Chinese and Japanese Word Segmentation Using Word-Level and Character-Level Information

Tetsuji Nakagawa
Corporate Research and Development Center
Oki Electric Industry Co., Ltd.
2−5−7 Honmachi, Chuo-ku, Osaka 541-0053, Japan
nakagawa378@oki.com

Abstract

In this paper, we present a hybrid method for Chinese and Japanese word segmentation. Word-level information is useful for analysis of known words, while character-level information is useful for analysis of unknown words, and the method utilizes both these two types of information in order to effectively handle known and unknown words. Experimental results show that this method achieves high overall accuracy in Chinese and Japanese word segmentation.

1 Introduction

Word segmentation in Chinese and Japanese is an important and difficult task. In these languages, words are not separated by explicit delimiters, and word segmentation must be conducted first in most natural language processing applications. One of the problems which makes word segmentation more difficult is existence of unknown (out-of-vocabulary) words. Unknown words are defined as words that do not exist in a system's dictionary. The word segmentation system has no knowledge about these unknown words, and determining word boundaries for such words is difficult. Accuracy of word segmentation for unknown words is usually much lower than that for known words.

In this paper, we propose a hybrid method for Chinese and Japanese word segmentation, which utilizes both word-level and character-level information. Word-level information is useful for analysis of known words, and character-level information is useful for analysis of unknown words. We use these two types of information at the same time to obtain high overall performance.

This paper is organized as follows: Section 2 describes previous work on Chinese and Japanese word segmentation on which our method is based. Section 3 introduces the hybrid method which combines word-level and character-level processing. Section 4 shows experimental results of Chinese and Japanese word segmentation. Section 5 discusses related work, and Section 6 gives the conclusion.

2 Previous Work on Word Segmentation

Our method is based on two existing methods for Chinese or Japanese word segmentation, and we explain them in this section.

2.1 The Markov Model-Based Method

Word-based Markov models are used in English part-of-speech (POS) tagging (Charniak et al., 1993; Brants, 2000). This method identifies POS-tags $T = t_1, \ldots, t_n$, given a sentence as a word sequence $W = w_1, \ldots, w_n$, where n is the number of words in the sentence. The method assumes that each word has a state which is the same as the POS of the word and the sequence of states is a Markov chain. A state t transits to another state s with probability $P(s|t)$, and outputs a word w with probability $P(w|t)$. From such assumptions, the probability that the word sequence W with parts-of-speech T is generated is

$$\begin{aligned} P(W,T) &= \prod_{i=1}^{n} P(w_i t_i | w_0 t_0 \ldots w_{i-1} t_{i-1}), \\ &\simeq \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1}), \end{aligned} \quad (1)$$

where $w_0(t_0)$ is a special word(part-of-speech) representing the beginning of the sentence. Given a word sequence W, its most likely POS sequence $\hat{T}$ can be found as follows:

$$\begin{aligned} \hat{T} &= \operatorname*{argmax}_T P(T|W), \\ &= \operatorname*{argmax}_T \frac{P(W,T)}{P(W)}, \\ &= \operatorname*{argmax}_T P(W,T), \\ &\simeq \operatorname*{argmax}_T \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1}). \end{aligned} \quad (2)$$

The equation above can be solved efficiently by the Viterbi algorithm (Rabiner and Juang, 1993).

In Chinese and Japanese, the method is used with some modifications. Because each word in a

