A CLASSIFICATION TREE APPROACH TO AUTOMATIC SEGMENTATION OF JAPANESE COMPOUND SENTENCES

Yujie Zhang and Kazuhiko Ozeki

Department of Computer Science and Information Mathematics,
The University of Electro-Communications, Tokyo, Japan
Email: zhang@achilleus.cs.uec.ac.jp, ozeki@cs.uec.ac.jp

ABSTRACT

It is well known that direct parsing of a long Japanese compound sentence is extremely difficult. Various pre-processing methods have been proposed to segment such a sentence into shorter, simpler ones prior to parsing. The problem with the conventional methods is that some kind of segmentation patterns or heuristic preference scores must be given manually, hence no guarantee for optimality. This paper proposes a new method of sentence segmentation based on a classification tree technique. In this method, optimal segmentation patterns and the optimal order of their application are automatically acquired from training data, linguistic phenomena together with their occurence frequencies being taken into account. Generation of a classification tree is conducted on an EDR corpus, and evaluation results are reported. It is shown that pruning reduces the tree size by a factor of about 1/4 without affecting the performance.

1. INTRODUCTION

It is well known that direct parsing of a long Japanese compound sentence, comprising many coordinate clauses, is extremely difficult. Various pre-processing methods have been proposed to segment such a sentence into shorter, simpler ones prior to parsing [1]. Sentence segmentation have also been discussed from a view point of document revision support system [2], because a long compound sentence is difficult to understand even for humans. The techniques of sentence segmentation reported so far can be summarized as follows:

(1) Segmentation points are estimated by matching between prescribed segmentation patterns and an input sentence. The segmentation patterns are described by using the part-of-speech and the orthographic representation of morphemes obtained by morphological analysis [1].

(2) Dependency analysis on a clause sequence is conducted based on subordination relation among clauses [3]. Then dependency structure candidates are ordered by using heuristic dependency scores between clauses. Finally the segmentation points are determined in accordance with the top candidate for the dependency structure [2].

These techniques have been reported effective. However, the problem with these conventional methods is that the segmentation patterns or the heuristic dependency scores must be given manually, hence no guarantee for optimality.
This paper proposes a new method of automatic segmentation of long compound sentences

using a classification tree technique [4,5,6,7] based on the surface information obtained by morphological analysis. In this method, optimal segmentation patterns and the optimal order of their application are automatically acquired from training data, linguistic phenomena together with their occurrence frequencies being taken into account. The rest of the paper describes the details of the method, and reports the experimental results on an EDR corpus, including the effects of pruning.

2. CLASSIFICATION TREE

The classification tree employed in this work is of the following type:

(1) It is a binary tree: each intermediate node has two child-nodes.

(2) *Gini index* [4] is employed for the measurement of impurity.

(3) In the tree generation stage, if there is no test that reduces the impurity at a node, then the node is decided to be a leaf.

(4) A leaf is labeled with "YES" (segmentation point) or "NO" (not segmentation point) by the majority rule for the training data that reach the leaf.

The data given to the tree and the tests at tree nodes will be described in detail in the following sections.

3. DATA AND ATTRIBUTES

3.1 Segmentation Points

The syntactic unit employed here is *bunsetsu* phrase, which comprises a content word with or without being followed by a string of function words. In segmenting a long compound sentence, it is important to define precisely what a correct segmentation point should be. A correct segmentation point here is the boundary between two consecutive bunsetsu phrases X and Y, where X must satisfy the following conditions:

(1) X is not the sentence-final bunsetsu phrase.

(2) X is a predicate bunsetsu phrase containing such word as a verb or an adjective.

(3) X modifies (in a wide sense) the sentence-final bunsetsu phrase.

Segments obtained by dividing a Japanese sentence at such segmentation points are parallel, coordinate clauses.

3.2 Essential Phrases

A brief explanation of Japanese grammatical terms relevant to the present work will be appropriate here. *Taigen* refers to non-conjugating content words such as nouns and pronouns. *Yougen* refers to conjugating content words such as verbs, adjectives, adjectival nouns, and noun+copulas. A yougen changes its ending depending on its function. A base form is called a *shushi* form. When a yougen modifies a yougen, it takes a *renyou* form.

