A Stochastic Language Model using Dependency and Its Improvement by Word Clustering

Shinsuke Mori
Tokyo Research Labolatory,
IBM Japan, Ltd.
1623-14 Shimotsuruma
Yamatoshi, Japan

Makoto Nagao
Kyoto University
Yoshida-honmachi Sakyo
Kyoto, Japan

Abstract

In this paper, we present a stochastic language model for Japanese using dependency. The prediction unit in this model is an attribute of "bunsetsu". This is represented by the product of the head of content words and that of function words. The relation between the attributes of "bunsetsu" is ruled by a context-free grammar. The word sequences are predicted from the attribute using word n-gram model. The spell of Unknow word is predicted using character n-gram model. This model is robust in that it can compute the probability of an arbitrary string and is complete in that it models from unknown word to dependency at the same time.

1 Introduction

An effectiveness of stochastic language modeling as a methodology of natural language processing has been attested by various applications to the recognition system such as speech recognition and to the analysis system such as part-of-speech (POS) tagger. In this methodology a stochastic language model with some parameters is built and they are estimated in order to maximize its prediction power (minimize the cross entropy) on an unknown input. Considering a single application, it might be better to estimate the parameters taking account of expected accuracy of recognition or analysis. This method is, however, heavily dependent on the problem and offers no systematic solution, as far as we know. The methodology of stochastic language modeling, however, allows us to separate, from various frameworks of natural language processing, the language description model common to them and enables us a systematic improvement of each application.

In this framework a description on a language is represented as a map from a sequence of alphabetic characters to a probability value. The first model is C. E. Shannon's n-gram model (Shannon, 1951). The parameters of the model are estimated from the frequency of n character sequences of the alphabet (n-gram) on a corpus containing a large number of sentences of a language. This is the same model as used in almost all of the recent practical applications in that it describes only relations between sequential elements. Some linguistic phenomena, however, are better described by assuming relations between separated elements. And modeling this kind of phenomena, the accuracies of various application are generally augmented.

As for English, there have been researches in which a stochastic context-free grammar (SCFG) (Fujisaki et al., 1989) is used for model description. Recently some researchers have pointed out the importance of the lexicon and proposed lexicalized models (Jelinek et al., 1994; Collins, 1997). In these models, every headword is propagated up through the derivation tree such that every parent receives a headword from the head-child. This kind of specialization may, however, be excessive if the criterion is predictive power of the model. Research aimed at estimating the best specialization level for 2-gram model (Mori et al., 1997) shows a class-based model is more predictive than a word-based 2-gram model, a completely lexicalized model, comparing cross entropy of a POS-based 2-gram model, a word-based 2-gram model and a class-based 2-gram model, estimated from information theoretical point of view. As for a parser based on a class-based SCFG, Charniak (1997) reports better accuracy than the above lexicalized models, but the clustering method is not clear enough and, in addition, there is no report on predictive power (cross entropy or perplexity). Hogenhout and Matsumoto (1997) propose a word-clustering method based on syntactic behavior, but no language model is discussed. As the experiments in the present paper attest, word-class relation is dependent on language model.

In this paper, taking Japanese as the object language, we propose two complete stochastic language models using dependency between *bunsetsu*, a sequence of one or more content words followed by zero, one or more function words, and evaluate their predictive power by cross entropy. Since the number of sorts of *bunsetsu* is enormous, considering it as a symbol to be predicted would surely invoke the data-sparseness problem. To cope with this problem we

[0] This work is done when the auther was at Kyoto Univ.

use the concept of class proposed for a word n-gram model (Brown et al., 1992). Each *bunsetsu* is represented by the class calculated from the POS of its last content word and that of its last function word. The relation between *bunsetsu*, called dependency, is described by a stochastic context-free grammar (Fu, 1974) on the classes. From the class of a *bunsetsu*, the content word sequence and the function word sequence are independently predicted by word n-gram models equipped with unknown word models (Mori and Yamaji, 1997).