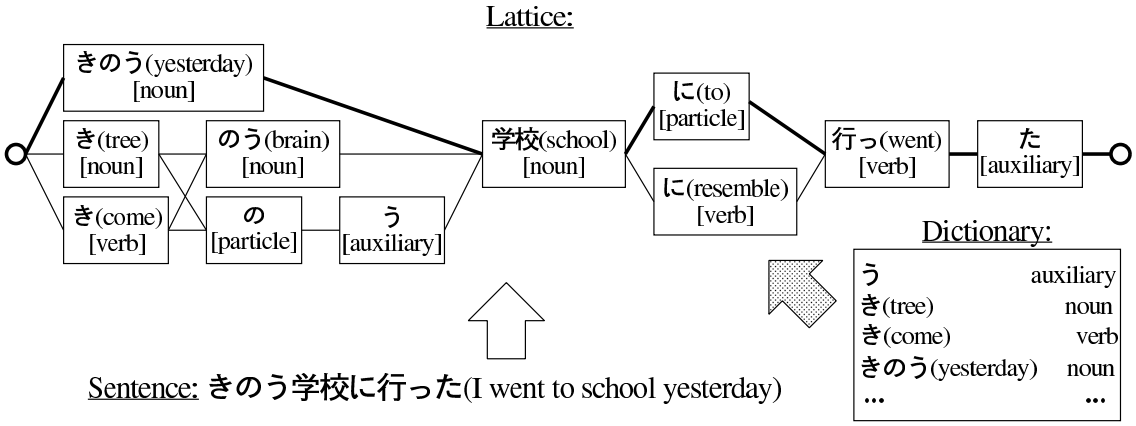

Figure 1: Example of Lattice Used in the Markov Model-Based Method

sentence is not separated explicitly in Chinese and Japanese, both segmentation of words and identification of the parts-of-speech tags of the words must be done simultaneously. Given a sentence S, its most likely word sequence $\hat{W}$ and POS sequence $\hat{T}$ can be found as follows where W ranges over the possible segments of S ($w_1 \cdots w_n = S$):

$$\begin{aligned}(\hat{W},\hat{T}) &= \underset{W,T}{\operatorname{argmax}}\, P(W,T|S),\\ &= \underset{W,T}{\operatorname{argmax}}\, \frac{P(W,T,S)}{P(S)},\\ &= \underset{W,T}{\operatorname{argmax}}\, P(W,T,S),\\ &= \underset{W,T}{\operatorname{argmax}}\, P(W,T),\\ &\simeq \underset{W,T}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}). \quad (3)\end{aligned}$$

The equation above can be solved using the Viterbi algorithm as well.

The possible segments of a given sentence are represented by a lattice, and Figure 1 shows an example. Given a sentence, this method first constructs such a lattice using a word dictionary, then chooses the best path which maximizes Equation (3).

This Markov model-based method achieves high accuracy with low computational cost, and many Japanese word segmentation systems adopt it (Kurohashi and Nagao, 1998; Matsumoto et al., 2001). However, the Markov model-based method has a difficulty in handling unknown words. In the constructing process of a lattice, only known words are dealt with and unknown words must be handled with other methods. Many practical word segmentation systems add candidates of unknown words to the lattice. The candidates of unknown words can be generated by heuristic rules(Matsumoto et al., 2001) or statistical word models which predict the probabilities for any strings to be unknown words (Sproat et al., 1996; Nagata, 1999). However, such heuristic rules or word models must be carefully designed for a specific language, and it is difficult to properly process a wide variety of unknown words.

Tag	Description
B	The character is in the beginning of a word.
I	The character is in the middle of a word.
E	The character is in the end of a word.
S	The character is itself a word.

Table 1: The '**B**, **I**, **E**, **S**' Tag Set

2.2 The Character Tagging Method

This method carries out word segmentation by tagging each character in a given sentence, and in this method, the tags indicate word-internal positions of the characters. We call such tags position-of-character (POC) tags (Xue, 2003) in this paper. Several POC-tag sets have been studied (Sang and Veenstra, 1999; Sekine et al., 1998), and we use the '**B**, **I**, **E**, **S**' tag set shown in Table 1 [1].

Figure 2 shows an example of POC-tagging. The POC-tags can represent word boundaries for any sentences, and the word segmentation task can be reformulated as the POC-tagging task. The tagging task can be solved by using general machine learning techniques such as maximum entropy (ME) models (Xue, 2003) and support vector machines (Yoshida et al., 2003; Asahara et al., 2003).

[1]The '**B**, **I**, **E**, **S**' tags are also called '**OP-CN**, **CN-CN**, **CN-CL**, **OP-CL**' tags (Sekine et al., 1998) or '**LL**, **MM**, **RR**, **LR**' tags (Xue, 2003).

