Automatic Extraction of New Words from Japanese Texts using Generalized Forward-Backward Search

Masaaki NAGATA
NTT Information and Communication Systems Laboratories
1-2356 Take, Yokosuka-Shi, Kanagawa, 238-03 Japan
nagata@nttnly.isl.ntt.jp

Abstract

We present a novel new word extraction method from Japanese texts based on expected word frequencies. First, we compute expected word frequencies from Japanese texts using a robust stochastic N-best word segmenter. We then extract new words by filtering out erroneous word hypotheses whose expected word frequencies are lower than the predefined threshold. The method is derived from an approximation of the generalized version of the Forward-Backward algorithm. When the Japanese word segmenter is trained on a 4.7 million word segmented corpus and tested on 1000 sentences whose out-of-vocabulary rate is 2.1%, the accuracy of the new word extraction method is 43.7% recall and 52.3% precision.

Introduction

Segmentation of sentences into words is trivial in English because words are delimited by spaces. It is a simple task to count word frequencies in a given text. It is also a simple task to list all new words (unknown words), namely, the words in a given text that are not found in the system dictionary. However, several languages such as Japanese, Chinese and Thai do not put spaces between words and so in these languages word segmentation, word frequency counting, and new word extraction remain unsolved problems in computational linguistics.

Most Japanese NLP applications require word segmentation as a first stage because there are phonological units and semantic units whose pronunciation and/or meaning is not trivially derivable from the pronunciation and/or meaning of the individual characters. It is well known that the accuracy of word segmentation greatly depends on the coverage of the dictionary, in other words, the Out-Of-Vocabulary (OOV) rate of the target texts.

Our goal is to provide a method to automatically extract new words from Japanese texts. This method should adapt the dictionary of the word segmenter to new domains and applications. It should also maintain the dictionary by collecting new words in the target domain. The application of the word segmenter is described elsewhere (Nagata, 1996).

The approach we take is as follows: First, we design a statistical language model that can assign a reasonable word probability to an arbitrary substring in the input sentence, whether or not it is truly a word. Second, we devised a method to obtain the expected word N-gram count in the target texts, using an N-best word segmentation algorithm (Nagata, 1994). Finally, we extract new words by filtering out spurious word hypotheses whose expected word frequencies are lower than the threshold.

Japanese Morphological Analysis

Before we start, we briefly explain the difficulties of Japanese morphological analysis, especially when the input sentence includes unknown words. Suppose the input sentence is "ペンシルバニア大学はENIACの50周年を祝う。", which means "University of Pennsylvania celebrates the 50th anniversary of ENIAC", where the words ペンシルバニア (transliteration of 'Pennsylvania') and ENIAC (the name of the world's first computer) are not registered in the system dictionary. Figure 1 shows three possible analyses of the input sentence, where each box represents a word hypothesis whose meaning and part of speech are shown above and under the box. The tag <UNK> represents an unknown word.

One of the hardest problems in handling unrestricted Japanese text is the identification of unknown words. In Figure 1, the string ENIAC is successfully tokenized as an unknown word. However, there is ambiguity in the segmentation of the string ペンシルバニア大学.

In the first analysis, the system considers ペンシルバニア ('Pennsylvania') as an unknown word,

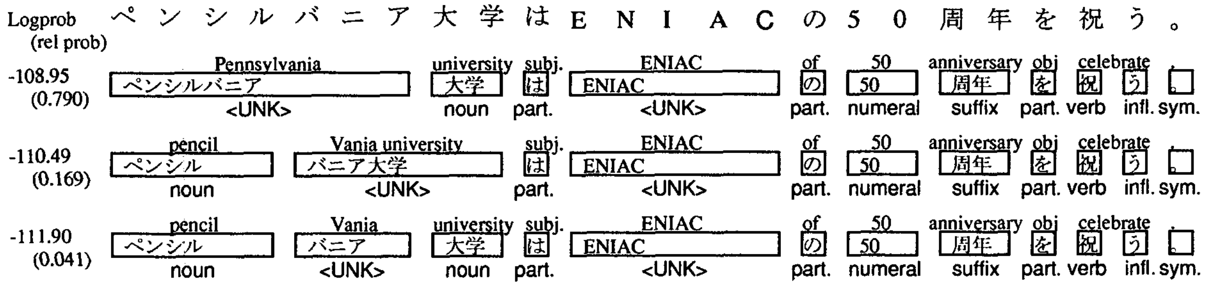

Figure 1: Japanese Morphological Analysis Example

because 大学 ('university') is registered in the dictionary. This is correct. In the second analysis, the system guesses バニア大学 ('Vania university') as an unknown word, because ペンシル (transliteration of 'pencil') is registered in the dictionary and some university names are registered in the dictionary, such as スタンフォード大学 ('Stanford University') and ケンブリッジ大学 ('Cambridge University'). In the third analysis, the system considers バニア ('Vania') as an unknown word, because both ペンシル and 大学 are registered in the dictionary.

It is often the case that we have overlapping word hypotheses if the input sentence contains unknown words, such as ペンシルバニア, バニア大学, and バニア in Figure 1. We need a criteria to select the most likely word hypothesis from among the overlapping candidates. In fact, it is fairly difficult to get plausible analyses like the ones shown in Figure 1, because failure to identify an unknown word affects the segmentation of the neighboring words. Obviously, a robust word segmenter is the essential first step.

In the following sections, we first describe a statistical language model to cope with unknown words. We then describe the word segmentation algorithm and the new word extraction method, with their derivation as an approximation of a generalization of the Forward-Backward algorithm (Baum, 1972). Finally, we show experiment results and prove its effectiveness.