The above model assumes that the syntactic behavior of each *bunsetsu* depends only on POS. The POS system invented by grammarians may not always be the best in terms of stochastic language modeling. This is experimentally attested by the paper (Mori et al., 1997) reporting comparisons between a POS-based n-gram model and a class-based n-gram model induced automatically. We now propose, based on this report, a word-clustering method on the model we have mentioned above to successfully improve the predictive power. In addition, we discuss a parsing method as an application of the model.

We also report the result of experiments conducted on EDR corpus (Jap, 1993) The corpus is divided into ten parts and the models estimated from nine of them are tested on the rest in terms of cross entropy. As the result, the cross entropy of the POS-based dependency model is 5.3536 bits and that of the class-based dependency model estimated by our method is 4.9944 bits. This shows that the clustering method we propose improves the predictive power of the POS-based model notably. Additionally, a parsing experiment proved that the parser based on the improved model has a higher accuracy than the POS-based one.

2 Stochastic Language Model based on Dependency

In this section, we propose a stochastic language model based on dependency. Formally this model is based on a stochastic context-free grammar (SCFG). The terminal symbol is the attribute of a *bunsetsu*, represented by the product of the head of the content part and that of the function part. From the attribute, a word sequence that matches the *bunsetsu* is predicted by a word-based 2-gram model, and unknown words are predicted from POS by a character-based 2-gram model.

2.1 Sentence Model

A Japanese sentence is considered as a sequence of units called *bunsetsu* composed of one or more content words and function words. Let $Cont$ be a set of content words, $Func$ a set of function words and $Sign$ a set of punctuation symbols. Then *bunsetsu* is defined as follows:

$$Bnst = Cont^{+} Func^{*} \cup Cont^{+} Func^{*} Sign,$$

where the signs "+" and "∗" mean positive closure and Kleene closure respectively. Since the relations between *bunsetsu* known as dependency are not always between sequential ones, we use SCFG to describe them (Fu, 1974). The first problem is how to choose terminal symbols. The simplest way is to select each *bunsetsu* as a terminal symbol. In this case, however, the data-sparseness problem would surely be invoked, since the number of possible *bunsetsu* is enormous. To avoid this problem we use the concept of class proposed for a word n-gram model (Brown et al., 1992). All *bunsetsu* are grouped by the attribute defined as follows:

$$\begin{aligned} &attrib(b) \qquad (1)\\ &= \langle last(cont(b)),\ last(func(b)),\ last(sign(b)) \rangle, \end{aligned}$$

where the functions $cont$, $func$ and $sign$ take a *bunsetsu* as their argument and return its content word sequence, its function word sequence and its punctuation respectively. In addition, the function $last(\boldsymbol{m})$ returns the POS of the last element of word sequence $\boldsymbol{m}$ or `NULL` if the sequence has no word. Given the attribute, the content word sequence and the function word sequence of the *bunsetsu* are independently generated by word-based 2-gram models (Mori and Yamaji, 1997).

2.2 Dependency Model

In order to describe the relation between *bunsetsu* called dependency, we make the generally accepted assumption that no two dependency relations cross each other, and we introduce a SCFG with the attribute of *bunsetsu* as terminals. It is known, as a characteristic of the Japanese language, that each *bunsetsu* depends on the single *bunsetsu* appearing just before it. We say of two sequential *bunsetsu* that the first to appear is the anterior and the second is the posterior. We assume, in addition, that the dependency relation is a binary relation – that each relation is independent of the others. Then this relation is representing by the following form of rewriting rule of CFG: $B \Rightarrow AB$, where A is the attribute of the anterior *bunsetsu* and B is that of the posterior.

