Word Acquisition of Japanese Classical Literature Using State Transition Model

Makoto Suzuki†
Department of Information Science,
Shonan Institute of Technology, Kanagawa, Japan
Tel: (+81) 466-30-0204, Email: m-suzuki@info.shonan-it.ac.jp

Bin Xu
Graduate School of Shonan Institute of Technology, Kanagawa, Japan
Tel: (+81) 466-30-0204, Email: 15t2005@sit.shonan-it.ac.jp

Naohide Yamagishi
Graduate School of Shonan Institute of Technology, Kanagawa, Japan
Tel: (+81) 466-30-0204, Email: 11t2016@sit.shonan-it.ac.jp

Masayuki Goto
Department of Industrial and Management Systems,
School of Creative Science and Engineering,
Waseda University, Tokyo, Japan
Tel: (+81) 3-5286-3290, Email: masagoto@waseda.jp

Abstract. We have proposed a method for word segmentation using character *N*-grams previously. The proposed method uses a state transition model and character *N*-grams. Without using an existing dictionary, the proposed method can make a new original dictionary for the text document to be extracted. Of course, it is also possible to add a dictionary newly created to an existing dictionary. We showed that good results are obtained in Chinese which is completely different from Japanese in grammar in the previous paper. Furthermore, our method can also obtain good results even about comment data of "Nico Nico Douga (niconico.com)" which contains many new words or abbreviations. This paper is based on "The Tale of Genji" that is a famous story of Japanese classical literatures. It is difficult to divide words correctly only with the morphological analyzer of the modern Japanese because the classical rule of grammar is different from modern rule. However, we show that our method brings about a good result for the Tale of Genji in the same way as the modern Japanese. Therefore, the proposed method is considered to be effective for all languages which are expressed in Unicode and are not written with a space between words.

Keywords: word segmentation, character *N*-gram, language-independent, state transition, Genji Monogatari

1. INTRODUCTION

In recent years, the automatic analysis of text data proliferating on the Internet by a computer is getting more popular. However, in natural languages written without a space between words such as Chinese and Japanese, it is necessary to use a specific syntax of the language to perform a morphological analysis. Accurately dividing new words, spoken words, and abbreviations, which are being produced in large numbers on the Internet, is extremely difficult using conventional morphological analysis techniques because text data include many of these words. Therefore, a word segmentation method that can recognize some word information from the text data is required. The present paper is based on "The Tale of Genji" that is a famous story of Japanese classical literature. It is difficult to divide words correctly only with the morphological analyzer of the modern Japanese because the classical rule of grammar is different from the modern rule. However, we show that our method brings about a good result for the Tale of Genji in the same way as the modern Japanese. Therefore, the proposed method is considered to be effective for all languages which are expressed in Unicode and are not written with a space between words.

2. PREVIOUS METHOD

2.1 Probabilistic method

Mochihashi's method can automatically detect words from character strings in every language without a prepared dictionary (Mochihashi et al. 2009). Mochihashi's algorithm can divide words from a sentence including spoken words, abbreviations, and new words in all kinds of languages automatically, whereas previous algorithms were not able to deal with such processing. Mochihashi assumes a string as the output from a nonparametric Bayesian hierarchical *N*-gram language model of words and characters. Based on this assumption, "words" are iteratively estimated by a combination of Markov Chain Monte Carlo methods and efficient dynamic programming.

2.2 Neural-network-based representation learning

With the rapid development of neural-network-based representation learning, it has become realistic to learn features automatically. Lai's method is based on representation learning (Lai et al. 2013). In this previous study, all characters (including numbers and punctuation marks) are classified as belonging to one of four states: the beginning of a word is classified as belonging to state B, the contents of the word are classified as belonging to state M, the end of the word is classified as belonging to state E, and characters that make up words are classified as belonging to state S. This method calculates the relationships between characters by means of a neural network and classifies the characters into four states.

