A Stochastic Morphological Analysis for Japanese employing Character n-Gram and k-NN Method

Kenji Nagamatsu and **Hidehiko Tanaka**
University of Tokyo, Faculty of Engineering
Kougaku-bu 13-goukan, University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan, 113

1 Introduction

Because Japanese corpora have been developed recently, it has become possible to perform stochastic morphological analysis for Japanese(Nagata, 1994; Takeuchi and Matsumoto, 1995; Mori and Nagao, 1996; Yamamoto et al., 1997). Although the same Hidden Markov Model-based approach as English can be fundamentally applicable with word/part-of-speech n-gram data, some problems peculiar to Japanese make the approach indirect. Before calculating the most likely part-of-speech(abbreviated to 'pos') sequence, it is required to segment input sentences into morphemes referring to word dictionaries.

On the other hand, there are a few researches on stochastic morphological analysis of Japanese WITHOUT word dictionaries (Papageorgiou, 1994; Yamamoto and Masuyama, 1997; Kageura, 1997). However, they require much computational power because they employ character n-gram data in the HMM-based method.

This paper proposes a new method of stochastic morphological analysis for Japanese texts which utilizes character n-gram data(thus, no need for word dictionaries) and a k-Nearest Neighbor method. Because the k-NN method is fundamental in memory-based reasoning and it does not require so much computational power as the HMM method, this new method is fast and achieves high accuracy in word segmentation and pos tagging.

First, section 2 describes the details of stochastic morphological analysis of Japanese texts and clarifies the problems in these researches. Next, section 3 describes our new k-NN based method and it is evaluated in section 4. Section 5 discusses the effectiveness and the problems of our method. Finally, after mentioning the future works in section 6, section 7 concludes this paper.

2 Related Researches

2.1 A Stochastic Approach with n-Gram Data and Hidden Markov Model

In English many researches on pos tagging have been performed with statistical data, such as n-gram, extracted from corpora(DeRose, 1988; Cutting et al., 1992; Merialdo, 1994; Dermatas and Kokkinakis, 1995). Now the accuracy of stochastic pos tagging has achieved 95% or higher.

The method of the pos tagging with HMM can be considered as the minimization process in the following production of probabilities. Then the most likely combination of pos tags, which has the minimum probability in the equation, is attached to the words in the input sentence.

$$\Pr(W,T) = \prod \Pr(t_i|t_{i-n+1},\ldots,t_{i-1}) \Pr(w_i|t_i) \quad (1)$$

W is a word sequence of an input sentence and T is a pos tag sequence corresponding to W.

Although this process is efficiently performed with forward DP backward A* search algorithm(Nagata, 1994), the computational complexity of the process is proportional to the number of candidates for each tag pattern to the power n. With pos n-gram data this complexity is almost insignificant because the number of possible tag patterns is small and the length of the word(pos) sequence is also small(20∼30 in English).

Another problem arises when no probability data exists for a given word(pos) pattern. This problem originates in the sparseness of corpora and the probability for the unknown pattern must be estimated from other probabilities(Katz, 1987; Brown et al., 1992; Dagan and Pereira, 1994; McMahon and Smith, 1996). The back-off method(Katz, 1987) interpolates the value using the probabilities of more general patterns. It will be showed that this interpolation is automatically integrated into our method.

2.2 Japanese-peculiar Problems in the HMM Approach

Japanese is an agglutinative language and all morphemes in sentences adhere to each other. Thus, pos tagging for Japanese texts requires a preliminary processing which segments sentences into morphemes by referring to word dictionaries.

Because every position between characters can be a boundary of morphemes, the word segmentation of Japanese texts cannot be performed uniquely and produces many variations of segmented sequences. This results in the decline of tagging accuracy and the inefficiency in speed.

After segmenting sentences the same HMM-based method as English is applicable to Japanese(Nagata, 1994; Takeuchi and Matsumoto, 1995; Mori and Nagao, 1996; Yamamoto et al., 1997). These researches have achieved more than 90% in the tagging precision. The precision of (Yamamoto et al., 1997) is reported to be more than 97% but because it has no unknown word model, this research cannot be directly applied to sentences containing unknown words.

This fact clarifies the importance of an unknown word model. In English the existence of unknown words does not interfere with other words. In Japanese, however, because all morphemes adhere to each other, the existence of unknown words may lead to the uncertainty in segmentation and directly to the decline of the maximum likelihood probability in the equation (1). With unknown word models more precise tagging will be achieved because in English there are some cases that prefixes or suffixes of unknown words show their pos tags. It is true in the case of Japanese because of the existence of Kanji and Katakana words[1].