Similarly to terminal symbols, non-terminal symbols can be defined as the attribute of *bunsetsu*. Also they can be defined as the product of the attribute and some additional information to reflect the characteristics of the dependency. It is reported that the dependency is more frequent between closer *bunsetsu* in terms of the position in the sentence (Maruyama and Ogino, 1992). In order to model these characteristics, we add to the attribute of *bunsetsu* an

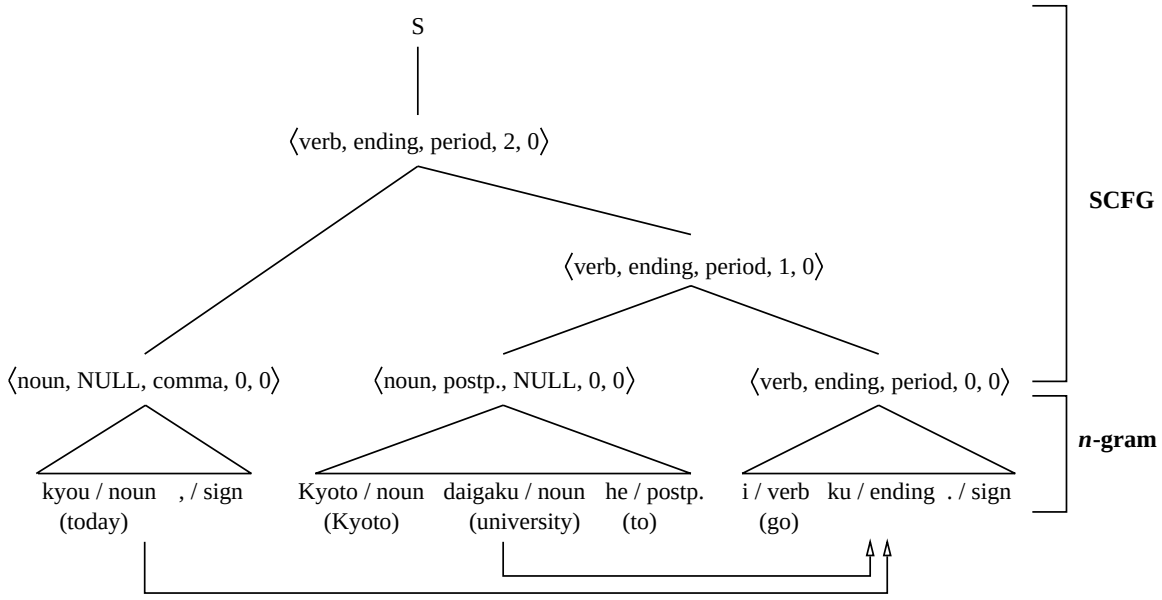

Figure 1: Dependency model based on *bunsetsu*

additional information field holding the number of *bunsetsu* depending on it. Also the fact that a *bunsetsu* has a tendency to depend on a *bunsetsu* with comma. For this reason the number of *bunsetsu* with comma depending on it is also added. To avoid data-sparseness problem we set an upper bound for these numbers. Let d be the number of *bunsetsu* depending on it and v be the number of *bunsetsu* with comma depending on it, the set of terminal symbols T and that of non-terminal symbols V is represented as follows (see Figure 1):

$$T = attrib(b) \times \{0\} \times \{0\}$$
$$V = attrib(b) \times \{1,\ 2,\ \cdots d_{max}\} \times \{0,\ 1,\ \cdots v_{max}\}.$$

It should be noted that terminal symbols have no *bunsetsu* depending on them. It follows that all rewriting rules are in the following forms:

$$S \Rightarrow \langle a,\ d,\ v\rangle \tag{2}$$
$$\langle a_1,\ d_1,\ v_1\rangle \Rightarrow \langle a_2,\ d_2,\ v_2\rangle\langle a_3,\ d_3,\ v_3\rangle \tag{3}$$
$$a_1 = a_3$$
$$d_1 = \min(d_3 + 1,\ d_{max})$$
$$v_1 = \begin{cases} \min(v_3 + 1,\ v_{max}) \\ \qquad\qquad \text{if } sign(a_2) = \text{comma} \\ v_3 \qquad\quad \text{otherwise} \end{cases}$$

where a is the attribute of *bunsetsu*.