Sentence: き の う | 学 校 | に | 行 っ | た
POC Tag: B I E B E S B E S

Figure 2: Example of the Character Tagging Method: Word boundaries are indicated by vertical lines ('|').

This character tagging method can easily handle unknown words, because known words and unknown words are treated equally and no other exceptional processing is necessary. This approach is also used in base-NP chunking (Ramshaw and Marcus, 1995) and named entity recognition (Sekine et al., 1998) as well as word segmentation.

3 Word Segmentation Using Word-Level and Character-Level Information

We saw the two methods for word segmentation in the previous section. It is observed that the Markov model-based method has high overall accuracy, however, the accuracy drops for unknown words, and the character tagging method has high accuracy for unknown words but lower accuracy for known words (Yoshida et al., 2003; Xue, 2003; Sproat and Emerson, 2003). This seems natural because words are used as a processing unit in the Markov model-based method, and therefore much information about known words (e.g., POS or word bigram probability) can be used. However, unknown words cannot be handled directly by this method itself. On the other hand, characters are used as a unit in the character tagging method. In general, the number of characters is finite and far fewer than that of words which continuously increases. Thus the character tagging method may be robust for unknown words, but cannot use more detailed information than character-level information.

Then, we propose a hybrid method which combines the Markov model-based method and the character tagging method to make the most of word-level and character-level information, in order to achieve high overall accuracy.

3.1 A Hybrid Method

The hybrid method is mainly based on word-level Markov models, but both POC-tags and POS-tags are used in the same time and word segmentation for known words and unknown words are conducted simultaneously.

Figure 3 shows an example of the method given a Japanese sentence "細川護熙首相が訪米", where the word "護熙"(person's name) is an unknown word. First, given a sentence, nodes of lattice for known words are made as in the usual Markov model-based method. Next, for each character in the sentence, nodes of POC-tags (four nodes for each character) are made. Then, the most likely path is searched (the thick line indicates the correct path in the example). Unknown words are identified by the nodes with POC-tags. Note that some transitions of states are not allowed (e.g. from **I** to **B**, or from any POS-tags to **E**), and such transitions are ignored.

Because the basic Markov models in Equation (1) are not expressive enough, we use the following equation instead to estimate probability of a path in a lattice more precisely:

$$\begin{aligned} P(W,T) &= \prod_{i=1}^{n} P(w_i t_i | w_0 t_0 \ldots w_{i-1} t_{i-1}), \\ &\simeq \prod_{i=1}^{n} \{\lambda_1 P(w_i|t_i) P(t_i) \\ &\quad +\lambda_2 P(w_i|t_i) P(t_i|t_{i-1}) \\ &\quad +\lambda_3 P(w_i|t_i) P(t_i|t_{i-2} t_{i-1}) \\ &\quad +\lambda_4 P(w_i t_i | w_{i-1} t_{i-1})\}, \\ &\quad (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1). \end{aligned} \quad (4)$$

The probabilities in the equation above are estimated from a word segmented and POS-tagged corpus using the maximum-likelihood method, for example,

$$P(w_i|t_i) = \begin{cases} \frac{f(w_i,t_i)}{\sum_w f(w,t_i)} & (f(w_i,t_i) > 0), \\ \frac{0.5}{\sum_w f(w,t_i)} & (f(w_i,t_i) = 0), \end{cases} \quad (5)$$

where $f(w,t)$ is a frequency that the word w with the tag t occurred in training data. Unseen events in the training data are handled as they occurred 0.5 times for smoothing. $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are calculated by deleted interpolation as described in (Brants, 2000). A word dictionary for a Markov model-based system is often constructed from a training corpus, and no unknown words exist in the training corpus in such a case. Therefore, when the parameters of the above probabilities are trained from a training corpus, words that appear only once in the training corpus are regarded as unknown words and decomposed to characters with POC-tags so that statistics about unknown words are obtained[2].