Statistical Language Model

Segmentation Model (Tagging Model)

Let the input Japanese character sequence be $C = c_1c_2 \ldots c_m$, and segment it into word sequence $W = w_1w_2 \ldots w_n$ whose part of speech sequence is $T = t_1t_2 \ldots t_n$. The word segmentation task can be defined as finding the set of word segmentation and parts of speech assignment $(\hat{W}, \hat{T})$ that maximize the joint probability of word sequence and tag sequence given character sequence $P(W,T|C)$. Since the maximization is carried out with fixed character sequence C, the word segmenter only has to maximize the joint probability of word sequence and tag sequence $P(W,T)$.

$$\begin{aligned}(\hat{W},\hat{T}) &= \arg\max_{W,T} P(W,T|C) \\ &= \arg\max_{W,T} P(W,T) \qquad (1)\end{aligned}$$

We call $P(W,T)$ the segmentation model, although it is usually called tagging model in English tagger research. In this paper, we compare three segmentation models: part of speech trigram, word unigram, and word bigram.

In the part-of-speech trigram model (POS trigram model), the joint probability $P(W,T)$ is approximated by the product of parts of speech trigram probabilities $P(t_i|t_{i-2},t_{i-1})$ and word output probabilities for given part of speech $P(w_i|t_i)$

$$P(W,T) = \prod_{i=1}^{n} P(t_i|t_{i-2},t_{i-1})P(w_i|t_i) \qquad (2)$$

In the word unigram and word bigram models, the joint probability $P(W,T)$ is approximated by the product of word unigram probabilities $P(w_i,t_i)$ and word bigram probabilities $P(w_i,t_i|w_{i-1},t_{i-1})$, respectively.

$$P(W,T) = \prod_{i=1}^{n} P(w_i,t_i) \qquad (3)$$

$$P(W,T) = \prod_{i=1}^{n} P(w_i,t_i|w_{i-1},t_{i-1}) \qquad (4)$$

Basically, parameters of these segmentation models are estimated by computing the relative frequencies of the corresponding events in the segmented training corpus. However, in order to handle unknown words, we have introduced a slight modification in computing the relative frequencies, as is described in the next section.

Word Model

We think of an unknown word as a word having a special part of speech <UNK>. We define a statistical word model to assign a word probability to each word hypothesis. It is formally defined as the joint probability of the character sequence $c_1 \ldots c_k$ if w_i is the unknown word. We decompose it into the product of word length probability and word spelling probability,

$$\begin{aligned} P(w_i|\texttt{<UNK>}) &= P(c_1 \ldots c_k|\texttt{<UNK>}) \\ &= P(k)P(c_1 \ldots c_k|k) \qquad (5) \end{aligned}$$

where k is the length of the character sequence. We call $P(k)$ the word length model, and $P(c_1 \ldots c_k|k)$ the word spelling model.

We assume that word length probability $P(k)$ obeys a Poisson distribution whose parameter is the average word length λ in the training corpus,

$$P(k) = \frac{(\lambda - 1)^{k-1}}{(k-1)!} e^{-(\lambda-1)} \qquad (6)$$

This means that we regard word length as the interval between hidden word boundary markers, which are randomly placed with an average interval equal to the average word length. Although this word length model is very simple, it plays a key role in making the word segmentation algorithm robust.

We approximate the spelling probability given word length $P(c_1 \ldots c_k|k)$ by the word-based character bigram model, regardless of word length. Since there are more than 3,000 characters in Japanese, the amount of training data would be too small if we divided them by word length.

$$P(c_1 \ldots c_k) = P(c_1|\#) \prod_{i=2}^{k} P(c_i|c_{i-2}) P(\#|c_k) (7)$$

Here, special symbol "#" indicates the word boundary marker.

Note that the word-based character bigram model is different from the sentence-based character bigram model. The former is estimated from the corpus segmented into words. It assigns a large probability to a character sequence that appears in the beginning (prefixes), the middle, and the end (suffixes) of a word. It also assigns a small probability to a character sequence that appears across a word boundary.

By using the word model, we can create modified segmentation models that take unknown words into consideration. The parameters of the modified POS trigram, word unigram, and word bigram are estimated by Equations (8), (9), (10), and (11), in Figure 2.

In Figure 2, $C(\cdot)$ denotes the count of the specified event in the training corpus. In the part of speech trigram model, $P(w_i|t_i)$ for an unknown word w_i is obtained, by definition, from the word model $P(w_i|\texttt{<UNK>})$. In the word unigram model, the unigram count $C(w_i)$ for unknown word w_i is given as the product of the total unigram count of unknown words $C(\texttt{<UNK>})$ and the word model probability $P(w_i|\texttt{<UNK>})$. The higher order N-gram counts involving unknown words are also obtained in the same manner.

In order to compute the parameters in Figure 2, we need the counts involving unknown words, such as $C(t_{i-2}, t_{i-1}, \texttt{<UNK>})$, $C(\texttt{<UNK>})$, and $C((w_{i-1}, t_{i-1}), \texttt{<UNK>})$. These counts are important because they represent the contexts in which unknown words likely to appear. To estimate these counts, we replace all words appearing only once in the training corpus with unknown word tags <UNK>, before computing relative frequencies. The underlying idea of the replacement is the same as Turing's estimates in back-off smoothing (Katz, 1987). We redistribute the probability mass of low count sequences to "unseen" sequences.

Generalized Forward Backward Reestimation

Generalization of the Forward and Viterbi Algorithm

In English part of speech taggers, the maximization of Equation (1) to get the most likely tag sequence, is accomplished by the Viterbi algorithm (Church, 1988), and the maximum likelihood estimates of the parameters of Equation (2) are obtained from untagged corpus by the Forward-Backward algorithm (Cutting et al., 1992). However, it is impossible to apply the Viterbi algorithm and the Forward-Backward algorithm for word segmentation of those languages that have no delimiter between words, such as Japanese and Chinese, because word segmentation hypotheses overlap one another.