The attribute sequence of a sentence is generated through applications of these rewriting rules to the start symbol S. Each rewriting rule has a probability and the probability of the attribute sequence is the product of those of the rewriting rules used for its generation. Taking the example of Figure 1, this value is calculated as follows:

$$\begin{aligned} &P(\langle \texttt{noun}, \texttt{NULL}, \texttt{comma}, 0, 0\rangle \\ &\quad \langle \texttt{noun}, \texttt{postp.}, \texttt{NULL}, 0, 0\rangle \\ &\quad \langle \texttt{verb}, \texttt{ending}, \texttt{period}, 0, 0\rangle) \\ &= P(S \Rightarrow \langle \texttt{verb}, \texttt{ending}, \texttt{period}, 2, 0\rangle) \\ &\quad \times P(\langle \texttt{verb}, \texttt{ending}, \texttt{period}, 2, 0\rangle \\ &\qquad \Rightarrow \langle \texttt{noun}, \texttt{NULL}, \texttt{comma}, 0, 0\rangle \\ &\qquad\quad \langle \texttt{verb}, \texttt{ending}, \texttt{period}, 1, 0\rangle) \\ &\quad \times P(\langle \texttt{verb}, \texttt{ending}, \texttt{period}, 1, 0\rangle \\ &\qquad \Rightarrow \langle \texttt{noun}, \texttt{postp.}, \texttt{NULL}, 0, 0\rangle \\ &\qquad\quad \langle \texttt{verb}, \texttt{ending}, \texttt{period}, 0, 0\rangle). \end{aligned}$$

The probability value of each rewriting rule is estimated from its frequency N in a syntactically annotated corpus as follows:

$$P(S \Rightarrow \langle a_1,\ d_1,\ v_1\rangle) = \frac{N(S \Rightarrow \langle a_1,\ d_1,\ v_1\rangle)}{N(S)}$$
$$P(\langle a_1,\ d_1,\ v_1\rangle \Rightarrow \langle a_2,\ d_2,\ v_2\rangle\langle a_3,\ d_3,\ v_3\rangle) = \frac{N(\langle a_1,\ d_1,\ v_1\rangle \Rightarrow \langle a_2,\ d_2,\ v_2\rangle\langle a_3,\ d_3,\ v_3\rangle)}{N(\langle a_1,\ d_1,\ v_1\rangle)}.$$

In a word n-gram model, in order to cope with data-sparseness problem, the interpolation technique is applicable to SCFG. The probability of the interpolated model of grammars G_1 and G_2, whose

probabilities are P_1 and P_2 respectively, is represented as follows:

$$P(A \Rightarrow \alpha) = \lambda_1 P_1(A \Rightarrow \alpha) + \lambda_2 P_2(A \Rightarrow \alpha)$$

$$0 \leq \lambda_j \leq 1 \ (j = 1,2) \ \text{ and } \ \lambda_1 + \lambda_2 = 1 \quad (4)$$

where $A \in V$ and $\alpha \in (V \cup T)^*$. The coefficients are estimated by held-out method or deleted interpolation method (Jelinek et al., 1991).

3 Word Clustering

The model we have mentioned above uses the POS given manually for the attribute of *bunsetsu*. Changing it into some class may improve the predictive power of the model. This change needs only a slight replacement in the model representing formula (1): the function *last* returns the class of the last word of a word sequence $\boldsymbol{m}$ instead of the POS. The problem we have to solve here is how to obtain such classes i.e. word clustering. In this section, we propose an objective function and a search algorithm of the word clustering.

3.1 Objective Function

The aim of word clustering is to build a language model with less cross entropy without referring to the test corpus. Similar research has been successful, aiming at an improvement of a word n-gram model both in English and Japanese (Mori et al., 1997). So we have decided to extend this research to obtain an optimal word-class relation. The only difference from the previous research is the language model. In this case, it is a SCFG in stead of a n-gram model. Therefore the objective function, called average cross entropy, is defined as follows:

$$\overline{H} = \frac{1}{m} \sum_{i=1}^{m} H(L_i, M_i), \quad (5)$$

where L_i is the i-th learning corpus and M_i is the language model estimated from the learning corpus excluding the i-th learning corpus.