[2] As described in Equation (5), we used the additive smoothing method which is simple and easy to implement. Although there are other more sophisticated methods such as Good-Turing smoothing, they may not necessarily perform well because the distribution of words is changed by this operation.

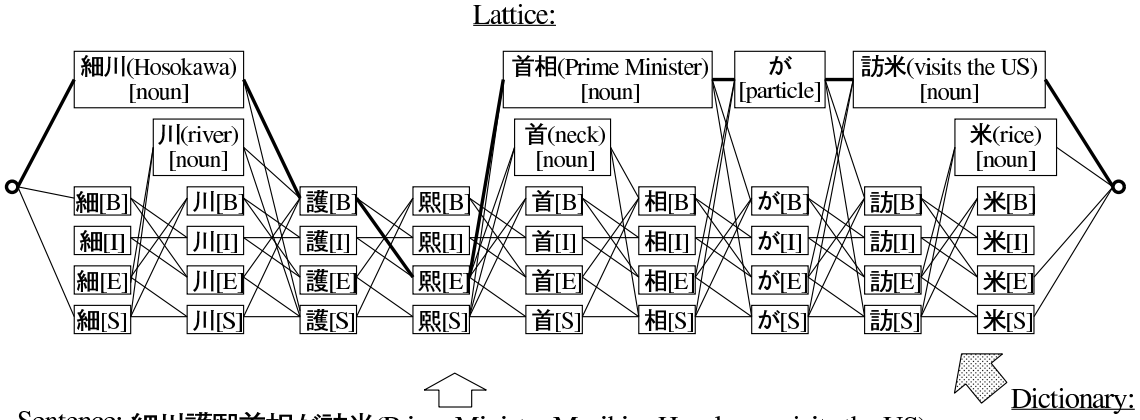

Figure 3: Example of the Hybrid Method

In order to handle various character-level features, we calculate word emission probabilities for POC-tags by Bayes' theorem:

$$\begin{aligned} & P(w_i|t_i) \\ = & \frac{P(t_i|w_i, t_i \in \mathbf{T}_{POC})P(w_i, t_i \in \mathbf{T}_{POC})}{P(t_i)}, \\ = & \frac{P(t_i|w_i, t_i \in \mathbf{T}_{POC}) \sum_{t \in \mathbf{T}_{POC}} P(w_i, t)}{P(t_i)}, \end{aligned} \quad (6)$$

where $\mathbf{T}_{POC} = \{\mathbf{B}, \mathbf{I}, \mathbf{E}, \mathbf{S}\}$, w_i is a character and t_i is a POC-tag. In the above equation, $P(t_i)$ and $P(w_i, t)$ are estimated by the maximum-likelihood method, and the probability of a POC tag t_i, given a character w_i ($P(t_i|w_i, t_i \in \mathbf{T}_{POC})$) is estimated using ME models (Berger et al., 1996). We use the following features for ME models, where c_x is the xth character in a sentence, $w_i = c_{i'}$ and y_x is the character type of c_x (Table 2 shows the definition of character types we used):

(1) Characters ($c_{i'-2}$, $c_{i'-1}$, $c_{i'}$, $c_{i'+1}$, $c_{i'+2}$)

(2) Pairs of characters ($c_{i'-2}c_{i'-1}$, $c_{i'-1}c_{i'}$, $c_{i'-1}c_{i'+1}$, $c_{i'}c_{i'+1}$, $c_{i'+1}c_{i'+2}$)

(3) Character types ($y_{i'-2}$, $y_{i'-1}$, $y_{i'}$, $y_{i'+1}$, $y_{i'+2}$)

(4) Pairs of character types ($y_{i'-2}y_{i'-1}$, $y_{i'-1}y_{i'}$, $y_{i'-1}y_{i'+1}$, $y_{i'}y_{i'+1}$, $y_{i'+1}y_{i'+2}$)

Parameters of ME are trained using all the words in training data. We use the Generalized Iterative Scaling algorithm (Darroch and Ratcliff, 1972) for parameter estimation, and features that appeared less than or equal to 10 times in training data are ignored in order to avoid overfitting.

