Towards Automatic Grammar Acquisition from a Bracketed Corpus

Thanaruk Theeramunkong
Japan Advanced Institute of
Science and Technology
Graduate School of Information Science
15 Asahidai Tatsunokuchi
Nomi Ishikawa 923-12 Japan
ping@jaist.ac.jp

Manabu Okumura
Japan Advanced Institute of
Science and Technology
Graduate School of Information Science
15 Asahidai Tatsunokuchi
Nomi Ishikawa 923-12 Japan
oku@jaist.ac.jp

Abstract

In this paper, we propose a method to group brackets in a bracketed corpus (with lexical tags), according to their local contextual information, as a first step towards the automatic acquisition of a context-free grammar. Using a bracketed corpus, the learning task is reduced to the problem of how to determine the nonterminal label of each bracket in the corpus. In a grouping process, a single nonterminal label is assigned to each group of brackets which are similar. Two techniques, *distributional analysis* and *hierarchical Bayesian clustering*, are applied to exploit local contextual information for computing similarity between two brackets. We also show a technique developed for determining the appropriate number of bracket groups based on the concept of entropy analysis. Finally, we present a set of experimental results and evaluate the obtained results with a model solution given by humans.

1 Introduction

Designing and refining a natural language grammar is a difficult and time-consuming task and requires a large amount of skilled effort. A hand-crafted grammar is usually not completely satisfactory and frequently fails to cover many unseen sentences. Automatic acquisition of grammars is a solution to this problem. Recently, with the increasing availability of large, machine-readable, parsed corpora, there have been numerous attempts to automatically acquire a CFG grammar through the application of enormous existing corpora[Lar90][Mil94][Per92][Shi95].

Lari and Young[Lar90] proposed so-called *inside-outside* algorithm, which constructs a grammar from an unbracketed corpus based on probability theory. The grammar acquired by this method is assumed to be in Chomsky normal form and a large amount of computation is required. Later, Pereira[Per92] applied this algorithm to a partially bracketed corpus to improve the computation time. Kiyono[Kiy94b][Kiy94a] combined symbolic and statistical approaches to extract useful grammar rules from a partially bracketed corpus. To avoid generating a large number of grammar rules, some basic grammatical constraints, local boundaries constraints and X bar-theory were applied. Kiyono's approach performed a refinement of an original grammar by adding some additional rules while the inside-outside algorithm tries to construct a whole grammar from a corpus based on Maximum Likelihood. However, it is costly to obtain a suitable grammar from an unbracketed corpus and hard to evaluate results of these approaches. As the increase of the construction of bracketed corpora, an attempt to use a bracketed (tagged) corpus for grammar inference was made by Shirai[Shi95]. Shirai constructed a Japanese grammar based on some simple rules to give a name (a label) to each bracket in the corpus. To reduce the grammar size and ambiguity, some hand-encoded knowledge is applied in this approach.

In our work, like Shirai's approach, we make use of a bracketed corpus with lexical tags, but instead of using a set of human-encoded predefined rules to give a name (a label) to each bracket, we introduce some statistical techniques to acquire such label automatically. Using a bracketed corpus, the grammar learning task is reduced to the problem of how to determine the nonterminal label of each bracket in the corpus. More precisely, this task is concerned with the way to classify brackets to some certain groups and give each group a label. We propose a method to group brackets in

a bracketed corpus (with lexical tags), according to their local contextual information, as a first step towards the automatic acquisition of a context-free grammar. In the grouping process, a single nonterminal label is assigned to each group of brackets which are similar. To do this, we apply and compare two types of techniques called *distributional analysis*[Har51] and *hierarchical Bayesian clustering*[Iwa95] for setting a measure representing similarity among the bracket groups. We also propose a method to determine the appropriate number of bracket groups based on the concept of entropy analysis. Finally, we present a set of experimental results and evaluate our methods with a model solution given by humans.

