_{y,n}(\boldsymbol{y})$.

Thus the candidate evaluation metric of a phoneme-to-text transcription (Mori et al., 1999) composed of the word $n$-gram model and the word-based pronunciation model is as follows:

$$P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i)P(w_i)$$

$$P(\boldsymbol{y}_i|w_i)P(w_i) \qquad (5)$$
$$= \begin{cases} P(w_i|\boldsymbol{w}_{i-n+1}^{i-1})P(\boldsymbol{y}_i|w_i) & \text{if } w_i \in \mathcal{W}, \\ P(\texttt{UW}|\boldsymbol{w}_{i-n+1}^{i-1})M_{y,n}(\boldsymbol{y}_i) & \text{if } w_i \notin \mathcal{W}. \end{cases}$$

## 4 LM Estimation from a Stochastically Segmented Corpus (SSC)

To cope with segmentation errors, the concept of stochastic segmentation is proposed (Mori and Takuma, 2004). In this section, we briefly explain a method of calculating word $n$-gram probabilities on a stochastically segmented corpus in the target domain. For a detailed explanation and proofs of the mathematical soundness, please refer to the paper (Mori and Takuma, 2004).

[2] One of the Japanese syllabaries *Katakana* is used to spell out imported words by imitating their Japanese-constrained pronunciation and the phoneme sequence itself is the correct transcription result for them. Mori et. al. (1999) reported that approximately 33.0% of the unknown words in a test corpus were imported words.

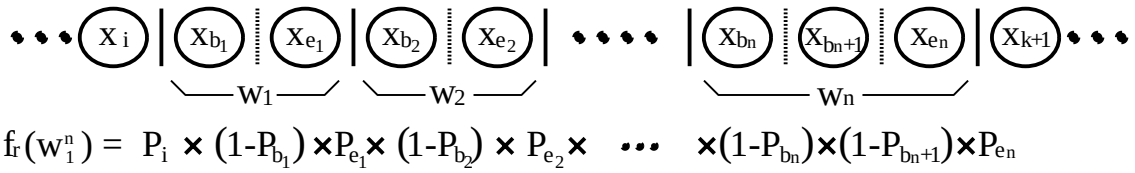


Figure 1: Word $n$-gram frequency in a stochastically segmented corpus (SSC).

### 4.1 Stochastically Segmented Corpus (SSC)

A stochastically segmented corpus (SSC) is defined as a combination of a raw corpus $C_r$ (hereafter referred to as the character sequence $\boldsymbol{x}_1^{n_r}$) and word boundary probabilities $P_i$ that a word boundary exists between two characters $x_i$ and $x_{i+1}$. Since there are word boundaries before the first character and after the last character of the corpus, $P_0 = P_{n_r} = 1$.

In (Mori and Takuma, 2004), the word boundary probabilities are defined as follows. First the word boundary estimation accuracy $\alpha$ of an automatic word segmenter is calculated on a test corpus with word boundary information. Then the raw corpus is segmented by the word segmenter. Finally $P_i$ is set to be $\alpha$ for each $i$ where the word segmenter put a word boundary and $P_i$ is set to be $1 - \alpha$ for each $i$ where it did not put a word boundary. We adopted the same method in the experiments.

### 4.2 Word $n$-gram Frequency

Word $n$-gram frequencies on an SSC is calculated as follows:

**Word 0-gram frequency:** This is defined as an expected number of words in the SSC:

$$f(\cdot) = 1 + \sum_{i=1}^{n_r - 1} P_i.$$

**Word $n$-gram frequency ($n \geq 1$):** Let us think of a situation (see Figure 1) in which a word sequence $\boldsymbol{w}_1^n$ occurs in the SSC as a subsequence beginning at the $(i+1)$-th character and ending at the $k$-th character and each word $w_m$ in the word sequence is equal to the character sequence beginning at the $b_m$-th character and ending at the $e_m$-th character ($\boldsymbol{x}_{b_m}^{e_m} = w_m,\ 1 \leq \forall m \leq n$; $e_m + 1 = b_{m+1},\ 1 \leq \forall m \leq n-1$; $b_1 = i + 1$; $e_n = k$). The word $n$-gram frequency of a word sequence $f_r(\boldsymbol{w}_1^n)$ in the SSC is defined by the summation of the stochastic frequency at each occurrence of the character sequence of the word sequence $\boldsymbol{w}_1^n$ over all of the occurrences in the SSC:

$$f_r(\boldsymbol{w}_1^n) = \sum_{(i,\boldsymbol{e}_1^n) \in O_n} P_i \left[ \prod_{m=1}^{n} \left\{ \prod_{j=b_m}^{e_m - 1} (1 - P_j) \right\} P_{e_m} \right],$$

where $\boldsymbol{e}_1^n = (e_1, e_2, \cdots, e_n)$ and $O_n = \{(i, e_1^n) | \boldsymbol{x}_{b_m}^{e_m} = w_m, 1 \leq m \leq n\}$.

### 4.3 Word $n$-gram probability

Similar to the word $n$-gram probability estimation from a decisively segmented corpus, word $n$-gram probabilities in an SSC are estimated by the maximum likelihood estimation method as relative values of word $n$-gram frequencies:

$$P_r(w) = \frac{f_r(w)}{f_r(\cdot)},$$

$$P_r(w_n | \boldsymbol{w}_1^{n-1}) = \frac{P_r(\boldsymbol{w}_1^n)}{P_r(\boldsymbol{w}_1^{n-1})} \quad (n \geq 2).$$

## 5 Phoneme-to-Text Transcription with an Infinite Vocabulary

The vocabulary of an LM estimated from an SSC consists of all subsequences occurring in it. Adding a module describing a stochastic relationship between these subsequences and input signal sequences, we can build a phoneme-to-text transcription system equipped with an almost infinite vocabulary.

### 5.1 Word Candidate Enumeration

Given a phoneme sequence as an input, the dictionary of a phoneme-to-text transcription system described in Subsection 3.3 returns pairs of a word and a probability per Equation (4). Similarly, the dictionary of a phoneme-to-text system with an infinite vocabulary must be able to take a phoneme sequence $\boldsymbol{y}$ and return all possible pairs of a character sequence $w$ and the probability $P(\boldsymbol{y}|w)$ as word candidates. This is done as follows:

1. First we prepare a single character dictionary containing all characters $x$ in the language annotated with their all possible phoneme sequences $\boldsymbol{\mathcal{Y}}_x = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k\}$. For

example, the Japanese single character dictionary contains a character $x$ = "日" annotated with its all possible phoneme sequences $\mathcal{Y}_{日}$ = {bi, hi, jitsu, ka, ni, nichi, nit}.

2. Then we build a phoneme-to-text transcription system for single characters equipped with the vocabulary consisting of the union set of phoneme sequences for all characters. Given a phoneme sequence $\boldsymbol{y}$, this module returns all possible character sequences $w$ with its generation probability $P(\boldsymbol{y}|w)$. For example, given a subsequence of the input phoneme sequence $\boldsymbol{y}$ = nittere, this module returns $\mathcal{W}$ = {日テレ, 日手レ, 日照レ, ニッテレ, ニッ手レ, ニッ照レ, $\cdots$} as a word candidate set along with their generation probabilities.
3. There are various methods to calculate the probability $P(\boldsymbol{y}|w)$. The only condition is that given $w = x_1x_2\cdots x_m$, $P(\boldsymbol{y}|w)$ must be a stochastic language model (cf. Section 3) on the alphabet $\mathcal{Y}$. In the experiments, we assumed the uniform distribution of phoneme sequences for each character as follows:[3]

$$P(\boldsymbol{y}|w) = P(\boldsymbol{y}|x_1x_2\cdots x_m) = \prod_{i=1}^{m} \frac{1}{|\mathcal{Y}_{x_i}|}. \quad (6)$$

The module we described above receives a phoneme sequence and enumerates its decompositions to subsequences contained in the single character dictionary. This module is implemented using a dynamic programming method. In the experiments we limited the maximum length of the input to 16 phonemes.