3. WORD ACQUISITION METHOD

3.1 State transition model

Our method of word segmentation is based on a state transition model (Yamagishi and Suzuki. 2011). We assume that documents are written based on a predetermined state transition model. Although some models of state transition have been proposed, we use a simple model in this paper, which is shown in Figure 1.

The state transition model is expressed under the assumption that the input document is described based on four states, namely, A, B, C, and D, specified beforehand. In this form, the A state represents the words (noun, verb, etc.); the state B represents the punctuation and particle conjunction (punctuation marks, conjunction, etc.); C state represents the postposition (namely, the word after a noun); and the D state represents the preposition (namely, the word before a noun). For example, the initial state is state B; the acceptance state can be either A, B, or C.

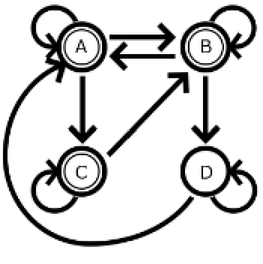
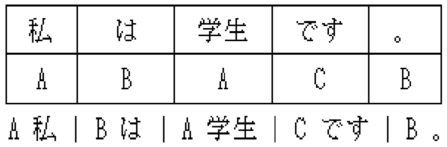


Figure 1: State transition diagram

An example sentence of word segmentation is shown in Figure 2. In this case, the word dictionary is divided into the following parts.



私	は	学生	です	。
A	B	A	C	B

A 私 | B は | A 学生 | C です | B 。

Figure 2: Example of a Japanese sentence

3.2 Algorithm of Word Acquisition

In the present paper, we propose a new word acquisition method. The proposed method creates a word segmentation dictionary automatically from a certain quantity of text data. Therefore, we use part of the target text data as the learning data to build the word dictionary. In addition, although the text data in the input have delimiters, we do not know what these delimiters are. Moreover, the text data contain words, but we do not know which character strings are words and which are tags (e.g., states A-D) of the character string.

The general flow of the proposed method is as follows.

First, we select the character strings that appear more often as candidate words by using an *N*-gram model. After that, we repeat each step I-IV, I) Initial selection, II) Word segmentation, III) Candidate refinement, and IV) Localization control, in seven phases basically.

Although each step differs slightly in each of seven phases, the steps are generally as follows:

I. Initial selection: We initialize the word dictionary if no flag is attached to any of the word candidates. We search the character strings that are above a certain frequency of appearance using 2- to 30-grams. The character strings are reduced one at a time starting from 30-grams. The number of occurrences of relatively large strings is deleted and not used in the subsequent string search. We perform this initial processing only during phase 1.

II. Word segmentation: We separate words using a word dictionary that we configured. We separate the learning documents using the maximum matching method. The character string is registered as a word candidate if the character string is between words, and we register the word candidate as a word if the candidate is not registered in the dictionary. We perform word separation using this dictionary by the subsequent procedures.

III. Candidate refinement: We organize the word dictionary by state transition diagram. Specifically, we register the candidate word that is more likely to be a word based on the results of the word segmentation. For example, a state B word is often sandwiched between state A words, or a state A word is often sandwiched between state B words. In other words, this processing is intended based on the state transition model. We repeatedly update the base dictionary based on each transition pattern in the state transition model until no update is generated.

IV. Localization control: We delay convergence in order to avoid over-fitting. At this point, we reset states A and B, and run steps I and II once more. At this point in time, we judge frequently appearing words to belong to state B. Furthermore, we delete words that appear to be incorrectly separated. For example, we delete character strings that do not appear in a word list or those that couple a state A word with a state B word.

Since we cannot trust the dictionary used in the initial selection and word segmentation, we gradually improve the reliability of the dictionary by candidate refinement and localization control. If a reliable dictionary can be obtained, a dictionary-based separation method will be used.

We repeat the aforementioned steps while changing the phase if the result converges to a certain degree. The phases are as follows:

1. Search for words belonging to state AB (steps I-IV)

   We separate the state A and B words based only on frequency of appearance using the character *N*-gram.