Figure 3 shows an example of overlapping word hypotheses and possible word segmentations for the string 全国都道府県 ('all prefectures in the nation'). We assume 全国 ('all nation'), 全 ('all'), 国都 ('national capital'), 都道府県 ('prefectures'), 都道 ('metropolitan road'), 都 ('metropolis'), 道府県 ('prefectures'), 道 ('road'), 府県 ('prefectures'), 府 ('prefecture'), and 県 ('prefecture') are registered in the dictionary. There are 15 possible word segmentations in this example. In Japanese, a lot of words consist of one character. Moreover, sequence of characters may constitute a different word.

$$P(t_i|t_{i-2},t_{i-1}) = \begin{cases} \frac{C(t_{i-2},t_{i-1},\text{<UNK>})}{C(t_{i-2},t_{i-1})} & \text{if } t_i = \text{<UNK>} \\ \frac{C(t_{i-2},t_{i-1},t_i)}{C(t_{i-1},t_{i-1})} & \text{otherwise} \end{cases} \tag{8}$$

$$P(w_i|t_i) = \begin{cases} P(w_i|\text{<UNK>}) & \text{if } t_i = \text{<UNK>} \\ \frac{C(w_i,t_i)}{C(t_i)} & \text{otherwise} \end{cases} \tag{9}$$

$$P(w_i,t_i) = \begin{cases} \frac{C(\text{<UNK>})}{\sum_i C(w_i,t_i)} \times P(w_i|\text{<UNK>}) & \text{if } t_i = \text{<UNK>} \\ \frac{C(w_i,t_i)}{\sum_i C(w_i,t_i)} & \text{otherwise} \end{cases} \tag{10}$$

$$P(w_i,t_i|w_{i-1},t_{i-1}) = \begin{cases} \frac{C((w_{i-1},t_{i-1}),\text{<UNK>})}{C(w_{i-1},t_{i-1})} \times P(w_i|\text{<UNK>}) & \text{if } t_i = \text{<UNK>} \\ \frac{C((w_{i-1},t_{i-1}),(w_i,t_i))}{C(w_{i-1},t_{i-1})} & \text{otherwise} \end{cases} \tag{11}$$

Figure 2: Modified Segmentation Models with Consideration to Unknown Words.

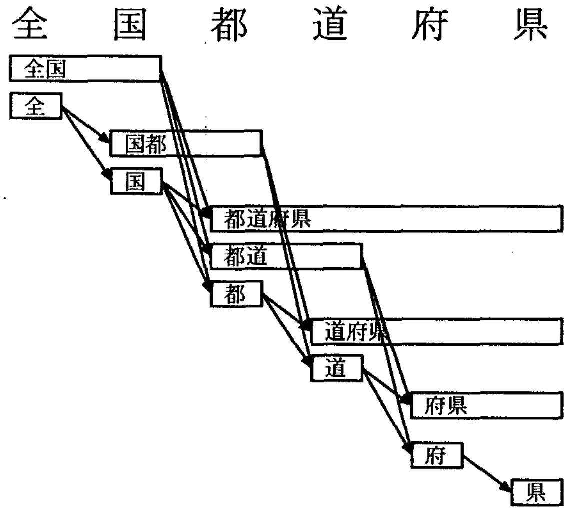

Figure 3: Overlapping Word Hypotheses and Possible Word Segmentations

For Japanese word segmentation, we define a generalized Forward algorithm and a generalized Viterbi algorithm as follows. Let the input Japanese character sequence of length n be $C = c_1 c_2 \dots c_n$, and c_p^q denote the substring $c_{p+1} \dots c_q$. We define a function D that maps a character sequence c_p^q to a list of word hypotheses $\{w_i\}$. Function D is the generalization of the dictionary. Here, w_i denotes a combination of orthography (formally denoted by w_i) and part of speech t_i, for simplicity. We use word bigram as the segmentation model in the following example. Other segmentation models, such as part of speech trigram and word unigram, can be used in the same manner.

In the generalized forward algorithm, the forward probability $\alpha_p^q(w_i)$ is the joint probability of the character sequence c_0^q and the event that the final word in the segmentation of c_0^q is w_i that spans the substring c_p^q. Forward probabilities can be recursively computed as follows.

$$\alpha_q^r(w_{i+1}) = \sum_{0 \le p < q} \sum_{w_i \in D(c_p^q)} \alpha_p^q(w_i) P(w_{i+1}|w_i)$$
$$w_{i+1} \in D(c_q^r), 0 \le q < n, q < r \le n \tag{12}$$

The generalized forward algorithm starts from the beginning of the input sentence, and proceeds character by character. At each point q in the sentence, it sums over the product of the forward probability of the word segmentation hypotheses ending at the point $\alpha_p^q(w_i)$ and the transition probability to the word hypotheses starting at that point $P(w_{i+1}|w_i)$.

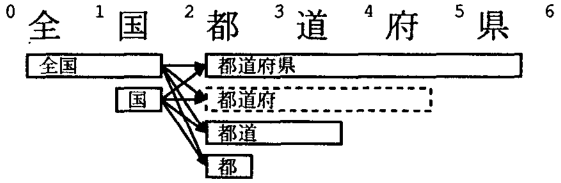

Figure 4: One Step in the Generalized Forward Algorithm.

Figure 4 shows a snapshot of the generalized forward algorithm. The input is 全国都道府県, and the current point q is 2. The word hypotheses ending at point 2 ($w_i \in D(c_p^2)$) are 全国 (c_0^2) and 国 (c_1^2). Those starting at point 2 ($w_{i+1} \in D(c_2^q)$) are 都道府県 (c_2^6), 都道 (c_2^4), and 都 (c_2^3). The string 都道府 (c_2^5) is not registered in the dictionary. All combinations of these words are examined.

The generalized Viterbi algorithm can be obtained by replacing summation with maximization in Equation (12). Here, $\phi_p^q(w_i)$ is the probability of the most likely word segmentation sequence for the character sequence c_0^q whose final word w_i spans the substring c_p^q.

$$\phi_q^r(w_{i+1}) = \max_{0 \leq p < q} \max_{w_i \in D(c_p^q)} \phi_p^q(w_i) P(w_{i+1}|w_i)$$
$$w_{i+1} \in D(c_q^r), 0 \leq q < n, q < r \leq n \quad (13)$$

Note that the original Forward algorithm and the Viterbi algorithm is the special case in Equation (12) and (13) where p and q are fixed as $p = q - 1$ and $r = q + 1$.