### 5.2 Modeling Contexts of Word Candidates

Word $n$-gram probability estimated from an SSC may not be as accurate as an LM estimated from a corpus segmented appropriately by hand. Thus we use the following interpolation technique:

$$P(w_i|H_i) = \lambda_s P_s(w_i|H_i) + \lambda_r P_r(w_i|H_i),$$

where $H_i$ is history before $w_i$, $P_s$ is the probability estimated from a segmented corpus $C_s$, and $P_r$ is the probability estimated by our method from a raw corpus $C_r$. The $\lambda_s$ and $\lambda_r$ are interpolation coefficients which are estimated by the deleted interpolation method (Jelinek et al., 1991).

[3] More precisely, it may happen that the same phoneme sequence is generated from a character sequence in multiple ways. In this case the generation probability is calculated as the summation over all possible generations.

In the experiments, the word bi-gram model in our phoneme-to-text transcription system is combined with word bi-gram probabilities estimated from an SSC. Thus the phoneme-to-text transcription system of our new framework refers to the following LM to measure the likelihood of word sequences:

$$P(w_i) \quad (7)$$
$$= \begin{cases} \lambda_s P_s(w_i|w_{i-1}) + \lambda_r P_r(w_i|w_{i-1}) \\ \quad \text{if } w_i \in \mathcal{W}, \\ \lambda_s P_s(\texttt{UW}|w_{i-1}) M_{x,n}(w_i) + \lambda_r P_r(w_i|w_{i-1}) \\ \quad \text{if } w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r, \\ \lambda_s P_s(\texttt{UW}|w_{i-1}) M_{x,n}(w_i), \quad \because P_r(w_i) = 0 \\ \quad \text{if } w_i \notin \mathcal{W} \wedge w_i \notin \mathcal{S}_r, \end{cases}$$

where $\mathcal{S}_r$ is the set of all subsequences appearing in the SSC.

Our LM based on Equation (7) and an existing LM (cf. Equation (5)) behave differently when they predict an out-of-vocabulary word appearing in the SSC, that is $w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r$. In this case our LM has reliable context information on the OOV word to help the system choose the proper word. Our system also clearly functions better than the LM interpolated with a word $n$-gram model estimated from the automatic segmentation result of the corpus when the result is a wrong segmentation. For example, when the automatic segmentation result of the sequence "日テレ" (the abbreviation of Japan TV broadcasting corporation) has a word boundary between "日" and "テ," the uni-gram probability $P$(日テレ) is equal to 0 and an OOV word "日テレ" is never enumerated as a candidate.[4] To the contrary, using our method $P$(日テレ) $> 0$ when the sequence "日テレ" appears in the SSC at least once. Thus the sequence is enumerated as a candidate word. In addition, when the sequence appears frequently in the SSC, $P$(日テレ) $\gg 0$ and the word may appear at a high position in the candidate list even if the automatic segmenter always wrongly segments the sequence into "日" and "テレ."

### 5.3 Default Character for Phoneme

In very rare cases, it happens that the input phoneme sequence cannot be decomposed into phoneme sequences in the vocabulary and those

[4] Two word fragments "日" and "テレ" may be enumerated as word candidates. The notion of word may be necessary for the user's facility. However, we do not discuss the necessity of the notion of word in the phoneme-to-text transcription system.