2.3 A Stochastic Approach with Character n-Gram Data

On the other hand, there are a few research on stochastic morphological analysis of Japanese WITHOUT dictionaries(Papageorgiou, 1994; Yamamoto and Masuyama, 1997; Kageura, 1997). Because of employing character n-gram data these do not require to segment sentences in advance.

(Yamamoto and Masuyama, 1997) tackled on these problems by employing extended character n-gram data[2] in the HMM-based method.

This method achieved 95.91% in segmenting precision and 94.13% in tagging precision. To process 1000 sentences, however, it took about 670sec(UltraSPARC 140MHz) because the computational complexity of the HMM-based method is proportional to the number of possible extended characters for one character to the power n.

3 A Stochastic Approach with k-NN Method

3.1 k-NN Method in Fundamentals

Although the Nearest Neighbor method is basically a searching method for the similar examples, it can be also utilized as a classification method(Dasarathy, 1991).

Given both an example database and a similarity metric $sim(e_i, e_j)$ over examples, the k-NN method decides the class of an input data by majority from the classes of the k nearest examples.

This method is simple enough to be implemented easily and has been reported to be effectively superior to other methods(such as decision tree, neural network, etc.) in some benchmarks(Michie et al., 1994). (Zavrel and Daelemans, 1997) also reports that the weighted k-NN method is comparable to or better than the back-off method in PP attachment and POS tagging.

3.2 Extended Character n-Gram Data

Before providing a detailed description of the proposed method, it is required to introduce our n-gram data format. This format is based on character n-gram and extended for the k-NN method(called extended character n-gram data).

A raw n-gram data is formulated as a triplet $\langle C_i, B_j, T_i \rangle$. C_i is a list of characters in an n-gram and i ranges from 1 to n. B_j is a list of booleans which tells whether a position j in the n-gram is a boundary of morphemes or not (0 for false, 1 for true) and the index j ranges from 0 to n, where a position j denotes the position between a character C_j and a character C_{j-1}. T_i denotes the pos tag of the word containing a character C_i.

After extracting all raw data from some corpus, two probabilities for B_j and T_i are calculated with the data which share a same surface string C_i. Extended character n-gram data is defined by this new triplet $\langle C_i, p^B_j, p^t_i \rangle$, where p^B_j is probability

[1] In Japanese, foreign words are spelled with special characters(Katakana) and we call them Katakana words.

[2] 'Extended character' contains the character, the pos tag of the word containing this character and the information whether the word terminates at the right position of this character.

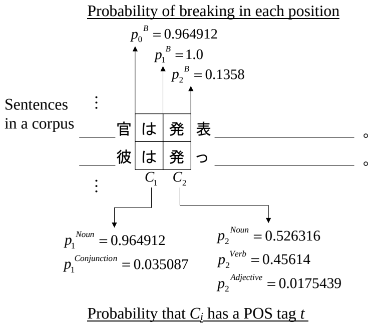

Figure 1: Extended Character n-Gram (in the case of 2-gram)

that a position j is a boundary of morphemes and p_i^t is probability that t is the pos tag of the word which contains a character C_i(see Figure 1).

3.3 Applying k-NN Method to Japanese Morphological Analysis

Without a priori knowledge about Kanjis with similar meaning or collocations, it is acceptable that the more characters are shared between two strings, the more similar are these. Thus, the simplest similarity metric is employed in this paper.

$$sim(e_1, e_2) = \sum_{i=1}^{n} w_i \delta(e_1^i, e_2^i) \qquad (2)$$

, where $\delta(e_1^i, e_2^i)$ is 1 when $e_1^i = e_2^i$; otherwise 0. e_1^i is the i-th character in an extended n-gram e_1.

There are many candidates(window patterns) when searching near examples around some position in an input sentence(see Figure 2). Some patterns, however, are useless because they contain "*" between the position and meaningful characters. By ignoring these useless patterns, you have only to consider those patterns in Figure 3. One of the known problems in the k-NN method is that the number of examples compared with an input data becomes too large. With this pruning the number of the possible window patterns becomes equal to $(n+1)(n+2)/2$ for p_i^B or $n(n+1)/2$ for p_i^t, which is actually equal to the number of reference to dictionaries and it is insignificant.

With these pruned window patterns and the similarity metric(equation (2)), the procedure of the morphological analysis employing the k-NN method is summarized as follows, where $\hat{p}_i^B$ denotes the probability of boundary at a position i in an input sentence and $\hat{p}_i^t$ denotes the probability that a character C_i in an input sentence has a pos tag t.