2. Search for words belonging to state AB + decompose the connection between state AA (steps II-IV)

   The character *N*-gram may recognize the character string of state AB as the character string of state A. State AB is defined as a continuous pattern of states A and B. State AA may be a continuous pattern of states A, B, and A. Here, state B may be hidden between two state A strings. We separate these character strings. Moreover, we combine these character strings, because these strings may be fixed phrases that appear frequently.

3. Reset the information of the states thus far (steps II-IV)

   We reset the words of state A and state B in the beginning of this step. High-frequency words are classified as belonging to state B in the dictionary.

4. Rerun Phases 1 and 2

   We consider the words with very high frequency to be B states and execute phases 1 and 2 again.

5. Search for state ABCD words + decompose and compose the connection between state AA (steps II-IV)

   We search state A, B, C, and D words based on the state transition model. We separate and combine the state AA words. This is the same processing described in phase 2.

6. Search for words belonging to state ABCD + compose the connections between state ABA words (steps II-IV)

   The process of "searching for words belonging to state ABCD" is the same as that described in phase 5. On the other hand, we combine the state ABA words. The state ABA words are a continuous pattern of state A, B, and A words, and these character strings may be fixed phrases that appear frequently.

7. Search for words belonging to state ABCD + partially compose the connections between state ABB words or state BBA words (steps II-IV)

   The process of "searching for words belonging to state ABCD" is the same as that described in phase 5. On the other hand, we combine the words belonging to states ABB and BBA. State ABB words are a continuous pattern of A, B, and B states. State BBA words are a continuous pattern of state B, B, and A words, and these character strings may be fixed phrases that appear frequently.

The proposed algorithm requires some parameters, the values of which must be set. However, the optimum values of these parameters vary with the number of input documents and the characteristics of the language. Even if we do not provide the optimum values for these parameters, we do not need to set the value strictly because the algorithm only decreases in execution efficiency at a certain range. Moreover, state B is exclusive and preferential to other states. In other words, a word that is classified as belonging to state B cannot belong to any other state.

4. EXPERIMENT

In this experiment, we used "The Tale of Genji" in the "Aozora Bunko[1]". The Tale of Genji is a famous story of Japanese classical literatures. We evaluated the final results in Phase 7 based on the following three criteria.

Criteria P: (the number of positions of "pause between words" in the estimated result that coincide with "pause between words" in the corpus) / (the number of positions that are automatically judged to be "pause between words" by the proposed algorithm)

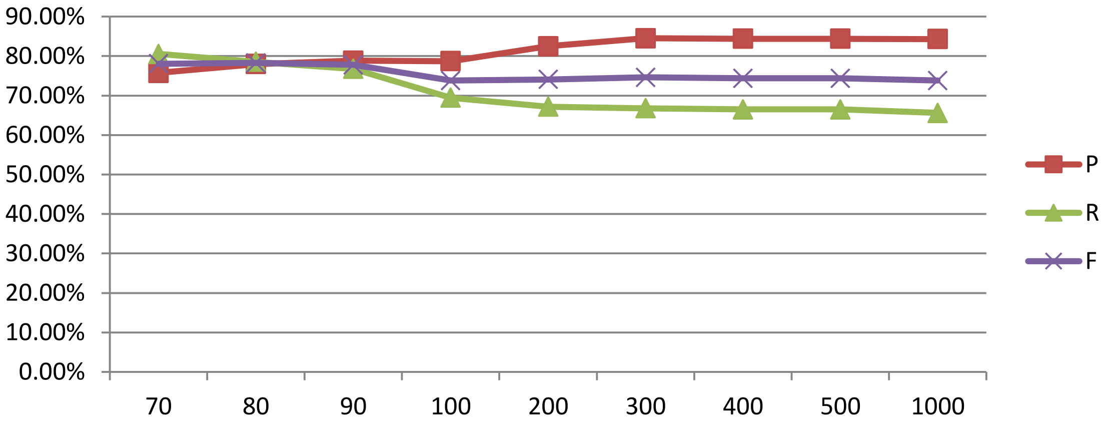


Figure 3: Correct-identification rate for parameter values in the range 70–1000.