3.2 Algorithm

The solution space of the word clustering is the set of all possible word-class relations. The cardinality of the set, however, is too enormous for the dependency model to calculate the average cross entropy for all word-class relations and select the best one. So we abandoned the best solution and adopted a greedy algorithm as shown in Figure 2.

Let $m_1, m_2, \ldots, m_n$ be $\mathcal{M}$ sorted
in the descending order of frequency.
$c_1 := \{m_1, m_2, \ldots, m_n\}$
$C = \{c_1\}$
foreach i $(1, 2, \cdots, n)$
  $f(m_i) := c_1$
foreach i $(1, 2, \cdots, n)$
  $c := \mathbf{argmin}_{c \in C \cup c_{new}} \overline{H}(move(f, m_i, c))$
  if $(\overline{H}(move(f, m_i, c)) < \overline{H}(f))$ **then**
    $f := move(f, m_i, c)$
    update interpolation coefficients.
    if $(c = c_{new})$ **then**
      $C := C \cup \{c_{new}\}$

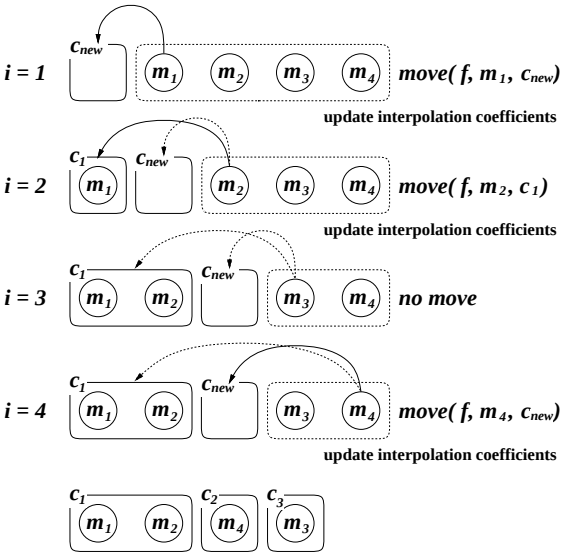

Figure 2: The clustering algorithm.

4 Syntactic Analysis

Syntactic Analysis is defined as a function which receives a character sequence as an input, divides it into a *bunsetsu* sequence and determines dependency relations among them, where the concatenation of character sequences of all the *bunsetsu* must be equal to the input. Generally there are one or more solutions for any input. A syntactic analyzer chooses the structure which seems the most similar to the human decision. There are two kinds of analyzer: one is called a rule-based analyzer, which is based on rules described according to the intuition of grammarians; the other is called a corpus-based analyzer, because it is based on a large number of analyzed examples. In this section, we describe a stochastic syntactic analyzer, which belongs to the second category.

4.1 Stochastic Syntactic Analyzer

A stochastic syntactic analyzer, based on a stochastic language model including the concept of dependency, calculates the syntactic tree (see Figure 1) with the highest probability for a given input $\boldsymbol{x}$ according to the following formula:

$$\hat{\boldsymbol{m}} = \mathop{\mathbf{argmax}}_{\boldsymbol{w}(T)=\boldsymbol{x}} P(T|\boldsymbol{x})$$

Table 1: Corpus.

	#sentences	#*bunsetsu*	#word
learning	174,524	1,610,832	4,251,085
test	19,397	178,415	471,189

$$= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\mathbf{argmax}}\, P(T|\boldsymbol{x})P(\boldsymbol{x})$$
$$= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\mathbf{argmax}}\, P(\boldsymbol{x}|T)P(T) \quad (\because \text{Bayes' formula})$$
$$= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\mathbf{argmax}}\, P(T) \quad (\because P(\boldsymbol{x}|T) = 1),$$

where $\boldsymbol{w}(T)$ represents the character sequence of the syntactic tree T. $P(T)$ in the last line is a stochastic language model including the concept of dependency. We use, as such a model, the POS-based dependency model described in section 2 or the class-based dependency model described in section 3.