What our method is doing for unknown words can be interpreted as follows: The method examines all possible unknown words in a sentence, and probability for an unknown word of length k, $w_i = c_j \cdots c_{j+k-1}$ is calculated as:

Character Type	Description
Alphabet	Alphabets
Numeral	Arabic and Chinese numerals
Symbol	Symbols
Kanji	Chinese Characters
Hiragana	Hiragana (Japanese scripts)
Katakana	Katakana (Japanese scripts)

Table 2: Character Types

$$\begin{aligned} & P(w_i t_i|h) \\ = & \begin{cases} P(c_j\mathbf{S}|h) & (k=1), \\ P(c_j\mathbf{B}|h) \prod_{l=j+1}^{j+k-2} P(c_l\mathbf{I}|h) P(c_{j+k-1}\mathbf{E}|h) & \\ & (k>1), \end{cases} \end{aligned} \quad (7)$$

where h is a history of the sequence. In other words, the probability of the unknown word is approximated by the product of the probabilities of the composing characters, and this calculation is done in the framework of the word-level Markov model-based method.

4 Experiments

This section gives experimental results of Chinese and Japanese word segmentation with the hybrid method. The following values are used to evaluate the performance of word segmentation:

R : Recall (The number of correctly segmented words in system's output divided by the number of words in test data)

P : Precision (The number of correctly segmented words in system's output divided by the number of words in system's output)

F : F-measure ($F = 2 \times R \times P/(R+P)$)

R_{known} : Recall for known words

$R_{unknown}$: Recall for unknown words

Corpus	# of Training Words	# of Testing Words (known/unknown)	# of Words in Dictionary	Rate of Unknown Words
AS	5,806,611	11,985 (11,727/ 258)	146,212	0.0215
HK	239,852	34,955 (32,463/2,492)	23,747	0.0713
PK	1,121,017	17,194 (16,005/1,189)	55,226	0.0692
RWCP	840,879	93,155 (93,085/ 70)	315,602	0.0008

Table 3: Statistical Information of Corpora

4.1 Experiments of Chinese Word Segmentation

We use three Chinese word-segmented corpora, the Academia Sinica corpus (AS), the Hong Kong City University corpus (HK) and the Beijing University corpus (PK), all of which were used in the First International Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003) at ACL-SIGHAN 2003.

The three corpora are word-segmented corpora, but POS-tags are not attached, therefore we need to attach a POS-tag (state) which is necessary for the Markov model-based method to each word. We attached a state for each word using the Baum-Welch algorithm (Rabiner and Juang, 1993) which is used for Hidden Markov Models. The algorithm finds a locally optimal tag sequence which maximizes Equation (1) in an unsupervised way. The initial states are randomly assigned, and the number of states is set to 64.

We use the following systems for comparison:

Bakeoff-1, 2, 3 The top three systems participated in the SIGHAN Bakeoff (Sproat and Emerson, 2003).

Maximum Matching A word segmentation system using the well-known maximum matching method.

Character Tagging A word segmentation system using the character tagging method. This system is almost the same as the one studied by Xue (2003). Features described in Section 3.1 (1)–(4) and the following (5) are used to estimate a POC tag of a character $c_{i'}$, where t_x is a POC-tag of the xth character in a sentence:

(5) Unigram and bigram of previous POC-tags ($t_{i'-1}, t_{i'-2}t_{i'-1}$)

All these systems including ours do not use any other knowledge or resources than the training data. In this experiments, word dictionaries used by the hybrid method and Maximum Matching are constructed from all the words in each training corpus. Statistical information of these data is shown in Table 3. The calculated values of λ_i in Equation (4) are shown in Table 4.

Corpus	λ_1	λ_2	λ_3	λ_4
AS	0.037	0.178	0.257	0.528
HK	0.048	0.251	0.313	0.388
PK	0.055	0.207	0.242	0.495
RWCP	0.073	0.105	0.252	0.571

Table 4: Calculated Values of λ_i

The results are shown in Table 5. Our system achieved the best F-measure values for the three corpora. Although the hybrid system's recall values for known words are not high compared to the participants of SIGHAN Bakeoff, the recall values for known words and unknown words are relatively well-balanced. The results of Maximum Matching and Character Tagging show the trade-off between the word-based approach and the character-based approach which was discussed in Section 3. Maximum Matching is word-based and has the higher recall values for known words than Character Tagging on the HK and PK corpus. Character Tagging is character-based and has the highest recall values for unknown words on the AS, HK and PK corpus.