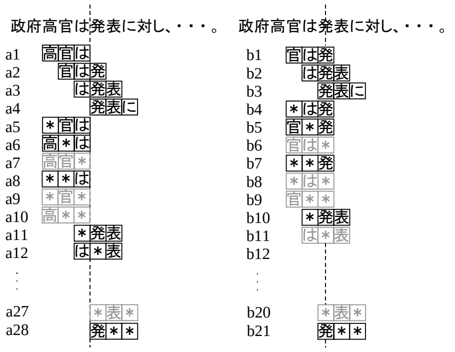

Figure 2: Possible Window Patterns. The left is a figure for calculating p_i^B and the right is one for p_i^t. "*" means 'don't care' and grayed patterns are useless for calculating similarity.

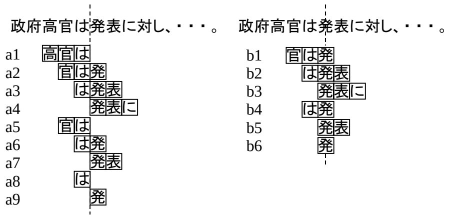

Figure 3: Pruned Window Patterns

1. Search all extended n-gram data from the database which matches with substrings around the position i of the input sentence.

2. Calculate similarity values between every n-gram and an input sentence with the equation (2) and choose the most similar n-gram e_o.

3. **(Estimate of $\hat{p}_i^B$)** Assign the value p_j^B of e_o to $\hat{p}_i^B$, where j corresponds to the position i.

4. **(Estimate of $\hat{p}_i^t$)** Assign the list of values p_j^t $(t = Noun, Verb, ...)$ to $\hat{p}_i^t$.

5. Loop step 1. through step 4. until all p_i^B and p_i^t in the input sentence are determined.

6. **(Segmentation)**If the value $\hat{p}_i^B$ is more than 0.5, consider the position i as a boundary of morphemes and segment the input sentence.

7. **(POS tagging)** To each segmented morpheme, assign the pos tag t with the maximum $\hat{p}_i^t$ in the morpheme.

	All Data Set			
	1-gram	2-gram	3-gram	4-gram
(1)	9,448,229	8,799,431	8,172,812	7,724,156
(2)	5710	345,108	1,572,361	3,319,892
(3)	1654.68	25.498	5.1978	2.3266
(4)	250K	18.5M	111M	294M

Table 1: Summary of Extracted n-Gram Data for All Data Set. (1):Total # of n-gram, (2):# of n-gram patterns, (3):Ave. freq. per pattern, (4):Data size.

4 Experiments

4.1 Extraction of Extended n-Gram Data

In these experiments we utilized the EDR corpus (Electronic Dictionary Research Institute Ltd., 1995), which contains 207,802 Japanese sentences which are already segmented into morphemes and tagged. The number of pos tags employed in this corpus is 15.

To perform both open-data evaluation and close-data evaluation, we selected 1000 sentences from the corpus and call the set "test data set". The data set containing the remaining 206,802 sentences is called "training data set" and the data set containing all sentences(207,802) of the EDR corpus is called "all data set".

From both the all data set and the training data set, extended n-gram data have been extracted for $n = 1, 2, 3, 4$. The statistics of the data are showed in Table 1. The n-gram data extracted from the all data set is employed in the close-data evaluation and the other n-gram data is employed in the open-data evaluation.

Table 1 shows that the data size of 3-gram and 4-gram is very large. It is also known that the size of example databases is one of the problems in the k-NN method. This problem does no harm in the accuracy of pos tagging but may affect the processing time. This is also mentioned later in section 4.4 and discussed in section 5.

4.2 Experiment 1: Word Segmentation

With the extracted n-gram data we evaluated the accuracy of segmenting sentences into morphemes by the proposed method. Thus, this experiment evaluates the accuracy of $\hat{p}_i^B$ obtained in step 3.

The procedure of the experiment is as follows.

1. Store all the extended n-gram data whose n is equal to or less than N into the database.
2. Choose one sentence from the test data set and segment it with our method.

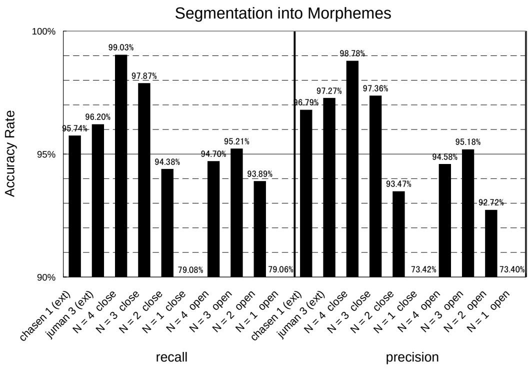

Figure 4: Accuracy in Word Segmentation

3. Compare the output and the answer in the corpus.
4. With the number of the matched morphemes, calculate the precision and the recall.
5. Loop step 3. through step 5. until all sentences in the test data set are processed.
6. Calculate the average precision rate and the average recall rate.