Table 2: Correct-identification rate vs. parameter value

Parameters	70	80	90	100	200	300	400	500	1000
P (%)	75.76	77.99	78.83	78.70	82.50	84.50	84.41	84.39	84.31
R (%)	80.55	78.54	76.78	69.46	67.20	66.79	66.49	66.49	65.61
F (%)	78.08	78.26	77.79	73.79	74.07	74.61	74.39	74.38	73.79

Criteria R: (the number of positions of "pause between words" in the estimated result that coincide with "pause between words" in the corpus) / (the number of positions that are described as "pause between words" in the corpus)

Criteria F: The F-measure is a harmonic mean of the Precision and the Recall.

Here, we show the experimental results of the word segmentation.

We set the parameters in the program of the proposed method. These parameters are numerical values that determine the granularity of the word segmentation. The length of a word becomes longer as the value of the parameter becomes larger. We think that the result of the word segmentation is the best when the value of the parameter is 80 in case of the evaluation in Table 1.

Table 1: Evaluation result (The parameter is 80.)

Criteria	Value	Ratio
P	509,863/647,877	77.99%
R	509,863/734,071	78.54%
F	(2*P*R) / (P+R)	78.26%

[1] http://www.aozora.gr.jp/

Table 3: State of the word dictionary when the value of the parameter is 80

A	B	C	D	AC	AD	ACD	Del	None	Sum
16,324	27	0	0	6	8	0	0	27,833	44,198

Moreover, we show the results of the evaluation criteria of the case of changing the value of the parameter in Figure 3 and Table 2.

Here, we show an example of the word segmentation when the value of the parameter is 80 in Figures 4. Here, "[N]" means "none," which means the word was not classified because we did not have information enough to classify it.

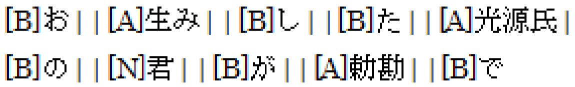

[B]お | | [A]生み | | [B]し | | [B]た | | [A]光源氏 |
[B]の | | [N]君 | | [B]が | | [A]勅勘 | | [B]で

Figure 4: Examples of the Japanese results (80)

On the other hand, we show an example of the word segmentation when the value of the parameter is 1000 in Figures 5.

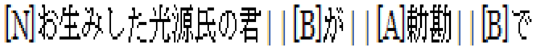

[N]お生みした光源氏の君 | | [B]が | | [A]勅勘 | | [B]で

Figure 5: Examples of the Japanese results (1,000)

In this way, there is a tendency that the length of a word becomes longer as the value of the parameter becomes larger.

We show the number of pseudo-words with each state in the automatically configured word dictionary as follows. In Table 3, the value of a cell is the number of words whose state corresponds to "A" or "ACD" when the value of the parameter is 80. Meanwhile, the value of "Del" is the number of deleted words or word candidates in the final step. Finally, the value of "None" is the number of unrecognized words.

In addition, we show 27 kinds of words with the state B the generated word dictionary when the parameter is 80 as follows.

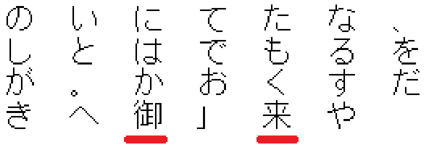

、をだ
なるすや
たもく来
てでお」
にはか御
いと。へ
のしがき

Figure 6: State B when the parameter is 80

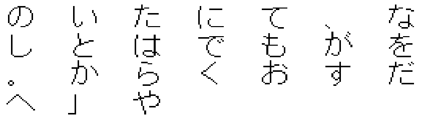

なをだ
、がす
ておお
にでく
たはらや
いとか」
のし。へ

Figure 7: State B when the parameter is 90

In Figure 7, we show 24 kinds of words with the state B in the generated word dictionary when the parameter is 90. It seems that the number of the state B words has been converged when the value of the parameter is about 90.