4.2 Experiments of Japanese Word Segmentation

We use the RWCP corpus, which is a Japanese word-segmented and POS-tagged corpus.

We use the following systems for comparison:

ChaSen The word segmentation and POS-tagging system based on extended Markov models (Asahara and Matsumoto, 2000; Matsumoto et al., 2001). This system carries out unknown word processing using heuristic rules.

Maximum Matching The same system used in the Chinese experiments.

Character Tagging The same system used in the Chinese experiments.

As a dictionary for ChaSen, Maximum Matching and the hybrid method, we use IPADIC (Matsumoto and Asahara, 2001) which is attached to ChaSen. Statistical information of these data is shown in Table 3. The calculated values of λ_i in Equation (4) are shown in Table 4.

Corpus	System	R	P	F	R_{known}	$R_{unknown}$
AS	Hybrid method	0.973	0.971	0.972	0.979	0.717
	Bakeoff-1	0.966	0.956	0.961	0.980	0.364
	Bakeoff-2	0.961	0.958	0.959	0.966	0.729
	Bakeoff-3	0.944	0.945	0.945	0.952	0.574
	Maximum Matching	0.917	0.912	0.915	0.938	0.000
	Character Tagging	0.962	0.959	0.960	0.966	0.744
HK	Hybrid method	0.951	0.948	0.950	0.969	0.715
	Bakeoff-1	0.947	0.934	0.940	0.972	0.625
	Bakeoff-2	0.940	0.908	0.924	0.980	0.415
	Bakeoff-3	0.917	0.915	0.916	0.936	0.670
	Maximum Matching	0.908	0.830	0.867	0.975	0.037
	Character Tagging	0.917	0.917	0.917	0.932	0.728
PK	Hybrid method	0.957	0.952	0.954	0.970	0.774
	Bakeoff-1	0.962	0.940	0.951	0.979	0.724
	Bakeoff-2	0.955	0.938	0.947	0.976	0.680
	Bakeoff-3	0.955	0.938	0.946	0.977	0.647
	Maximum Matching	0.930	0.883	0.906	0.974	0.020
	Character Tagging	0.932	0.931	0.931	0.943	0.786

Table 5: Performance of Chinese Word Segmentation

Corpus	System	R	P	F	R_{known}	$R_{unknown}$
RWCP	Hybrid method	0.993	0.994	0.993	0.993	0.586
	ChaSen	0.991	0.992	0.991	0.991	0.243
	Maximum Matching	0.880	0.918	0.898	0.880	0.100
	Character Tagging	0.972	0.968	0.970	0.972	0.629

Table 6: Performance of Japanese Word Segmentation

The results are shown in Table 6[3]. Compared to ChaSen, the hybrid method has the comparable F-measure value and the higher recall value for unknown words (the difference is statistically significant at 95% confidence level). Character Tagging has the highest recall value for unknown words as in the Chinese experiments.

5 Discussion

Several studies have been conducted on word segmentation and unknown word processing. Xue (2003) studied Chinese word segmentation using the character tagging method. As seen in the previous section, this method handles known and unknown words in the same way basing on character-level information. Our experiments showed that the method has quite high accuracy for unknown words, but accuracy for known words tends to be lower than other methods.

[3]In this evaluation, R_{known} and $R_{unknown}$ are calculated considering words in the dictionary as known words. Words which are in the training corpus but not in the dictionary are handled as unknown words in the calculations. The number of known/unknown words of the RWCP corpus shown in Table 3 is also calculated in the same way.