2 Grammar Acquisition with a Bracketed Corpus

In this section, we give a brief explanation of grammar acquisition using a bracketed corpus. In this work, the grammar acquisition utilizes a lexical-tagged corpus with bracketings. An example of the parse structures of two sentences in the corpus is shown graphically in Figure 1.

Sentence (1) : A big man slipped on the ice.
Parse Tree (1) : (((ART,"a")((ADJ,"big")(NOUN,"man")))
((VI,"slipped")((PREP,"on")((ART,"the")(NOUN,"ice")))))

Sentence (2) : The boy dropped his wallet somewhere.
Parse Tree (2) : (((ART,"the")(NOUN,"boy"))
(((VT,"dropped")((PRON,"his")(NOUN,"wallet")))
(ADV,"somewhere")))

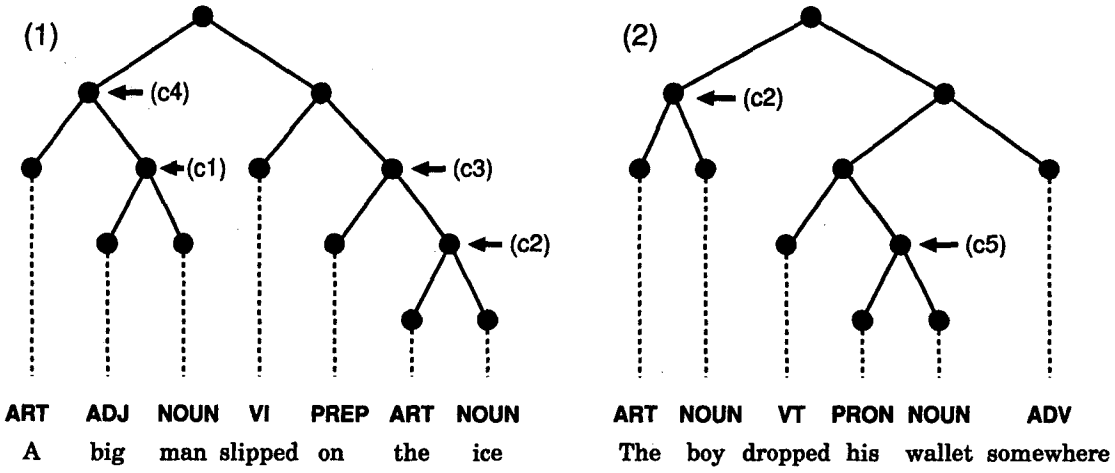

Figure 1: The graphical representation of the parse structures of *a big man slipped on the ice* and *the boy dropped his wallet somewhere*

In the parse structures, each terminal category (leaf node) is given a name (tag) while there is no label for each nonterminal category (intermediate node). With this corpus, the grammar learning task corresponds to a process to determine the nonterminal label of each bracket in the corpus. More precisely, this task is concerned with the way to classify the brackets into some certain groups and give each group a label. For instance, in Figure 1, it is reasonable to classify the brackets[1] (c2),(c4) and (c5) into a same group and give them a same label (e.g., NP(noun phrase)). As the result, we obtain three grammar rules: $NP \rightarrow (ART)(NOUN)$, $NP \rightarrow (PRON)(NOUN)$ and $NP \rightarrow (ART)(c1)$. To perform this task, our grammar acquisition algorithm operates in five stages as follows.

1. Assign a unique label to each node of which lower nodes are assigned labels. At the initial step, such node is one whose lower nodes are lexical categories[2]. This process is performed throughout all parse trees in the corpus.

[1] A bracket corresponds to a node in Figure 1.
[2] In Figure 1, there are three unique labels derived: $c_1 \rightarrow (ADJ)(NOUN)$, $c_2 \rightarrow (ART)(NOUN)$ and $c_5 \rightarrow (PRON)(NOUN)$.