In order to handle unknown words, the dictionary function D returns a word hypothesis tagged as unknown word if the substring c_p^q is not registered in the dictionary, such as 都道府 (c_2^5) in Figure 4. The word model assigns a reasonable probability to the unknown word. Therefore, in the generalized forward algorithm and the generalized Viterbi algorithm, we hypothesize all substrings in the input sentence as words, and examine all possible combinations of these word hypotheses.

Since we can define the generalized Backward algorithm in the same manner, we can define the generalized Forward-Backward algorithm to estimate the word N-gram counts in Japanese texts, and to reestimate the word N-gram probabilities in the segmentation model. However, we give a more intuitive account of the method to introduce an approximation of the generalized Forward-Backward algorithm.

Expected Word N-gram Count

By using the above mentioned word segmentation algorithm, we can get all word segmentation hypotheses of the input sentence. Once we get them, we can estimate word N-gram count in an unsegmented Japanese corpus.

Let O_j^i be the jth word segmentation hypothesis for the ith sentence in the corpus. $P(O_j^i)$ can be computed by using the segmentation model. The Bayes *a posteriori* estimate of the word unigram count $C^i(w_i)$ and the word bigram count $C^i(w_{i-1}, w_i)$ in the ith sentence can be computed as,

$$C^i(w_\alpha) = \sum_j (\frac{P(O_j^i)}{\sum_k P(O_k^i)} \times n_j^i(w_\alpha)) \quad (14)$$

$$C^i(w_\alpha, w_\beta) = \sum_j (\frac{P(O_j^i)}{\sum_k P(O_k^i)} \times n_j^i(w_\alpha, w_\beta)) \quad (15)$$

Here, $n_j^i(w_\alpha)$ and $n_j^i(w_\alpha, w_\beta)$ denote the number of times the unigram w_α and the bigram w_α, w_β appeared in the jth candidate of the ith sentence [1].

The estimate of the total unigram count $C(w_\alpha)$ and the total bigram count $C(w_\alpha, w_\beta)$ can be obtained by summing the counts over all sentences in the corpus.

$$C(w_\alpha) = \sum_i C^i(w_\alpha) \quad (16)$$

$$C(w_\alpha, w_\beta) = \sum_i C^i(w_\alpha, w_\beta) \quad (17)$$

The estimate of the unigram probability and the bigram probability can be obtained as the relative frequency of the associated events.

$$f(w_\alpha) = \frac{C(w_\alpha)}{\sum_\alpha C(w_\alpha)} \quad (18)$$

$$f(w_\beta|w_\alpha) = \frac{C(w_\alpha, w_\beta)}{C(w_\alpha)} \quad (19)$$

If necessary, we can reestimate the word N-gram probabilities by replacing $P(w_\alpha)$ and $P(w_\beta|w_\alpha)$ with $f(w_\alpha)$ and $f(w_\beta|w_\alpha)$.

Extraction of New Words in Texts

Expected word unigram counts (expected word frequencies) in the corpus (Equation (16)) can be used as a measure of likelihood that a particular substring in the input texts is actually a word. Let θ denote the minimum expected word frequency that we use to classify a given word hypothesis w_α as a word.

$$C(w_\alpha) > \theta \quad (20)$$

Those words that are not found in the dictionary and whose expected frequencies in the corpus are larger than the threshold θ are extracted as the new words in the input texts.

In theory, expected word N-gram counts can be obtained by the generalized Forward-Backward algorithm. In order to save computation time, however, we approximated the weighted sum of the word N-gram counts over all the word segmentation hypotheses in a sentence (Equation (14)), by that of the N-best word segmentation hypotheses[2].

[1] Note that the (Generalized) Forward-Backward algorithm is devised to compute these expected word N-gram count without listing all word segmentation hypotheses.

[2] If we only use the best word segmentation, it is called the Viterbi reestimation. Our method might be called N-best reestimation. It is designed to be more accurate than the Viterbi reestimation and more efficient than the generalized Forward-Backward algorithm.

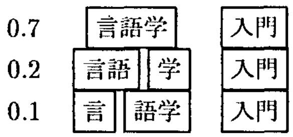

Figure 5: An example of computing the expected word frequencies

N-best word segmentation hypotheses can be obtained by using the Forward-DP Backward-A^* algorithm (Nagata, 1994). It consists of a forward dynamic programming search to record the probabilities of all partial word segmentation hypotheses, and a backward A^* algorithm to extract the N-best hypotheses. It is a generalization of the tree-trellis search (Soong and Huang, 1991), in the sense that its forward Viterbi search is replaced with the generalized Viterbi search described in this paper.

In reestimating the word N-gram probabilities, we introduce two modifications to the normal reestimation procedure. The first modification is that, instead of using the relative frequency in an unsegmented corpus (Equation (18) and (19)), we combine the N-gram count in the segmented corpus with the estimated N-gram count in the unsegmented corpus to increase estimate reliability. This is because a fairly large amount of segmented Japanese corpus were available in our experiments.

$$f(w_\alpha) = \frac{C_{seg}(w_\alpha) + C_{unseg}(w_\alpha)}{\sum_\alpha C_{seg}(w_\alpha) + \sum_\alpha C_{unseg}(w_\alpha)} \quad (21)$$

$$f(w_\beta|w_\alpha) = \frac{C_{seg}(w_\alpha, w_\beta) + C_{unseg}(w_\alpha, w_\beta)}{C_{seg}(w_\alpha) + C_{unseg}(w_\alpha)} (22)$$

where $C_{seg}(\cdot)$ denotes the count in the segmented corpus, and $C_{unseg}(\cdot)$ denotes the estimated count in the unsegmented corpus.

The second modification is that we prune the expected N-gram counts in the unsegmented corpus if they are lower than a predefined threshold, before computing Equation (21) and (22). This is because $C_{unseg}(\cdot)$ is unreliable, especially when $C_{unseg}(\cdot)$ is low.