corresponding to subsequences of the SSC and, as a result, the transcription system does not output any candidate sentence. To avoid this situation, we prepare a default character for every phoneme and the transcription system also enumerates the default character for each phoneme. In Japanese from the viewpoint of transcription accuracy, it is better to set the default characters to *katakana*, which are used mainly for transliteration of imported words. Since a *katakana* is pronounced uniquely ($|\boldsymbol{\mathcal{Y}}_{x_i}| = 1$),

$$P(\boldsymbol{y}|w) = P(\boldsymbol{y}|x_1 x_2 \cdots x_m) = 1. \quad (8)$$

From Equations (4), (6), and (8), the PM of our transcription system is as follows:

$$P(\boldsymbol{y}_i|w_i) \quad (9)$$
$$= \begin{cases} \dfrac{f(\boldsymbol{y}_i,\ w_i)}{f(w_i)}, & \text{if } w_i \in \mathcal{W}, \\ \displaystyle\prod_{j=1}^{m} \frac{1}{|\mathcal{Y}_{x_j}|}, & \text{if } w_i \notin \mathcal{W} \wedge w_i \in \mathcal{S}_r, \\ 1, & \text{if } w_i \notin \mathcal{W} \wedge w_i \notin \mathcal{S}_r, \end{cases}$$

where $w_i = x_1 x_2 \cdots x_m$.

### 5.4 Phoneme-to-Text Transcription with an Infinite Vocabulary

Finally, the transcription system with an infinite vocabulary enumerates candidate sentence $\boldsymbol{x} = w_1 w_2 \cdots w_h$ in the descending order of the following evaluation function value composed of an LM $P(w_i)$ defined by Equation (7) and a PM $P(\boldsymbol{y}_i|w_i)$ defined by Equation (9):

$$P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}) = \prod_{i=1}^{h} P(\boldsymbol{y}_i|w_i)P(w_i)$$

Note that there are only three cases since the case decompositions in Equation (7) and Equation (9) are identical.

## 6 Evaluation

As an evaluation of our phoneme-to-text transcription system, we measured transcription accuracies of several systems on test corpora in two domains: one is a general domain in which we have a small annotated corpus with word boundary information and phoneme sequence for each word, and the other is a target domain in which only a large raw corpus is available. As the transcription result, we took the word sequence of the highest probability. In this section we show the results and evaluate our new framework.

Table 1: Annotated corpus in general domain

| | #sentences | #words | #chars |
|---|---|---|---|
| learning | 20,808 | 406,021 | 598,264 |
| test | 2,311 | 45,180 | 66,874 |

Table 2: Raw corpus in the target domain

| | #sentences | #words | #chars |
|---|---|---|---|
| learning | 797,345 | — | 17,645,920 |
| test | 1,000 | — | 20,935 |

### 6.1 Conditions on the Experiments

The segmented corpus used in our experiments is composed of articles extracted from newspapers and example sentences in a dictionary of daily conversation. Each sentence in the corpus is segmented into words and each word is annotated with a phoneme sequence. The corpus was divided into ten parts. The parameters of the model were estimated from nine of them (learning) and the model was tested on the remaining one (test). Table 1 shows the corpus size. Another corpus we used in the experiments is composed of daily business reports. This corpus is not annotated with word boundary information nor phoneme sequence for each word. For evaluation, we selected 1,000 sentences randomly and annotated them with the phoneme sequences to be used as a test set. The rest was used for LM estimation (see Table 2).

### 6.2 Evaluation Criterion

The criterion we used for transcription systems is precision and recall based on the number of characters in the longest common subsequence (LCS) (Aho, 1990). Let $N_{COR}$ be the number of characters in the correct sentence, $N_{SYS}$ be that in the output of a system, and $N_{LCS}$ be that of the LCS of the correct sentence and the output of the system, so the recall is defined as $N_{LCS}/N_{COR}$ and the precision as $N_{LCS}/N_{SYS}$.

### 6.3 Models for Comparison

In order to clarify the difference in the usages of the target domain corpus, we built four transcription systems and compared their accuracies. Below we explain the models in detail.