Table 1. Values for the conjunctive attribute. Morphemes 'te', 'de', 'tame', 'nagara', 'ba', 'ga (conjunctive)', 'ha','mo', 'ga (case)' are particles.

value	main word	last morpheme
v-renyou	verb, taigen+copula	renyou form
'te'-renyou	yougen	conjunctive particles 'te', 'de'
'tame'	yougen	formal noun, temporal noun
A	yougen	conjunctive particles 'nagara', and etc.
B	yougen	conjunctive particles 'ba', and etc.
C	yougen	conjunctive particles 'ga', and etc.
an-renyou	adjectival noun	renyou form
a-renyou	adjective	renyou form
v-rentai	verb, taigen+copula	rentai form
yougen	yougen	particles other than conjunctive particles
a-rentai	adjective, adjectival noun	rentai form
'ha'	taigen	kakari particle 'ha'
'mo'	taigen	kakari particle 'mo'
'ga'	taigen	case particle 'ga'
shushi	yougen	period '.'

When a yougen modifies a taigen, it takes a *rentai* form.
Some bunsetsu phrases in a sentence play an important role in estimating segmentation points, while others do not. A bunsetsu phrase whose final morpheme is a *kakari* particle such as 'ha' and 'mo', or a case particle 'ga' is considered to be important. A predicate bunsetsu phrase containing such word as a verb or an adjective is also important. Those important bunsetsu phrases are marked, and their attribute values are extracted. Three attributes

(1) conjunctive, (2) scope, (3) punctuation

are employed here. The values of the conjunctive attribute are defined according to the main word and the last morpheme in a bunsetsu phrase as in Table 1.

A,*B*, and *C* in the column "value" conform to the classification of conjunctive forms by Minami [3]. The value of the scope attribute is *scope* if the bunsetsu phrase contains a quotation particle 'to' or formal noun 'koto', and *null* otherwise. The value of the punctuation attribute is *punct* if the bunsetsu phrase is followed by a comma ',', and *null* otherwise. The important bunsetsu phrases represented by sets of attribute values defined above are referred to as essential phrases here. The following is an example of conversion from an ordinary sentence to a sequence of essential phrases. The suffix is the bunsetsu phrase number in the original sentence.

[Original Sentence]
16-nichi-ni (on 16th)$_1$ bei (American)$_2$ senseki (of registry)$_3$ tankah-ga (tanker [nominative])$_4$

hidan-shita (was shot)$_5$ toki, (when,)$_6$ kuehto-gun-ha (Kuwaiti forces [nominative])$_7$ misairu-no (missile's)$_8$ hirai-wo (coming [accusative])$_9$ tanchi, (detected,)$_{10}$ jigun-no (of their own forces)$_{11}$ chi-tai-kuh-misairu-de (with a surface to air missile)$_{12}$ geigeki-shiyou-to-shita-ga, (tried to intercept, but,)$_{13}$ shippai-ni (in failure)$_{14}$ owa-tta. (ended.)$_{15}$

[Essential Phrase Sequence]
(*'ga'*, *null*, *null*)$_4$ (*v-rentai*, *null*, *null*)$_5$ (*'tame'*, *null*, *punct*)$_6$ (*'ha'*, *null*, *null*)$_7$ (*v-renyou*, *null*, *punct*)$_{10}$ (*C*, *scope*, *punct*)$_{13}$ (*shushi*, *null*, *null*)$_{15}$

3.3 Candidates for Segmentation Points

It is probable that there is a correct segmentation point just after an essential phrase whose conjunctive attribute value is either *v-renyou*, *'te'*, *'tame'*, *A*, *B*, *C*, *an-renyou*, or *a-renyou*. This kind of essential phrases are referred to as segmentation phrases. The boundary between a segmentation phrase and the immediately succeeding bunsetsu phrase is a segmentation point candidate.