The result of this experiment is showed in Figure 4. For simplicity's sake the weight values in the equation (2) are set to 1.0 in the experiments. Section 5 discusses the problem of optimal weight.

In Figure 4 'chasen'(Matsumoto et al., 1997) and 'juman'(Matsumoto et al., 1994) are traditional morphological analyzers based on connectivity rules and costs. Because 'chasen' and 'juman' employ their own morpheme system, it is impossible to compare their output and the answers in the EDR corpus directly. '(ext)' means that the outputs of the analyzers were modified by hand and compared with the answers manually.

4.3 Experiment 2: POS Tagging

The experiment of pos tagging is also performed in the same way as the experiment of word segmentation. Figure 5 shows the result of this experiment.

4.4 Experiment 3: Processing Time

This evaluates the time to process 10000 sentences in the EDR corpus. We measured the sum of the user and the system time with the Unix 'time' command. The employed workstation was SUN SPARCserver 1000(SuperSPARC 85MHz). The average times of 5 trials are showed in Figure 6.

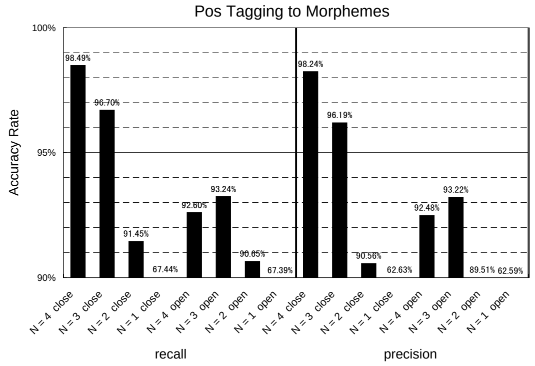

Figure 5: Accuracy in POS Tagging

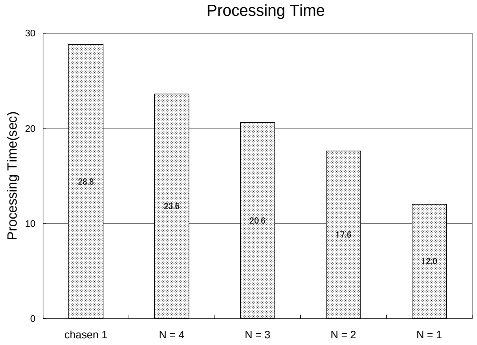

Figure 6: Processing time of 10000 sentences

5 Discussion

First, Figure 4 shows that in the close-data evaluation the larger N becomes, the higher rises its accuracy. It is almost true also in the open-data evaluation except for the reversal between $N = 3$ and $N = 4$(described later). This fact is a well-known characteristic of the k-NN method — the more examples are there in the database, the more accurate this method becomes.

The accuracy in the close-data evaluation is much higher than the accuracy in the open-data. This means that if input sentences have similar expressions to the stored examples, their analysis results become almost correct. This fact is also the advantage of the k-NN method. Because no intermediate calculation interferes, it can make the most of the stored examples, or the knowledge.

The best accuracy(95.18%) in the open-data evaluation is achieved with $N = 3$. (Yamamoto and Masuyama, 1997) achieved 95.91% under the open-data evaluation with the same corpus. These methods are comparable in word segmentation.

Next, Figure 5 shows the same tendency as Figure 4. The method with $N = 3$ achieved the best precision(93.22%) in the open-data evaluation but this is slightly lower than (Yamamoto and Masuyama, 1997). It is because the weight values in the equation (2) are all set to 1.0 in the experiments. In the word segmentation task the output labels of the k-NN method are only two and the influence of the weight is small. In the pos tagging, however, the number of the output labels is 15 and it is assumed that the selection of the most similar example was inadequate(see Section 6).

Finally, Figure 6 shows that our approach is very fast. It is natural because the main parts of the method, the search of strings and the estimate of probabilities are lightweight. Even the method with $N = 3$, which achieved the best result in the experiments, is much faster than 'chasen' — one of the fastest Japanese morphological analyzers.