Examples of Estimating Expected Word Frequencies

Finally, we show a simple example of estimating the word N-gram counts in an unsegmented sentence. Assume that the ith input sentence is the character sequence 言語学入門, which means "introduction to linguistics", and its best three word segmentation hypotheses are as shown in Figure 5. The leftmost numbers in Figure 5 are the relative probabilities of the word segmentation hypotheses, corresponding to $\frac{P(O_j^i)}{\sum_k P(O_k^i)}$ in Equation (14). The expected word unigram count of each word hypothesis in the sentence is,

$$\begin{aligned} C^i(\text{入門}) &= 0.7 + 0.2 + 0.1 = 1.0 \\ C^i(\text{言語学}) &= 0.7 \\ C^i(\text{言語}) = C^i(\text{学}) &= 0.2 \\ C^i(\text{言}) = C^i(\text{語学}) &= 0.1 \end{aligned}$$

The expected total number of the words in the sentence $\sum_\alpha C^i(w_\alpha)$ is 2.3. If all word hypotheses are not registered in the dictionary and the threshold θ is 0.15, we regard 入門 ('introduction'), 言語学 ('linguistics'), 言語 ('language'), and 学 ('study') as the new words. 言 ('say') and 語学 ('study of languages') are discarded.

Let us give another example that shows the effect of summing the expected word unigram counts over all the sentences in the corpus. Suppose the sentence "ペンシルバニア大学はENIAC の50周年を祝う。", which means "University of Pennsylvania celebrates the 50th anniversary of ENIAC.", is in the corpus, and the first three word segmentation hypotheses are as shown in Figure 1. The expected word unigram counts for ペンシルバニア ('Pennsylvania'), バニア大学 ('Vania University'), and バニア ('Vania') are 0.790, 0.169, and 0.041, respectively. Suppose also the sentence "ホワイトハウスはペンシルバニア通りにある。", which means "White House lies at Pennsylvania Avenue.", is in the corpus, and the expected word unigram counts for ペンシルバニア ('Pennsylvania'), バニア通り ('Vania Avenue'), and バニア ('Vania') are 0.825, 0.127, and 0.048, respectively. The expected word unigram counts in the corpus are,

$$\begin{aligned} C(\text{ペンシルバニア}) &= 0.790 + 0.825 = 1.615 \\ C(\text{バニア大学}) &= 0.169 \\ C(\text{バニア通り}) &= 0.127 \\ C(\text{バニア}) &= 0.041 + 0.048 = 0.089 \end{aligned}$$

Therefore, ペンシルバニア is definitely more likely to be a new word. The more often the unknown word appears in the corpus, the more it is likely to be extracted, even if there is word segmentation ambiguity in each sentence.

Experiments

Language Data

We used the EDR Japanese Corpus Version 1.0 (EDR, 1995) to train and test the word segmentation program. It is a corpus of approximately 5 million words (200,000 sentences). It was collected to build a Japanese Electronic Dictionary, and contains a variety of Japanese sentences taken from newspapers, magazines, dictionaries, encyclopedias, textbooks, etc. It has a variety of annotations on morphology, syntax, and semantics. We used word segmentation, pronunciation, and part of speech in the morphology information field of the annotation.

In this experiment, we randomly selected 90% of the sentences in the EDR Corpus for training the word segmentation program. We made two test sets from the rest of the corpus, one for a small size experiment (100 sentences) and the other for a medium size experiment (1000 sentences). Table 1 shows the number of sentences, words, and characters for training and test sets. Note that the test sets were not part of the training set. That is, open data were tested in the experiment.

Table 1: The amount of training and test data

	training	test-1	test-2
Sentences	192802	100	1000
Words	4746461	2463	25177
Characters	7521293	3912	39875

The training texts contained 133281 word types. We discarded word types that appeared only once in the training texts. This resulted in 65152 word types being registered in the dictionary of the word segmenter. We trained three segmentation models, namely, part of speech trigram, word unigram, and word trigram, after we replaced those words appeared only once in the training texts with the unknown word tag <UNK>, as described in the section of word model. After this replacement, there were 758172 distinct word bigrams. Again, we discarded word bigrams that appeared only once in the training texts for saving main memory, and used the remaining 294668 word bigrams. The word bigram probabilities were smoothed using deleted interpolation (Jelinek, 1985).

The training texts contained 3534 character types. We discarded characters that appeared only once in the training texts; 3167 character types remained. We then replaced the discarded characters with the unknown character tag to train the word spelling model. There were 91198 distinct character bigrams in the words in the training texts [3].

We made two spelling models. The first was trained using all words in the training texts, while the second was trained using those words whose frequency is less than or equal to 2. In principle, the spelling model of unknown words must be trained using the low frequency words. However, it might suffer from the sparse data problem because the total number of word tokens for training is decreased from 4746461 to 103919. We also made two length models. The average word lengths of all words and that of low frequency words were 1.58 and 4.49, respectively. Note that the average word length is the only parameter of the word length model.

Evaluation Measures

Word Segmentation accuracy is expressed in terms of recall and precision. First, we count the number of words in corpus segmentation (Std), the number of words in system segmentation (Sys), and the number of matching word segmentations (M). *Recall* is defined as M/Std, and *precision* is defined as M/Sys.

Figure 6 shows an example of computing precision and recall for the sentence “ロックフェラー研究所はアメリカの大富豪ロックフェラーが設立した学術研究所です。”, which means “Rockefeller Laboratory is an academic laboratory founded by an American millionaire, Rockefeller”. Because of the difference in the segmentation of ロックフェラー研究所, the number of words in corpus segmentation (Std=15) differs from that of system segmentation (Sys=14). Note that the system correctly tokenized 学術研究所, although it is not registered in the dictionary.

New word extraction accuracy is described in terms of recall, precision, and F-measure. First, we count the number of unknown words in the corpus segmentation (Std), the number of unknown words in the system segmentation (Sys), and the number of matching words (M). Here, unknown words are those that are not registered in the system dictionary. *Recall* is defined as M/Std, and *precision* is defined as M/Sys. Since recall and precision greatly depend on the frequency threshold, we used the F-measure to indicate the overall performance. F-measure is used in Information Retrieval, and is calculated by

$$F = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R} \tag{23}$$

where P is precision, R is recall, and β is the relative importance given to recall over precision.