3.4 Data for Classification

It is obvious that attribute values of segmentation phrases are very important for estimation of segmentation points. Also, whether a segmentation point candidate is a correct one or not is decided by the bunsetsu phrase modified by the segmentation phrase. Therefore essential phrases that appear after the segmentation phrase are expected to play an important role in estimating a segmentation point, whereas essential phrases that appear before the segmentation phrase are considered unimportant. Thus only the segmentation phrase and the succeeding essential phrases are tested. An input data given to a classification tree is an essential phrase sequence, a segmentation phrase being at the top. Thus n data are generated from an essential phrase sequence with n segmentation phrases. The task for a classification tree then is to judge if a segmentation point candidate, which is the boundary between the segmentation phrase and the immediately succeeding bunsetsu phrase, is a correct one. Training data and evaluation data are labeled with "YES" (segmentation point) or "NO" (not segmentation point) by syntactic information obtained from a corpus.

4. TESTS AT TREE NODES FOR CLASSIFICATION

4.1 Form of a Test

The set of tests V is defined to be the product set

$$(\{\text{conjunctive attribute values}\} \cup \{*\}) \times \{scope, null, *\} \times \{punct, null, *\},$$

where '$*$' denotes a wild card that matches any attribute value. Also introduced is a symbol '$+$', which matches any non-empty essential phrase sequence. Then a test is represented by $[X] < Y >$, where X is an element in $V \cup \{+\}$, and Y is a sequence of elements in $V \cup \{+\}$ with no continuation of $+$'s. $[X]$ checks matching between X and a segmentation phrase, and $< Y >$ checks matching between Y and the sequence of essential phrases that appear after the segmentation phrase. For example, a data that passes the test

$$[(A, *, punct)]<(\text{'te'-renyou}, *, *) + (*, scope, *) +>$$

is one that satisfies the following conditions:

(1) The segmentation phrase has the conjunctive attribute value A and the punctuation attribute value *punct*. The scope attribute value does not matter.

(2) The essential phrase immediately after the candidate segmentation point has the conjunctive attribute value *'te'-renyou*. The scope and punctuation attribute values do not matter.

(3) There exists an essential phrase having the scope attribute value *scope* between the second essential phrase after the segmentation point candidate and the last essential phrase. The conjunctive and punctuation attribute values of the phrase do not matter.

4.2 Tests at Tree Nodes

The test at each node of a classification tree is determined in the tree generation process as follows. The *known test* for a node is the test which the training data reaching the node have just passed. Let $[X] < Y >$ be the known test for a node. By replacing '+' in X with 't' ($t \in V$), and by replacing '+' in Y with 't' ,'$+t$', '$t+$', and '$+t+$' ($t \in V$) successively, new tests are generated. Then the one that attains maximum impurity reduction is selected as the test at the node. This node, if it is not judged to be a leaf, is expanded into a 'yes' child-node, which collects the training data that pass the test, and a 'no' child-node, which collects the training data that fail to pass the test. The known test for the 'yes' child-node is set to be the same as the test at the parent-node, and the known test for the 'no' child-node is set to be the same as the known test for the parent-node. Started with the initial known test $[+] < + >$ for the root node, the above procedure is recursively executed, until a stopping condition is satisfied.

5. EXPERIMENTS

5.1 Experimental Data

From an EDR corpus [8], 2000 sentences, each of which has more than 30 morphemes, were randomly selected. Then the dependency structure for each sentence was determined by using bracket information given in the copus. It turned out that there were 1835 well-formed sentences among the 2000. A well-formed sentence here is one which satisfies the conditions that each non sentence-final bunsetsu phrase modifies one and only one succeeding bunsetsu phrase, and that two pairs of bunsetsu phrases in modification relation never cross with each other. The 1835 sentences were segmented into bunsetsu phrases, and the main word for each bunsetsu phrase was extracted by using the bracket information. Also, the conjugation form was determined for each phrase-final conjugating word by looking up a word dictionary attached to the corpus. Based on these results, sentences were then converted to essential phrase sequences, and segmentation phrases were detected to make experimental data. Each data was labeled 'YES' or 'NO' depending on whether the segmentation point candidate is correct or not as indicated by the bracket information. The resulting number of total data was 2484.