In this way, most of the words that were recognized as having a state B are appropriate for the Japanese delimiters, although there were a few words that we did not know well in the results. We underline the incorrectly segmented words in Figure 6.

5. CONSIDERATION

We now consider the effect of the parameter. This parameter is a numerical value used to determine the granularity of word segmentation. As the value of this parameter increases, the length of the character sequences that may be identified as individual words increases.

Figure 8 and Table 4 show experimental results for the correct-identification rate as the parameter is varied from 5 to 100. Comparing the results of Figures 1 and 2, we see that F values are higher for parameter values less than or equal to 100.

Meanwhile, Figure 9 and Table 5 show variations in the number of words identified as being in state B as a function of the parameter value. When the parameter value is 5, 132 words are identified as being in state B. The number of words identified in state B decreases suddenly as the parameter value approaches 30; for higher parameter values, this number gradually stabilizes and exhibits a trend of slow decrease.

Looking at the actual word segmentation results, we see that, when the parameter is set to 5, almost every individual character is distinguished as a word. Although the F value is high, the subdivision of words is too fine, with too many separators identified, yielding extremely poor results. Inspecting Figures 1 and 2 in the light of these results on word segmentation, we see that word segmentation results are best for parameter values in the approximate range 80 to 90, corresponding to where the curves of precision (P) and recall (R) intersect. This demonstrates the possibility of allowing a computer to determine the value of this parameter automatically.

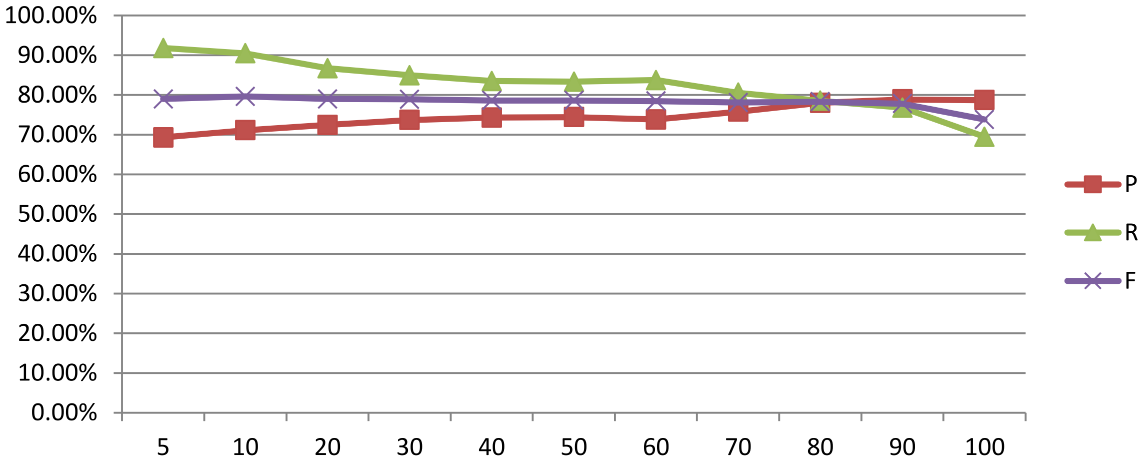


Figure 8: Correct-identification rate for parameter values in the range 5–100.

Table 4: Correct-identification rate vs. parameter value for parameter values less than or equal to 100

Parameters	5	10	20	30	40	50	60	70	80	90	100
P (%)	69.28	71.08	72.45	73.67	74.30	74.41	73.80	75.76	77.99	78.83	78.70
R (%)	91.81	90.48	86.73	84.92	83.47	83.33	83.71	80.55	78.54	76.78	69.46
F (%)	78.97	79.62	78.95	78.90	78.62	78.62	78.44	78.08	78.26	77.79	73.79