Compared with (Yamamoto and Masuyama, 1997) employing the HMM-based method, our method is much faster. While the paper reported the processing time for 1000 sentences was 670sec with UltraSPARC 140MHz, in our method with $N = 3$ it took 2.06sec with SuperSPARC 85MHz.

As the size of the database becomes larger, the search of strings becomes slower. This decline of speed is inevitable and known as one of the problems in the k-NN method. To answer this problem some techniques such as pruning non-informative examples have been proposed(see Section 6).

6 Future Works

- Pruning non-informative examples

  The data extracted in section 4.1 contain all the possible examples along with non-informative ones. These non-informative examples are harmful not only to the processing speed but also to the accuracy in estimating $\hat{p}_i^B$ and $\hat{p}_i^t$. The pruning of non-informative examples is required(Aha et al., 1991).

- Acquisition of the optimal weight parameters

  The weight parameters in the equation (2) were not fully utilized in this paper. (Zavrel and Daelemans, 1997) utilized the k-NN method with the weight parameters calculated by the information gain and has achieved relatively good result in PP attachment and POS tagging tasks.

7 Conclusion

A stochastic approach based on HMM has some problems when applied to Japanese morphological analysis. In order to answer the problems, this paper proposed a new method of stochastic morphological analysis for Japanese texts employing the k-NN method and character n-gram data.

This method was evaluated thoroughly and it was showed that the method achieved high accuracy in word segmentation and pos tagging. Its accuracy is comparable to or better than other traditional morphological analyzers or other HMM-based stochastic methods. Moreover, it was also showed that the processing time of this method was much faster than the 'chasen' which is one of the fastest Japanese morphological analyzers.

From the experiments it is concluded that the stochastic approach employing character n-gram data and the k-NN method is practically competent for Japanese morphological analysis.

References

D. W. Aha, D. Kibler, and M. K. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Peter F. Brown, Della Vincent J. Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the Third Conference of Applied Natural Language Processing*, pages 133–143.

Ido Dagan and Fernando Pereira. 1994. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of ACL-94*.

B. V. Dasarathy. 1991. *Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press.

Evangelos Dermatas and George Kokkinakis. 1995. Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–163.

S.J. DeRose. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39.

Japan Electronic Dictionary Research Institute Ltd. 1995. Edr electronic dictionary technical guide.

Kyo Kageura. 1997. bigram (a method of segmenting compound kanji strings based on character bigram). In *Proceedings of The 3rd Annual Meeting of The Association for Natural Language Processing*, pages 477–480, mar.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401.

Yuji Matsumoto, Sadao Kurohashi, Takehito Utsuro, Yutaka Taeki, and Makoto Nagao, 1994. *Japanese Morphological Analysis System JUMAN Manual.* University of Kyoto.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Osamu Imaichi, and Tomoaki Imamura, 1997. *Japanese Morphological Analysis System ChaSen Manual.* Nara Institute of Science and Technology. NAIST Technical Report NAIST-IS-TR97007.

John G. McMahon and Francis J. Smith. 1996. Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics*, 22(2):217–247.

Bernard Merialdo. 1994. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

D. Michie, D. Spiegelhalter, and C. Taylor. 1994. *Machine Learning, Neural and Statistical Classification.* Prentice Hall.

Shinsuke Mori and Makoto Nagao. 1996. Japanese morphological analysis by superposition of morpheme bi-gram and part-of-speech bi-gram. In *Natural Language Processing SIG(Information Processing Society of Japan)*, pages 37–44. 112-6.

Masaaki Nagata. 1994. A stochastic japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *Proceedings of COLING*, pages 201–207.

Constantine P. Papageorgiou. 1994. Japanese word segmentation by hidden markov model. In *Proceedings of the Human Language Technology Workshop*, pages 283–288.

Kouichi Takeuchi and Yuji Matsumoto. 1995. Learning parameters of japanese morphological analyzer based-on hidden markov model. In *NLP Special Interest Group(Information Processing Society of Japan)*, pages 13–19. 108-3.

Mikio Yamamoto and Masakazu Masuyama. 1997. (japanese morphological analysis employing the chain probability of extended character with pos and breaking information). In *Proceedings of The 3rd Annual Meeting of The Association for Natural Language Processing*, pages 421–424, mar.

Kazuhide Yamamoto, Jun Kawai, Sumita Eiichiro, and Osamu Furuse. 1997. Morphological analysis utilizing n-gram of mixed category. In *Proceedings of the 54th annual meeting of Information Processing Society of Japan*, pages 2.51–2–52, March.

Jakub Zavrel and Walter Daelemans. 1997. Memory-based learning: Using similarity for smoothing. In *Proceedings of ACL/EACL(To appear)*.