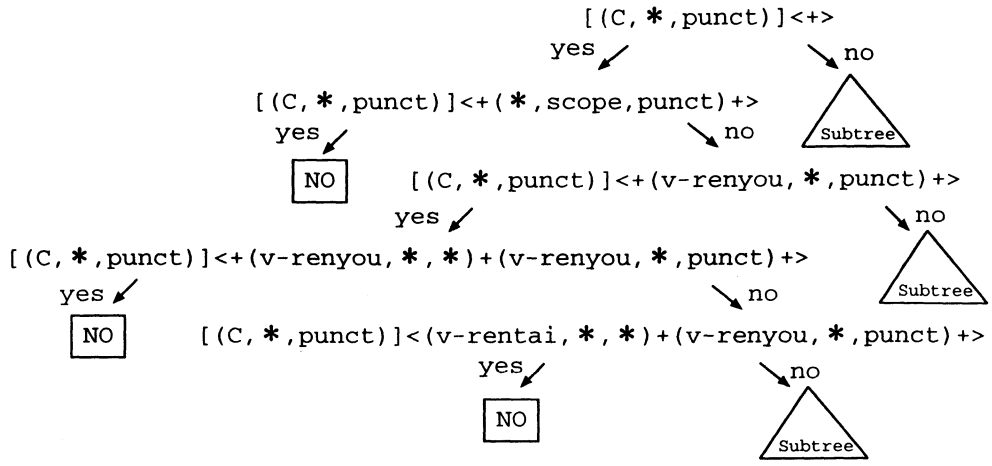

Fig.1 Part of the generated tree near the root.

5.2 Tree Generation and Segmentation Experiment

From the 1835 sentences, 400 sentences were randomly selected for creating evaluation data. The remaining 1435 sentences were used to make training data. The resulting numbers of the evaluation data and the training data were 555 and 1435, respectively. A classification tree was generated on the training data by using the method described above. The number of nodes in the generated tree was 771, among which 386 were leaves.

A part of the tree near the root related to the conjunctive attribute value C is shown in Fig.1. In this classification tree, an input data having a segmentation phrase with the conjunctive attribute value C and the punctuation attribute value *punct*, for example, first goes to a 'yes' child-node. Then the essential phrase sequence after the segmentation point candidate is tested. If there is an essential phrase having *scope* and *punct* attribute values between the essential phrase immediately after the segmentation point candidate and the final essential phrase in the sentence, then the data goes to a leaf with 'NO' class label, where the segmentation point candidate is judged not to be a segmentation point.

The performance of the classification tree was measured by using the evaluation data. There were 7 sentences among the 400 sentences that have no segmentation point candidates. Some examples of segmentation results are shown below. The symbol '?' designates a segmentation point candidate, and the suffix is the serial number for the segmentation point candidates. Estimation results are shown by (Y) (segmentation point) and (N) (not segmentation point). The symbol '|' indicates a correct segmentation point. Example 1 is the same sentence as the one that appeared in 3.2.

[Example 1]

16-nichi-ni (on 16th) bei (American) senseki (of registry) tankah-ga (tanker [nominative]) hidan-shita (was shot) toki, (when,) $?_1(N)$ kuehto-gun-ha (Kuwaiti forces [nominative]) misairu-no (missile's) hirai-wo (coming [accusative]) tanchi, (detected,) $?_2(N)$ jigun-no (of their own forces) chi-tai-kuh-misairu-de (with a surface to air missile) geigeki-shiyou-to-shita-ga, (tried to intercept, but,) $?_3(Y)|$ shippai-ni (in failure) owa-tta. (ended.)

[Example 2]
yamai-ga (disease [nominative]) susumi, (getting worse,) $?_1(Y)$ sutajio-no (in the studio) sofa-ni (on a sofa) yoko-ni (down) nari-nagara (lying) $?_2(N)$ shiji-wo (instructions [accusative]) dashite-ita (issuing) Kamei-san-ha, (Mr.Kamei [nominative],) saigo-no (last) rohru-ga (roll [nominative]) owatta-toki, (when finished,) namida-gunda. (eyes wet with tears.)