2. Calculate the similarity of every pair of the derived labels.
3. Merge the most similar pair to a single new label(i.e., a label group) and recalculate the similarity of this new label with other labels.
4. Repeat (3) until a termination condition is detected. As the result of this step, a certain set of label groups is derived.
5. Replace labels in each label group with a new label in the corpus. For example, if (ART)(NOUN) and (PRON)(NOUN) are in the same label group, we replace them with a new label (such as NP) in the whole corpus.
6. Repeat (1)-(5) until all brackets(nodes) in the corpus are assigned labels.

In this paper, as a first step of our grammar acquisition, we focus on step (1)-(4), that is how to group nodes of which lower nodes are lexical categories. Figure 2 depicts an example of the grouping process.

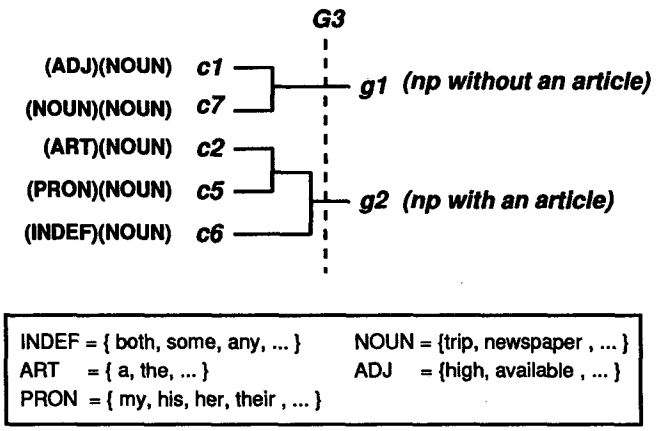

Figure 2: A part of the bracket grouping process

To compute the similarity of a pair of labels(in step 2), we propose two types of techniques called *distributional analysis* and *hierarchical Bayesian clustering* as shown in section 3. In section 4, we introduce the concept of *differential entropy* as the termination condition used in step (4).

3 Local Contextual Information as Similarity Measure

In this section, we describe two techniques which utilize "local context information" to calculate similarity between two labels. The term "local contextual information" considered here is represented by a pair of words immediately before and after a label. In the rest of this section, we first describe *distributional analysis* in subsection 3.1. Next, we give the concept of *Bayesian clustering* in subsection 3.2.

3.1 Distributional Analysis

Distributional analysis is a statistical method originally proposed by Harris[Har51] to uncover regularities in the distributional relations among the features of speech. Applications of this technique are varied[Bri92][Per93]. In this paper, we apply this technique to group similar brackets in a bracketed corpus. The detail of this technique is illustrated below.

Let P_1 and P_2 be two probability distributions over environments. The relative entropy between P_1 and P_2 is:

$$D(P1||P2) = \sum_{e \in Environments} P_1(e) \times \log \frac{P_1(e)}{P_2(e)}$$

Relative entropy $D(P1||P2)$ is a measure of the amount of extra information beyond $P2$ needed to describe $P1$. The *divergence* between P_1 and P_2 is defined as $D(P1||P2) + D(P2||P1)$, and is a measure of how difficult it is to distinguish between the two distributions. The environment is

a pair of words immediately before and after a label(bracket). A pair of labels is considered to be identical when they are distributionally similar, i.e., the divergence of their probability distributions over environments is low.

The probability distribution can be simply calculated by counting the occurrence of (c_i) and $(word_1\ c_i\ word_2)$. For the example in Figure 1, the numbers of appearances of (c_1), (c_2), (c_5), $(ART\ c_1\ VI)$, $(PREP\ c_2\ NULL)$ and $(VT\ c_5\ ADV)$ are collected from the whole corpus. $NULL$ stands for a blank tag representing the beginning or ending mark of a sentence.