[3] There are more than 3000 (some say more than 10000) charters in Japanese, and their frequency distribution is skewed. In order to save memory, we used a type of character bigram model that considers unknown characters, like the word bigram model used in the segmentation model.

```
JC00092627
ロックフェラー研究所はアメリカの大富豪ロックフェラーが設立した学術研究所です。
corpus segmentation               system segmentation
ロックフェラー研究所/ロック       | ロックフェラー/ロックフェ          Rockefeller
                                  > 研究所/ケンキュウジョ/名詞         laboratory
は/ハ/助詞                          は/ハ/助詞                         particle (topic)
アメリカ/アメリカ/名詞              アメリカ/アメリカ/名詞             America
の/ノ/助詞                          の/ノ/助詞                         of
大/ダイ/接頭語                      大/ダイ/接頭語                     big
富豪/フゴウ/名詞                    富豪/フゴウ/名詞                   rich man
ロックフェラー/ロックフェラ         ロックフェラー/ロックフェ          Rockefeller
が/ガ/助詞                          が/ガ/助詞                         particle (subject)
設立/セツリツ/動詞                  設立/セツリツ/動詞                 found
し/シ/語尾                          し/シ/語尾                         inflectional suffix
た/タ/助動詞                        た/タ/助動詞                       auxiliary verb (past)
学術研究所/ガクジュツケンキ       | 学術研究所/NIL/<UNK>               academic laboratory
です/デス/助動詞                    です/デス/助動詞                   be
。/。/記号                          。/。/記号                         .  (period)

sys=15, std=14, matched=13
precision=87.7 (13/15), recall=92.9 (13/14)
```

Figure 6: Comparison between the corpus segmentation (left) and the system segmentation (right). All words are listed in UNIX **sdiff** style.

Word Segmentation Accuracy

In order to decide the best configuration of the underlying Japanese word segmenter, we compared three segmentation models: part of speech trigram, word unigram, and word bigram. We also compared three word models: all words, low frequency words, and the combination of the two. The third word model consisted of the spelling model trained using all words and the length model trained using low frequency words.

Table 2 shows, for the small test set (100 sentences), the segmentation accuracy of the various combinations of the segmentation models and the word models.

It is obvious that word bigram outperformed the part of speech trigram as well as word unigram. As for the word model, it seems the combination of the spelling model for all words and the length model for low frequency words is the best, but the difference is small. In the following experiment, we decided to use word bigram as the segmentation model, and the combination of the spelling model of all words and the length model of low frequency words as the word model.

New Word Extraction Accuracy

We tested the new word extraction method using the medium size test set (1000 sentences). It contains 538 unknown word types. 8 word types appeared twice in the test set. The other 530 word types appeared only once. The out-of-vocabulary rate of the test set is 2.2%. To count the expected word frequencies, we used the top-10 word segmentation hypotheses. We limited the maximum character length of the a unknown word to 8 in order to save computation time.

We tested three variations of the new word extraction method. The first one was "No Reestimation"; it uses the word segmenter's outputs as they are when extracting new words. The second and the third ones carry out reestimation before extraction, where the pruning thresholds of the expected N-gram counts in the reestimation are 0.95 and 0.50, respectively. Reestimations were carried out three times.

Table 3 shows the new word extraction accuracies for a variety of expected word frequency thresholds θ, with and without reestimation. In Table 3, we set $\beta = 1.0$ to compute F-measure.

As Table 3 shows, the higher the threshold is, the higher the precision and the lower the recall become. When we put equal importance on recall and precision, the best value for the expected word frequency threshold is around 0.10 where the recall is 43.7% and the precision is 52.3%.

Figure 7 shows excerpts of correctly extracted new words (**matched**), incorrectly extracted word hypotheses (**sys-matched**), and new words that were not extracted (**std-matched**), when the frequency threshold was 0.5 and reestimation was not carried out. We find that the overall quality of the extracted word hypotheses is satisfactory, al-

Table 2: Language Models and Segmentation Accuracies (100 test sentences)

word model	POS trigram		word unigram		word bigram	
	recall	prec.	recall	prec.	recall	prec.
all words	91.6	88.8	88.7	87.3	94.6	89.4
low frequency words	91.5	89.5	88.4	88.0	94.3	90.1
all words + l.f.w. length	91.5	89.3	88.8	87.6	94.7	89.9

Table 3: New Word Extraction Accuracy (1000 test sentences)

freq.	No Reestimation			freq>0.95, 3 iter.			freq>0.50, 3 iter.		
	recall	prec.	F	recall	prec.	F	recall	prec.	F
>0.00	56.1	34.2	42.5	50.6	37.9	43.4	39.6	56.7	46.6
>0.10	43.7	52.3	47.6	43.1	52.1	47.2	37.9	63.6	47.5
>0.50	36.4	65.6	46.8	36.1	65.8	46.6	36.6	65.2	46.9
>0.90	25.3	76.8	35.8	25.3	77.3	38.1	36.6	65.2	46.9
>0.95	23.2	78.1	35.8	23.4	78.3	36.1	36.6	65.2	46.9
>0.99	17.3	81.6	28.5	23.4	78.3	36.1	36.6	65.2	46.9

though the values of recall and precision are not so high. We discuss the reason for this in the next section.

Discussion

The problem of Japanese word segmentation is that people often can not agree on a single word segmentation. Therefore, the reported performance could be greatly underestimated. Most of the new words extracted by the system are acceptable as a word (at least for us), and may not necessarily be a wrong word entry. On the other hand, most of the new words not extracted by the system can be divided into shorter words that are registered in the dictionary.

For example, in the first sentence of Figure 8, データ・コミュニケーション ('data communication') is regarded as one word in corpus segmentation and counted as an unknown word in the test sentence. However, the system segmented it into データ ('data') and コミュニケーション ('communication'), both of which are found in the dictionary. In the second sentence of Figure 8, the system extracted ハノーヴァ公 ('Duke of Hanover') as a new word, while this word is divided into ハノーヴァ ('Hanover') and 公 ('Duke') in corpus segmentation. Most of extraction errors are of this category.