[Example 3]
yasumi-wo (holiday [accusative]) tora-nakere-ba, (if not take,) $?_1(N)$ tsugi-tsugi-ni (successively) kasan-sarete-yuku-node, (because of being accumulated,) $?_2(Y)$ matome-te (together) $?_3(N)$ moku, (Thursday,) kinyou-wo (Friday [accusative]) yasumi-ni (holiday) shite, (take, and,) $?_4(Y)$ shukyu-to (with a weekend) awase-te (joining together) $?_5(N)$ 4-renkyu-ni (4-day off) surukoto-mo (taking) dekiru. (possible.)

In Example 1, the estimation results are correct for all the segmentation point candidates. The segmentation point candidate '$?_1$' in Example 2, and '$?_2$', '$?_4$ in Example 3 are wrongly estimated to be segmentation points. However, the last bunsetsu phrase in Example 3 could be 'suru-koto-mo-dekiru' instead of 'dekiru', because in some parts of the EDR corpus, 'dekiru' is labeled as a function word. If we employ such bunsetsu phrase segmentation, then the estimation results for all the segmentation point candidates in Example 3 are turned into correct ones. Thus the type of errors as '$?_2$' and '$?_4$ in Example 3 are permissible ones. Such errors were corrected manually, and performance was evaluated in two different ways: one without such corrections, and the other with such corrections.

Evaluation measures employed in this work are as follows, and the evaluation results are shown in Table 2.

$$\text{Precision} = \frac{\#\text{correctly estimated segmentation points}}{\#\text{estimated segmentation points}}$$

$$\text{Recall} = \frac{\#\text{correctly estimated segmentation points}}{\#\text{correct segmentation points}}$$

$$\text{Accuracy} = \frac{\#\text{correctly segmented sentences}}{\#\text{evaluation sentences}}$$

Table 2. Evaluation of segmentation results. Figures in the parentheses show the results with manual corrections described above.

Precision %	81 (84)
Recall %	84 (84)
Accuracy %	72 (77)

The following problems were observed as to the cause of errors.

(a) When a segmentation point candidate is followed (not necessarily immediately) by an essential phrase that has the conjunctive attribute value *v-rentai* or the scope attribute value *scope*, estimation results are unreliable. This shows that it is difficult to estimate correctly the scope of a rentai clause and a quotation clause, as has been pointed out.
In Example 2 above, for example, the candidate '$?_1$' which is followed by an essential phrase 'dashite-ita (issuing)' with the conjunctive attribute value of *v-rentai*, was wrongly estimated as a segmentation point. In fact the segmentation phrase 'susumi, (getting worse,)' modifies 'dashite-ita (issuing)', not the sentence-final bunsetsu phrase.

(b) Morphological information given in the corpus is insufficient. In the EDR corpus, the classification of particles is rather coarse. For example, the particle 'to' has three functions: 'quotation', 'conjunctive', and 'parallel conjunction'. However, there is no label in the corpus to indicate which function 'to' has when it appears in a particular context.

(c) Some errors are obviously results from inconsistency of the bracket information in the corpus.

6. EFFECTS OF PRUNING

It is expected that pruning makes the classification tree more compact, and improves its generalization property. Various pruning methods have been proposed so far [4,7], from which one proposed by Gelfant *et al.* [9] was employed here. The method is described briefly as follows.
Let T be a classification tree, and D a set of data. The error rate for T evaluated on D is denoted by $R(T, D)$. Let the expression $T' \leq T$ denote that T' is a pruned subtree of T. Prepare two independent training data sets D^1 and D^2.
The pruning algorithm uses D^1 and D^2 alternately to grow and prune the tree in the following manner:

(1) Initially generate a full grown classification tree T_1 on D^1.

(2) Find a pruned subtree T_1^* of T_1 that minimizes the error rate on D^2:

$$T_1^* := \arg \min_{T' \leq T_1} R(T', D^2).$$

(3) Generate a full grown classification tree T_2 on D^2 extending branches from leaves of T_1^*.