**Model $\mathcal{B}$:** Baseline
A word bi-gram model built from the segmented general domain corpus.

Table 3: Phoneme-to-text transcription accuracy.

| | word bi-gram from the annotated corpus | raw corpus usage | unknown word model | General domain | | Target domain | |
|---|---|---|---|---|---|---|---|
| | | | | Precision | Recall | Precision | Recall |
| $\mathcal{B}$ | Yes | No | No | 89.80% | 92.30% | 68.62% | 78.40% |
| $\mathcal{D}$ | Yes | Auto. Seg. | No | 92.67% | 93.42% | 80.59% | 86.19% |
| $\mathcal{D}'$ | Yes | Auto. Seg. | Yes | 92.52% | 93.17% | 90.35% | 93.48% |
| $\mathcal{S}$ | Yes | Stoch. Seg. | Yes | 92.78% | 93.40% | 91.10% | 94.09% |

The vocabulary contains 10,728 words appearing in more than one corpora of the nine learning corpora. The automatic word segmenter used to build the other three models is based on the method explained in Section 3 with this LM.

**Model $\mathcal{D}$:** Decisive segmentation
A word bi-gram model estimated from the automatic segmentation result of the target corpus interpolated with model $\mathcal{B}$.

**Model $\mathcal{D}'$:** Decisive segmentation
Model $\mathcal{D}$ extended with our PM for unknown words

**Model $\mathcal{S}$:** Stochastic segmentation
A word bi-gram model estimated from the SSC in the target domain interpolated with model $\mathcal{B}$ and equipped with our PM for unknown words

### 6.4 Evaluation

Table 3 shows the transcription accuracy of the models. A comparison of the accuracies in the target domain of the Model $\mathcal{B}$ and Model $\mathcal{D}$ confirms the well known fact that even an automatic segmentation result containing errors helps an LM improve its performance. The accuracy of Model $\mathcal{D}$ in the general domain is also higher than that of Model $\mathcal{B}$. From this result we can say that over-adaptation has not occurred.

Model $\mathcal{D}'$, equipped with our PM for unknown words, is a natural extension of Model $\mathcal{D}$, a model based on an existing method. The accuracy of Model $\mathcal{D}'$ is higher than that of Model $\mathcal{D}$ in the target domain, but worse in the general domain. This is because the vocabulary of Model $\mathcal{D}'$ is enlarged with the words and the word fragments contained in the automatic segmentation result. Though no study has been reported on the method of Model $\mathcal{D}'$, below we take Model $\mathcal{D}'$ as an existing method for a more severe evaluation.

Comparing the accuracies of Model $\mathcal{D}'$ and Model $\mathcal{S}$ in both domain, it can be said that using our method we can build a more accurate model than the existing methods. The main reason is that our phoneme model PM is able to enumerate transcription candidates for out-of-vocabulary words and word $n$-gram probabilities estimated from the SSC helps the model choose the appropriate ones.

Table 4: Relationship between the raw corpus size and the accuracies.

| Raw corpus size | Precision | Recall |
|---|---|---|
| $1.765 \times 10^5$ chars (1/100) | 89.18% | 92.32% |
| $1.765 \times 10^6$ chars (1/10) | 90.33% | 93.40% |
| $1.765 \times 10^7$ chars (1/1) | 91.10% | 94.09% |

A detailed study of Table 3 tells us that the reduction rate of character error rate ($100\% - \text{recall}$) of Model $\mathcal{S}$ in the target domain (9.36%) is much larger than that in the general domain (3.37%). The reason for this is that the automatic word segmenter tends to make mistakes around characteristic words and expressions in the target domain and our method is much less influenced by those segmentation errors than the existing method is.

In order to clarify the relationship between the size of the SSC and the transcription accuracy, we calculated the accuracies while changing the size of the SSC (1/1, 1/10, 1/100). The result, shown in Table 4, shows that we can still achieve a further improvement just by gathering more example sentences in the target domain.

The main difference between the models is the LM part. Thus the accuracy increase is yielded by the LM improvements. This fact indicates that we can expect a similar improvement in other generative NLP systems using the noisy channel model by expanding the LM vocabulary with context information to an infinite size.