There are three types of obvious extraction errors. The first type is the truncation of long words. Some transliterated Western-origin words exceed the predefined maximum length for unknown word. The third sentence of Figure 8 is an example of this type. In Japanese, 'illustration' is transliterated into 9 characters イラストレーション, which exceeds the maximum unknown word length of 8 characters in our system. Since イラスト (the transliteration of 'illust', which also means illustration in Japanese) is registered in the dictionary, レーション (the transliteration of 'ration') is incorrectly extracted as a new word.

The second type is the fragmentation of numerals. Since we did not use any tokenizers, numerals tend to be divided arbitrarily. In the second sentence in Figure 8, the system divided "1676" into "16" and "76". In fact, it may output "1" and "676", "16" "7" and "6", or whatever.

The third type is the concatenation of noun(s) and particle. In other words, the system sometimes erroneously recognizes a noun phrase as a word. For example, the Japanese counterparts of "A of B", "A and B", and "A, B" are recognized as a word. This may be because the probability of one long unknown word can be higher than the product of the probabilities of two short unknown (or infrequent) words and one known word. The fourth sentence of Figure 8 is an example of this type of error. The system considered 可制御かつ可観測 ('controllable and observable') as a word, while it is divided into 可 ('able'), 制御 ('control'), かつ ('and'), 可 ('able'), and 観測 ('observe') in the corpus.

As for reestimation, Table 3 shows no significant improvements in the new word extraction accuracy. The only effect of reestimation, in our experiment, is to increase the expected word frequencies of the unknown word hypotheses whose expected word frequencies are greater than the pruning threshold of reestimation.

This result does not necessarily mean that reestimation is useless. This is because most un-

```
matched=196
３万１４８７ しんかい２０００ キリシタン ジャップ トム・ニース トリップ フリードリッヒ レトリック
暗語 印紙 開明 楽天主義 凶悪犯 作業帽 賜杯 羨まし 日伯援護協会 百松 傍聴者 与信 ...

sys-matched=103
９０万７０００余 ＳＴＫ製 エクソン社 エストリッジ ジャカール機械 ファクチュア化 フローティング マニュ
乾式構法 汗牛充 順々 清掃局 占い師 村山大臣 東大宇宙航空 灘中、灘高 二浪 年功給 破壊読出し 陸上幕僚長 ...

std-matched=342
４０４ ＢＢＮアドバンスト・コンピューター社 Ｘ線天文学 あっと言う間 ギャラップ調査 レジャー産業
ロックフェラー研究所 引きも切らず 教員住宅 勤労意欲 国際情報化社会 仕立て物 実験班 真珠採取 吹き付け
清掃車 先客 先鞭をつけ 短音 倒れ込 ...

threshold=0.5
std=538, sys=299, matched=196
recall=36.4 (196/538), precision=65.6 (196/299)
```

Figure 7: Excerpts of correctly extracted new words (**matched**), incorrectly extracted word hypotheses (**sys-matched**), and not extracted new words (**std-matched**).

known words appeared only once in the test sentences. An ideal example to confirm that reestimation works well would have an unknown word appearing more than twice in the test sentences, and it is trivial to extract the word in one appearance, while it is difficult in the others, because of, for example, successive unknown words. If the test set were larger, or the out-of-vocabulary rate were higher, we believe that the effectiveness of reestimation would be more clearly shown.

Related Work

Recent years have seen several works on corpus-based word segmentation and dictionary construction for both Japanese and Chinese. For Chinese, (Sproat et al., 1994) used the word unigram model in their word segmenter based on weighted finite-state transducer. Word frequencies were estimated by the Viterbi reestimation (a reestimation procedure using the best analysis) from an unsegmented corpus of 20 million words. Initial estimates of the word frequencies were derived from the frequencies in the corpus of the strings of *hanzi* making up each word in the lexicon *whether or not* each string is actually an instance of the word in question.

(Chang et al., 1995) proposed an automatic dictionary construction method for Chinese from a large unsegmented corpus (311591 sentences) with the help of a small segmented seed corpus (1000 sentences). They combined Viterbi reestimation using the word unigram model with a post filter called the "Two-Class Classifier", which is a linear discrimination function to decide whether the string is actually a word or not based on features derived from the character N-gram in a large unsegmented corpus. The system's performance is compared with a word list derived from two on-line Chinese dictionaries (21141 words). The reported recall and precision values were 56.88% and 77.37% for two character words, and 6.12% and 85.97% for three character words, respectively.

For Japanese, (Nagao and Mori, 1994) proposed a method of computing an arbitrary length character N-gram, and showed that the character N-gram statistics obtained from a large corpus includes information useful for word extraction. However, they did not report any evaluation of their word extraction method.

(Teller and Batchelder, 1994) proposed a very naive probabilistic word segmentation method for Japanese, based on character type information and *hiragana* bigram frequencies. They claimed 98% word segmentation accuracy, while we claim 94.7%. However, their evaluation method is very optimistic, and completely different from ours. They count an error only when the system segmentation violates morpheme boundaries. In other words, they count an error only when the system segmentation is not acceptable to human judgement, while we count an error whenever the system segmentation does not exactly match the corpus segmentation, even if it is inconsistent.

We used the word bigram model for word segmentation, and expected word frequency for unknown word extraction. We compared the results with a segmented Japanese corpus, and reported 43.7% recall and 52.3% precision for 1000 sentences whose out-of-vocabulary rate is 2.1%. It is impossible to compare our results with (Chang et al., 1995), because the experiment conditions are completely different in terms of language (Chinese vs. Japanese), the size of seed segmented corpus, the size of target unsegmented corpus and its out-of-vocabulary rate, the size of initial word list, and the type of reference data (on-line dictionary vs. segmented corpus).

Our idea of filtering erroneous word hypothesis by expected word frequency is simple and straightforward. The major contribution of this paper is that we present a more accurate method for estimating word frequencies in an unsegmented corpus, even if it includes unknown words. This is achieved by introducing an explicit statistical model of unknown words, and by using an N-best word segmentation algorithm (Nagata, 1994) as an approximation of the generalized Forward-Backward algorithm.