Uchimoto et al. (2001) studied Japanese word segmentation using ME models. Although their method is word-based, no word dictionaries are used directly and known and unknown words are handled in a same way. The method estimates how likely a string is to be a word using ME. Given a sentence, the method estimates the probabilities for every substrings in the sentence. Word segmentation is conducted by finding a division of the sentence which maximizes the product of probabilities that each divided substring is a word. Compared to our method, their method can handle some types of features for unknown words such as "the word starts with an alphabet and ends with a numeral" or "the word consists of four characters". Our method cannot handle such word-level features because unknown words are handled by using a character as a unit. On the other hand, their method seems to have a computational cost problem. In their method, unknown words are processed by using a word as a unit, and the number of candidates for unknown words in a sentence which consists of n characters is equal to $n(n+1)/2$. Actually, they did not consider every substrings in a sentence, and limited the length of substrings to be less than or equal to five

characters. In our method, the number of POC-tagged characters which is necessary for unknown word processing is equal to $4n$, and there is no limitation for the length of unknown words.

Asahara et al. (2003) studied Chinese word segmentation based on a character tagging method with support vector machines. They preprocessed a given sentence using a word segmenter based on Markov models, and the output is used as features for character tagging. Their method is a character-based method incorporating word-level information and that is reverse to our approach. They did not use some of the features we used like character types, and our method achieved higher accuracies compared to theirs on the AS, HK and PK corpora (Asahara et al., 2003).

6 Conclusion

In this paper, we presented a hybrid method for word segmentation, which utilizes both word-level and character-level information to obtain high accuracy for known and unknown words. The method combines two existing methods, the Markov model-based method and character tagging method. Experimental results showed that the method achieves high accuracy compared to the other state-of-the-art methods in both Chinese and Japanese word segmentation. The method can conduct POS tagging for known words as well as word segmentation, but tagging identified unknown words is left as future work.

Acknowledgements

This work was supported by a grant from the National Institute of Information and Communications Technology of Japan.

References

Masayuki Asahara and Yuji Matsumoto. 2000. Extended Models and Tools for High-performance Part-of-Speech Tagger. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 21–27.

Masayuki Asahara, Chooi Ling Goh, Xiaojie Wang, and Yuji Matsumoto. 2003. Combining Segmenter and Chunker for Chinese Word Segmentation. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, pages 144–147.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Thorsten Brants. 2000. TnT — A Statistical Part-of-Speech Tagger. In *Proceedings of ANLP-NAACL 2000*, pages 224–231.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. Equations for Part-of-Speech Tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789.

J. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The annuals of Mathematical Statistics*, 43(5):1470–1480.

Sadao Kurohashi and Makoto Nagao. 1998. *Japanese Morphological Analysis System JUMAN version 3.61*. Department of Informatics, Kyoto University. (in Japanese).

Yuji Matsumoto and Masayuki Asahara. 2001. *IPADIC User's Manual version 2.2.4*. Nara Institute of Science and Technology. (in Japanese).

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. 2001. *Morphological Analysis System ChaSen version 2.2.8 Manual*. Nara Institute of Science and Technology.

Masaki Nagata. 1999. A Part of Speech Estimation Method for Japanese Unknown Words using a Statistical Model of Morphology and Context. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 227–284.

Lawrence R. Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. PTR Prentice-Hall.

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Proceedings of the 3rd Workwhop on Very Large Corpora*, pages 88–94.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing Text Chunks. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A Decision Tree Method for Finding and Classifying Names in Japanese Texts. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 171–177.

Richard Sproat and Thomas Emerson. 2003. The First International Chinese Word Segmentation Bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143.

Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. 1996. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The Unknown Word Problem: a Morphological Analysis of Japanese Using Maximum Entropy Aided by a Dictionary. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 91–99.

Nianwen Xue. 2003. Chinese Word Segmentation as Character Tagging. *International Journal of Computational Linguistics and Chinese*, 8(1):29–48.

Tatsumi Yoshida, Kiyonori Ohtake, and Kazuhide Yamamoto. 2003. Performance Evaluation of Chinese Analyzers with Support Vector Machines. *Journal of Natural Language Processing*, 10(1):109–131. (in Japanese).