4.2 Solution Search Algorithm

The stochastic context-free grammar used for syntactic analysis consists of rewriting rules (see formula (3)) in Chomsky normal form (Hopcroft and Ullman, 1979) except for the derivation from the start symbol (formula (2)). It follows that a CKY method extended to SCFG, a dynamic-programming method, is applicable to calculate the best solution in $O(n^3)$ time, where n is the number of input characters. It should be noted that it is necessary to multiply the probability of the derivation from the start symbol at the end of the process.

5 Evaluation

We constructed the POS-based dependency model and the class-based dependency model to evaluate their predictive power. In addition, we implemented parsers based on them which calculate the best syntactic tree from a given sequence of *bunsetsu* to observe their accuracy. In this section, we present the experimental results and discuss them.

5.1 Conditions on the Experiments

As a syntactically annotated corpus we used EDR corpus (Jap, 1993). The corpus was divided into ten parts and the models estimated from nine of them were tested on the rest in terms of cross entropy (see Table 1). The number of characters in the Japanese writing system is set to 6,879. Two parameters which have not been determined yet in the explanation of the models (d_{max} and v_{max}) are both set to 1. Although the best value for each of them can also be estimated using the average cross entropy, they are fixed through the experiments.

Table 2: Predictive power.

language model	#non-terminal +#terminal	cross entropy
POS-based model	576	5.3536
class-based model	10,752	4.9944

5.2 Evaluation of Predictive Power

For the purpose of evaluating the predictive power of the models, we calculated their cross entropy on the test corpus. In this process the annotated tree is used as the structure of the sentences in the test corpus. Therefore the probability of each sentence in the test corpus is not the summation over all its possible derivations. In order to compare the POS-based dependency model and the class-based dependency model, we constructed these models from the same learning corpus and calculated their cross entropy on the same test corpus. They are both interpolated with the SCFG with uniform distribution. The processes for their construction are as follows:

- POS-based dependency model
  1. estimate the interpolation coefficients in Formula (4) by the deleted interpolation method
  2. count the frequency of each rewriting rule on the whole learning corpus
- class-based dependency model
  1. estimate the interpolation coefficients in Formula (4) by the deleted interpolation method
  2. calculate an optimal word-class relation by the method proposed in Section 3.
  3. count the frequency of each rewriting rule on the whole learning corpus

The word-based 2-gram model for *bunsetsu* generation and the character-based 2-gram model as an unknown word model (Mori and Yamaji, 1997) are common to the POS-based model and class-based model. Their contribution to the cross entropy is constant on the condition that the dependency models contain the prediction of the last word of the content word sequence and that of the function word sequence.

Table 2 shows the cross entropy of each model on the test corpus. The cross entropy of the class-based dependency model is lower than that of the POS-based dependency model. This result attests experimentally that the class-based model estimated by our clustering method is more predictive than the POS-based model and that our word clustering

Table 3: Accuracy of each model.

language model	cross entropy	accuracy
POS-based model	5.3536	68.77%
class-based model	4.9944	81.96%
select always the next *bunsetsu*	–	53.10%

method is efficient at improvement of a dependency model.

We also calculated the cross entropy of the class-based model which we estimated with a word 2-gram model as the model M in the Formula (5). The number of terminals and non-terminals is 1,148,916 and the cross entropy is 6.3358, which is much higher than that of the POS-base model. This result indicates that the best word-class relation for the dependency model is quite different from the best word-class relation for the n-gram model. Comparing the number of the terminals and non-terminals, the best word-class relation for n-gram model is exceedingly specialized for a dependency model. We can conclude that word-class relation depends on the language model.

5.3 Evaluation of Syntactic Analysis

We implemented a parser based on the dependency models. Since our models, equipped with a word-based 2-gram model for *bunsetsu* generation and the character-based 2-gram as an unknown word model, can return the probability for any input, we can build a parser, based on our model, receiving a character sequence as input. Its evaluation is not easy, however, because errors may occur in *bunsetsu* generation or in POS estimation of unknown words. For this reason, in the following description, we assume a *bunsetsu* sequence as the input.