In English taggers, (Weischedel et al., 1993) proposed a statistical model to estimate word output probability $p(w_i|t_i)$ for an unknown word from spelling information such as inflectional endings, derivational endings, hyphenation, and capitalization. Our word model can be thought of a generalization of their statistical model. One potential benefit of our statistical model and segmentation algorithm is that they are completely independent of the target language and its writing system. We intend to test our word segmentation method on other languages, such as Chinese and Thai.

Conclusion

We present a new word extraction method for Japanese based on expected word frequency, which is computed by using a statistical language model and an N-best word segmentation algorithm. Although we have encouraging initial results, there are a number of questions to be answered, for example, the minimum seed segmented corpus size required, the minimum initial word list required, the effect of reestimation for a large unsegmented corpus with various out-of-vocabulary rates. Besides these questions, we are also thinking of assigning the part of speech to the extracted new words in order to construct a Japanese dictionary automatically.

References

[Baum, 1972] Leonard E. Baum. 1972. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, 3, pages 1-8.

[Chang et al., 1995] Jing-Shin Chang, Yi-Chung Lin, and Keh-Yih Su. 1995. Automatic Construction of a Chinese Electronic Dictionary, In *Proceedings of VLC-95*, pages 107-120.

[Church, 1988] Kenneth W. Church. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, In *Proceedings of ANLP-88*, pages 136-143.

[Cutting et al., 1992] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A Practical Part-of-Speech Tagger, In *Proceedings of ANLP-92*, pages 133-140.

[EDR, 1995] Japan Electronic Dictionary Research Institute. 1995. *EDR Electronic Dictionary Version 1 Technical Guide*, EDR TR2-003. Also available as *The Structure of the EDR Electronic Dictionary*, `http:///www.iijnet.or.jp/edr/`.

[Jelinek, 1985] Frederick Jelinek. 1985. Self-organized Language Modeling for Speech Recognition. IBM Report.

[Katz, 1987] Slava M. Katz. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer, IEEE Trans. ASSP-35, No.3, pp.400-401.

[Nagao and Mori, 1994] Makoto Nagao and Shinsuke Mori. 1994. A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese, in *Proceedings of COLING-94*, pages 611-615.

[Nagata, 1994] Masaaki Nagata. 1994. A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A^* N-Best Search Algorithm. In *Proceedings of COLING-94*, pages 201-207.

[Nagata, 1996] Masaaki Nagata. 1996. Context-Based Spelling Correction for Japanese OCR. To appear in *Proceedings of COLING-96*.

[Soong and Huang, 1991] Frank K. Soong and Eng-Fong Huang. 1991. A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition. In *Proceedings of ICASSP-91*, pages705-708.

[Sproat et al., 1994] Richard Sproat, Chinlin Shih, William Gale, and Nancy Chang. 1994. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese, In *Proceedings of ACL-94*, pages 66-73.

[Teller and Batchelder, 1994] Virginia Teller and Eleanor Olds Batchelder. 1994. A Probabilistic Algorithm for Segmenting Non-Kanji Japanese Strings, In *Proceedings of AAAI-94*, pages 742-747.

[Weischedel et al., 1993] Ralph Weischedel, Marie Meteer, Richard Schwartz, Lance Ramshaw, and Jeff Palmucci. 1993. Coping with Ambiguity and Unknown Words through Probabilistic Models, in *Computational Linguistics*, Vol.19, No.2, pages 359-382.

```
JC00076244
コンピューターは徐々に衛星や光ファイバーなどによるデータ・コミュニケーションを通して接続されている。
Computers are increasingly getting connected through data communication such as satellites and
optical fibers.

11c11,13
データ・コミュニケーション/      | データ/データ/名詞              data
                                > ・/・/記号                      - (hyphen)
                                > コミュニケーション/コミュ        communication

JC00001185
１６７６年ハノーヴァー公の顧問兼図書館長となり，ベルリン科学アカデミーの創立に努力して，１７００年院長となった.
In 1676, he became the consultant of Duke of Hanover and the head of the library, and he worked
hard to found Berlin science academy, then, in 1700, he became the president.

1c1,2
１６７６/１６７６/数字            | １６/１６/数字
                                > ７６/７６/数字
3,4c4
ハノーヴァー/ハノーヴァー/        | ハノーヴァー公/NIL/<UNK>        Duke of Hanover
公/コウ/接尾語                    <
8,9c8
図書館/トショカン/名詞            | 図書館長/トショカンチョウ/        head of library
長/チョウ/接尾語                  <
14c13,14
ベルリン科学アカデミー/ベル        | ベルリン/ベルリン/名詞          Berlin
                                > 科学アカデミー/カガクアカ        science academy

JC00071929
イラストレーションを対象とした公募展を毎年行い、日比野克彦ら新しい才能を発掘した実績をもつ。
He held public exhibition of illustration every year, and found many new talents, such as Mr.
Katsuhiko Hibino.

1c1,2
イラストレーション/イラスト        | イラスト/イラスト/名詞          ilust
                                > レーション/NIL/<UNK>            ration
10c11
毎年/マイトシ/名詞                | 毎年/マイネン/名詞              every year
14c15
日比野克彦/ヒビノカツヒコ/          | 日比野克彦/NIL/<UNK>            Katsuhiko Hibino

JC00165663
線形システムが可制御かつ可観測であれば，カルマン・フィルターは漸近安定である.
If linear system is controllable and observable, Karman filter is asymptotic stable.

3,7c3
可/カ/接頭語                      | 可制御かつ可観測/NIL/<UNK>      controllable and observable
制御/セイギョ/名詞                <
かつ/カツ/接続詞                  <
可/カ/接頭語                      <
観測/カンソク/名詞                <
10c6
れ/レ/助動詞                      | れ/レ/語尾                      inflectional suffix
15c11,12
漸近安定/ゼンキンアンテイ/        | 漸近/ゼンキン/名詞              asymptotic
                                > 安定/アンテイ/名詞              stability
```

Figure 8: Comparison between the corpus segmentation (left) and the system segmentation (right). Only differences are listed in UNIX **sdiff -s** style.