Sparse Data Considerations

Utilizing divergence as a similarity measure, there is a serious problem caused by the sparseness of existing data or the characteristic of language itself. In the formula of relative entropy, there is a possibility that $P_2(e)$ becomes zero. In this condition, we cannot calculate the divergence of two probability distributions. To cope with this problem, we extend the original probability to one shown in the following formula.

$$P(a\ c_i\ b) = \lambda \frac{N(a\ c_i\ b)}{N(c_i)} + (1-\lambda)\frac{1}{Ntags^2}$$

where, $N(\alpha)$ is the occurrence frequency of α, $Ntags$ is the number of terminal categories and λ is a interpolation coefficient. The first term in the right part of the formula is the original estimated probability. The second term is generally called a uniform distribution, where the probability of an unseen event is estimated to a uniform fixed number. λ is applied as a balancing weight between the observed distribution and the uniform distribution. Intuitively, when the size of data is large, the small number should be used as λ. In the experimental results in this paper, we assigned λ with a value of 0.6 .

3.2 Hierarchical Bayesian Clustering Method

As a probabilistic method, hierarchical Bayesian clustering was proposed by Iwayama[Iwa95] to automatically classify given texts. It was applied to improve the efficiency and the effectiveness of text retrieval/categorization. Referring to this method, we try to make use of *Bayesian posterior probability* as another similarity measure for grouping the similar brackets. In this section, we conclude the concept of this measure as follows.

Let's denote a posterior probability with $P(G|C)$, where C is a collection of data (i.e., in Figure 2, $C = \{c_1, c_2, ..., c_N\}$) and G is a set of groups(clusters) (i.e., $G = \{g_1, g_2, ...\}$). Each group(cluster) g_j is a set of data and the groups are mutually exclusive. In the initial stage, each group is a singleton set; $g_i = \{c_i\}$ for all i. The method tries to select and merge the group pair that brings about the maximum value of the posterior probability[3] $P(G|C)$. That is, in each step of merging, this method searches for the most plausible situation that the data in C are partitioned in the certain groups G. For instance, at a merge step $k+1$ $(0 \leq k \leq N-1)$, a data collection C has been partitioned into a set of groups G_k. That is each datum c belongs to a group $g \in G_k$. The posterior probability at the merging step $k+2$ can be calculated using the posterior probability at the merging step $k+1$ as shown below (for more detail, see[Iwa95]).

$$P(G_{k+1}|C) = \frac{PC(G_{k+1})}{PC(G_k)} \frac{SC(g_x \cup g_y)}{SC(g_x)SC(g_y)} P(G_k|C)$$

Here $PC(G_k)$ corresponds to the prior probability that N random data are classified in to a set of groups G_k. As for the factor of $\frac{PC(G_{k+1})}{PC(G_k)}$, a well known estimate[Ris89] is applied and it is reduced to a constant value A^{-1} regardless of the merged pair. For a certain merging step, $P(G_k|C)$ is identical independently of which groups are merged together. Therefore we can use the following measure to select the best group pair to merge. The similarity between two bracket groups(labels), g_x and g_y, can be defined by $SIM(g_x, g_y)$. Here, the larger $SIM(g_x, g_y)$ is, the more similar two brackets are.

$$SIM(g_x, g_y) = \frac{SC(g_x \cup g_y)}{SC(g_x)SC(g_y)}$$

$$SC(g) = \prod_{c \in g} P(c|g)$$

[3] Maximizing $P(G|C)$ is a generalization of *Maximum Likelihood* estimation.

$$
\begin{aligned}
P(c|g) &= \sum_{e \in Environments} P(c|g,e)P(e|g) \\
&\approx \sum_{e} P(c|e)P(e|g) \\
&= P(c)\sum_{e} \frac{P(e|c)P(e|g)}{P(e)}
\end{aligned}
$$

where $SC(g)$ expresses the probability that all the labels in a group g are produced from the group, an elemental probability $P(c|g)$ means the probability that a group g produces its member c and $P(e|c)$ denotes a relative frequency of an environment e of a label c, $P(e|g)$ means a relative frequency of an environment e of a group g and $P(e)$ is a relative frequency of an environment e of the entire label set. In the calculation of $SIM(g_x, g_y)$, we can ignore the value of $P(c)$ because it occurs $|g_x \cup g_y|$ times in both denominator and numerator. Normally, $SIM(g_x, g_y)$ is ranged between 0 and 1 due to the fact that $P(c|g_x \cup g_y) \leq P(c|g_x)$ when $c \in g_x$.