The criterion we adopted is the accuracy of dependency relation, but the last *bunsetsu*, which has no *bunsetsu* to depend on, and the second-to-last *bunsetsu*, which depends always on the last *bunsetsu*, are excluded from consideration.

Table 3 shows cross entropy and parsing accuracy of the POS-based dependency model and the class-based dependency model. This result tells us our word clustering method increases parsing accuracy considerably. This is quite natural in the light of the decrease of cross entropy.

The relation between the learning corpus size and cross entropy or parsing accuracy is shown in Figure 3. The lower bound of cross entropy is the entropy of Japanese, which is estimated to be 4.3033 bit (Mori and Yamaji, 1997). Taking this fact into consideration, the cross entropy of both of the models has stronger tendency to decrease. As for accuracy, there also is a tendency to get more accurate as the learning corpus size increases, but it is a strong tendency for the class-based model than for the POS-based model. It follows that the class-based model profits more greatly from an increase of the learning corpus size.

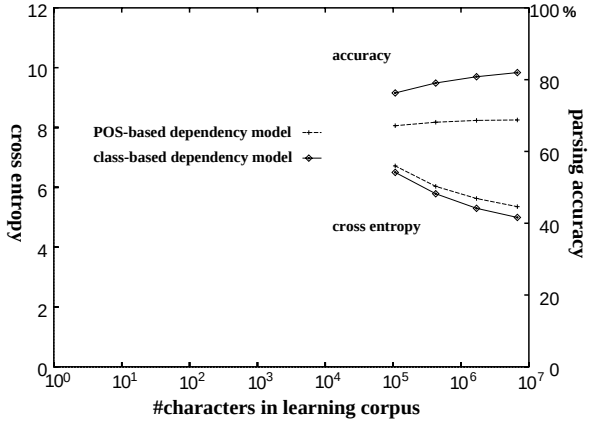

Figure 3: Relation between cross entropy and parsing accuracy.

6 Conclusion

In this paper we have presented dependency models for Japanese based on the attribute of *bunsetsu*. They are the first fully stochastic dependency models for Japanese which describes from character sequence to syntactic tree. Next we have proposed a word clustering method, an extension of deleted interpolation technique, which has been proven to be efficient in terms of improvement of the predictive power. Finally we have discussed parsers based on our model which demonstrated a remarkable improvement in parsing accuracy by our word-clustering method.

References

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

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 598–603.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23.

King Sun Fu. 1974. *Syntactic Methods in Pattern Recognition*, volume 12 of *Mathematics in Science and Engineering*. Accademic Press.

T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1989. A probabilistic parsing method for sentence disambiguation. In *Proceedings of the International Parsing Workshop*.

Wide R. Hogenhout and Yuji Matsumoto. 1997. A preliminary study of word clustering based on syntactic behavior. In *Proceedings of the Computational Natural Language Learning*, pages 16–24.

John E. Hopcroft and Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publishing.

Japan Electronic Dictionary Research Institute, Ltd., 1993. *EDR Electronic Dictionary Technical Guide*.

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

F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Rantnaparkhi, and S. Roukos. 1994. Decision tree parsing using a hidden derivation model. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 256–261.

Hiroshi Maruyama and Shiho Ogino. 1992. A statistical property of japanese phrase-to-phrase modifications. *Mathematical Linguistics*, 18(7):348–352.

Shinsuke Mori and Osamu Yamaji. 1997. An estimate of an upper bound for the entropy of japanese. *Transactions of Information Processing Society of Japan*, 38(11):2191–2199. (In Japanese).

Shinsuke Mori, Masafumi Nishimura, and Nobuyuki Ito. 1997. Word clustering for class-based language models. *Transactions of Information Processing Society of Japan*, 38(11):2200–2208. (In Japanese).

C. E. Shannon. 1951. Prediction and entropy of printed english. *Bell System Technical Journal*, 30:50–64.