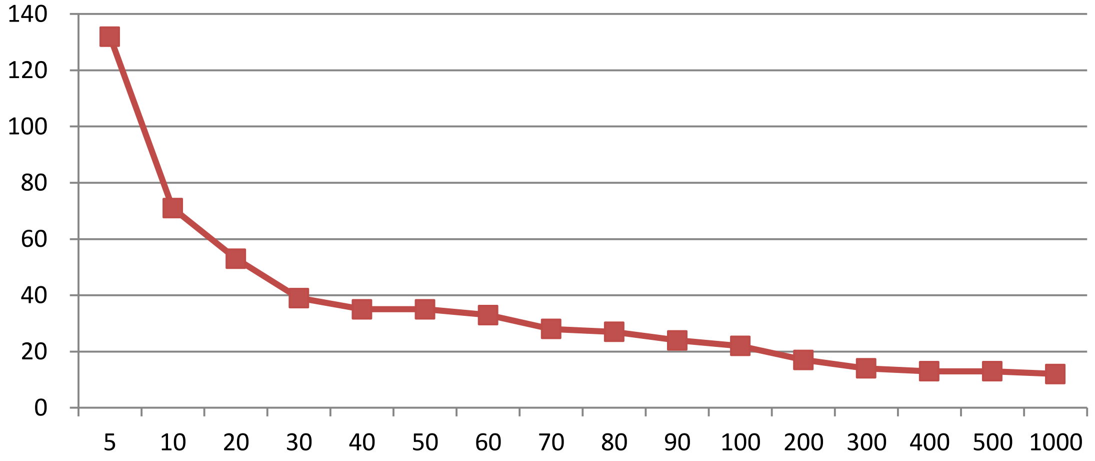


Figure 9: Number of words judged to be in state B for parameter values in the range 5–1000.

Table 5: Number of words judged to be in State B

Parameters	5	10	20	30	40	50	60	70	80	90	100	200	300	400	500	1000
The number of "State B Words"	132	71	53	39	35	35	33	28	27	24	22	17	14	13	13	12

6. CONCLUSION

By means of our experiments in this paper, we have shown that the proposed method facilitates the recognition of Japanese words by using the corpus of "The Tale of Genji" that is a famous story of Japanese classical literature.

The great advantage of the proposed method is that there is no need for us to set the dependency between individual words. We need to calculate the occurrence probability of the target word in the text data based on the probability method. Therefore, if we get new samples, it is necessary to recalculate the probability with the included samples. However, because there are no dependencies between words in the proposed method, we do not need to calculate the probability of the words again even if we get new samples. We can update the dictionary by adding a part containing the words detected from the new samples in the dictionary. Hence, once we can construct a dictionary that allows the extraction of words automatically with high accuracy, the proposed method is expected to detect new words or new abbreviations from the new samples quickly.

Moreover, we found the possibility to automatically determine the parameter by this experiment. In the future, it is desirable to construct the method adjusting the parameters automatically according to any language or text data. Furthermore, we would like to decrease the number of words that are incorrectly divided by our experiment.

ACKNOWLEDGMENT

The present study was supported by JSPS KAKENHI Grant Number 26282090 and 16K01267.

REFERENCES

Yamagishi, N. and Suzuki, M., (2011) An unsupervised word acquisition method by adaptation to a state transition model, Proc. of the 12th Student Paper Presentation of JIMA, 2011, 57-58 (in Japanese).

Mochihashi, D. and Yamada, T. and Ueda, N., (2009) Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling, ACL, 100–108

Lai, S. and Xu, L. and Chen, Y. and Liu, K. and Zhao, J., (2013) Chinese Word Segment Based on Character Repre-sentation Learning, Journal of Chinese Information Pro-cessing Vol. 27, No 5 (in Chinese).

Yoshimura, M., Kimura, F., and Maeda, A., (2011) Term Extraction for Text Analysis of Japanese Ancient Writings Based on Probability of Character *N*-grams, Proc. of the computers and the Humanities Symposium of Information Processing Society of Japan, 2011 Dec., 261-268 (in Japanese).