## 7 Related Work

The well-known methods for the unknown word problem are classified into two groups: one is to use an unknown word model and the other is to extract word candidates from a corpus before the application. Below we describe the relationship

between these methods and the proposed method.

In the method using an unknown word model, first the generation probability of an unknown word is modeled by a character $n$-gram, and then an NLP system, such as a morphological analyzer, searches for the best solution considering the possibility that all subsequences might be unknown words (Nagata, 1994; Bazzi and Glass, 2000). In the same way, we can build a phoneme-to-text transcription system which can enumerate unknown word candidates, but the LM is not able to refer to lexical context information to choose the appropriate word, since the unknown words are modeled to be generated from a single state. We solved this problem by allowing the LM to refer to information from an SSC.

When a machine-readable corpus in the target domain is available, we can extract word candidates from the corpus with a certain criterion and use them in application. An advantage of this method is that all of the occurrences of each candidate in the corpus are considered. Nagata (1996) proposed a method calculating word candidates with their uni-gram frequencies using a forward-backward algorithm. and reported that the accuracy of a morphological analyzer can be improved by adding the extracted words to its vocabulary. Comparing our method with this research, it can be said that our method executes the word candidate enumeration and their context calculation dynamically at the time of the solution search for an NLP task, phoneme-to-text transcription here. One of the advantages of our framework is that the system considers all substrings in the corpus as word candidates (that is the recall of the word extraction is 100%) and a higher accuracy is expected using a consistent criterion, namely the generation probability, for the word candidate enumeration process and solution search process.

The framework we propose in this paper, enlarging the vocabulary to an almost infinite size, is general and applicable to many other NLP systems based on the noisy channel model, such as speech recognition, statistical machine translation, etc. Our framework is potentially capable of improving the accuracies in these tasks as well.

## 8 Conclusion

In this paper we proposed a generative NLP system with an almost infinite vocabulary for languages without obvious word boundary information in written texts. In the experiments we compared four phoneme-to-text transcription systems in Japanese. The transcription system equipped with an infinite vocabulary showed a higher accuracy than the baseline model and the model based on the existing method. These results show the efficacy of our method and tell us that our approach is promising for the phoneme-to-text transcription task or other NLP systems based on the noisy channel model.

## References


Alfred V. Aho. 1990. Algorithms for finding patterns in strings. In *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, pages 273–278. Elseveir Science Publishers.

Issam Bazzi and James R. Glass. 2000. Modeling out-of-vocabulary words for robust speech recognition. In *Proc. of the ICSLP2000*.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Anne-Marie Derouault and Bernard Merialdo. 1986. Natural language modeling for phoneme-to-text transcription. *IEEE PAMI*, 8(6):742–749.

Frederick Jelinek, Robert L. Mercer, and Salim Roukos. 1991. Principles of lexical language modeling for speech recognition. In *Advances in Speech Signal Processing*, chapter 21, pages 651–699. Dekker.

Frederick Jelinek. 1985. Self-organized language modeling for speech recognition. Technical report, IBM T. J. Watson Research Center.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale. 1990. A spelling correction program based on a noisy channel model. In *Proc. of the COLING90*, pages 205–210.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–347.

Ken Lunde. 1998. *CJKV Information Processing*. O'Reilly & Associates.

Shinsuke Mori and Daisuke Takuma. 2004. Word n-gram probability estimation from a Japanese raw corpus. In *Proc. of the ICSLP2004*.

Shinsuke Mori, Tsuchiya Masatoshi, Osamu Yamaji, and Makoto Nagao. 1999. Kana-kanji conversion by a stochastic model. *Transactions of IPSJ*, 40(7):2946–2953. (in Japanese).

Masaaki Nagata. 1994. A stochastic Japanese morphological analyzer using a forward-DP backward-A$^*$ n-best search algorithm. In *Proc. of the COLING94*, pages 201–207.

Masaaki Nagata. 1996. Automatic extraction of new words from Japanese texts using generalized forward-backward search. In *EMNLP*.