(4) Find a pruned subtree T_2^* of T_2 that minimizes the error rate on D^1:

$$T_2^* := \arg \min_{T' \leq T_2} R(T', D^1).$$

(5) Generate a full grown classification tree T_3 on D^1 extending branches from leaves of T_2^*.

(6) Repeat 2 through 5 above, incrementing the suffix of T, until a stopping condition is satisfied.

This algorithm was applied to the current problem. Among the 1435 training sentences, 514 were unambiguous with respect to segmentation. The remaining 921 sentences were used as training sentences in this experiment. The 1339 training data generated from the 921 sentences were split into two sets D^1 (670 data) and D^2 (669 data). As the growing and pruning iteration proceeds, the size and the error rate of the classification tree changed as shown in Table 3. After k=6, the same results as in steps 4 and 5 alternately appeared.

Table 4 shows the size and the performance of the pruned trees measured on the 555 evaluation data described in 5.2. The last row of the table is for the unpruned tree described in 5.2. From this table, it is observed that pruning reduces the size of the tree by a factor of about 1/4 without affecting the performance, though it does not improve the performance for the evaluation data.

Table 3. Change of the tree size and the error rate by pruning. $|T|$ denotes the size (the number of nodes) of a tree T, and $j = (k \bmod 2) + 1$.

	Before pruning		After pruning	
Step k	$\|T_k\|$	$R(T_k, D^j)$	$\|T_k^*\|$	$R(T_k^*, D^j)$
1	431	0.283	167	0.214
2	519	0.312	183	0.175
3	457	0.275	201	0.197
4	519	0.312	183	0.175
5	457	0.275	201	0.197

Table 4. Performance of pruned trees measured on the evaluation data. The last row is for the unpruned tree.

With pruning	Tree size	Precision %	Recall %	Accuracy %
T_1^*	167	85	83	76
T_2^*	183	84	85	76
T_3^*	201	84	85	77
T_4^*	183	84	85	76
Without pruning	771	84	84	77

7. CONCLUSION

After a brief review of conventional techniques for segmentation of long Japanese compound sentences, a new method based on a classification tree technique was introduced. Generation of a classification tree was conducted on an EDR corpus, and evaluation results were reported. It was shown that pruning reduces the tree size by a factor of about 1/4 without affecting the performance, though it does not improve the performance in this application. To further improve the performance, more detailed morphological information will be necessary. Also, the problem of how to fully exploit morphological information in a classification tree technique, especially for determining the scope of a quotation clause and a rentai clause, remains open.

ACKNOWLEDGMENT

This work was supported in part by the Okawa Foundation for Information and Telecommunications.

REFERENCES

[1] Y. Kim and T. Ehara, "An automatic sentence breaking and subject supplement method for J/E machine translation," Trans. IPSJ, Vol. 36, No. 6, pp. 1018-1028, 1984 (in Japanese).

[2] E. Takeishi and Y. Hayashi, "Dividing Japanese complex sentences based on conjunctive expressions analysis," Trans. IPSJ, Vol. 33, No. 5, pp.652-663, 1992 (in Japanese).

[3] F. Minami, "The Structure of Modern Japanese," Taishukan-Shoten, 1974 (in Japanese).

[4] L. Breiman *et al.*, "Classification and Regression Trees," Chapman & Hall, 1984.

[5] Y. Zhang and K. Ozeki, "Automatic bunsetsu segmentation of Japanese sentences using a classification tree," Proc. PACLIC13, pp. 230-235, 1998.

[6] R. Kuhn and R. de Mori, "The application of semantic classification trees to natural language understanding," IEEE Trans. PAMI, Vol.17, No.5, pp.449-460, 1995.

[7] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, 1993.

[8] Japan Electronic Dictionary Research Institute, "Specification of EDR Electronic Dictionary Ver.1.5," 1996 (in Japanese).

[9] S. B. Gelfand, C. S. Ravishankar and E. J. Delp, "An iterative growing and pruning algorithm for classification tree design," IEEE Trans. PAMI, Vol.13, No.2, pp.163-174, 1991.