4 Differential Entropy as Termination Condition

During iteratively merging the most similar labels, all labels will finally be gathered to a single group. Due to this, it is necessary to provide a criterion for determining whether this merging process should be continued or terminated. In this section, we describe a criterion named *differential entropy* which is a measure of entropy (perplexity) fluctuation before and after merging a pair of labels. Let c_1 and c_2 be the most similar pair of labels based on divergence or Bayesian posterior probability. Also let c_3 be the result label. $P_{c_1}(e)$, $P_{c_2}(e)$ and $P_{c_3}(e)$ are probability distributions over environment e of c_1, c_2 and c_3, respectively. P_{c_1}, P_{c_2} and P_{c_3} are estimated probabilities of c_1, c_2 and c_3, respectively. The differential entropy (ΔE) is defined as follows.

$$
\begin{aligned}
\Delta E &= \textit{Consequence Entropy} - \textit{Previous Entropy} \\
&= -P_{c_3} \times \sum_{e} P_{c_3}(e)\log P_{c_3}(e) \\
&\quad + P_{c_1} \times \sum_{e} P_{c_1}(e)\log P_{c_1}(e) + P_{c_2} \times \sum_{e} P_{c_2}(e)\log P_{c_2}(e)
\end{aligned}
$$

where $\sum_e P_{c_i}(e)\log P_{c_i}(e)$ is the total entropy over various environments of label c_i. The larger ΔE is, the larger the information fluctuation before and after merging becomes. Generally, we prefer a small fluctuation to a larger one. When ΔE is large, the current merging process introduces a large amount of information fluctuation and its reliability should be low. From this viewpoint, we apply this measure as a criterion for determining the termination of the merging process which will be given in the next section.

5 Preliminary Experimental Results

In this section, we show some results of our preliminary experiments to confirm effectiveness of the proposed techniques. The corpus we used is constructed by EDR and includes nearly 48,000 bracketed, tagged sentences[EDR94]. As mentioned in the previous sections, we focus on only the rules with lexical categories as their right hand side[4]. For instance, $c_1 \rightarrow (ADJ)(NOUN)$, $c_2 \rightarrow (ART)(NOUN)$ and $c_5 \rightarrow (PRON)(NOUN)$ in Figure 1. To evaluate our method, we use the rule tokens which appear more than 500 times in the corpus. Table 1 gives some characteristics of the corpus.

From the 35 initial rules, we calculate the similarity between any two rules (i.e., any rule pair) based on divergence and Bayesian posterior probability (BPP). For the divergence measure, the smaller the value is, the more similar the rule pair is. Inversely, for BPP, the larger the value is, the more similar the pair looks. After calculating all pairs' similarities, we merge the most similar pair (the minimum divergence or the maximum BPP) to a new label and recalculate the similarity of the new label with other remaining labels. The merging process is carried out in an iterative way. Figure 3 shows the minimum divergence (left) and the maximum Bayesian posterior probability (right) of each merge step.

In each iterative step of the merging process, we calculate differential entropy for both cases. The differential entropy of each step equals to the entropy difference between the entropy of two rules before merging and the entropy of a new rule after merging as described in the previous section.

[4]Other types of rules can be acquired in almost the same way and are left now as our further work.

No.of sentences	48259
No.of initial rules ($f > 500$)	35 (from total 761 rules)
Total number of rule tokens	136087 (from total 152925)

Table 1: Some features of the corpus

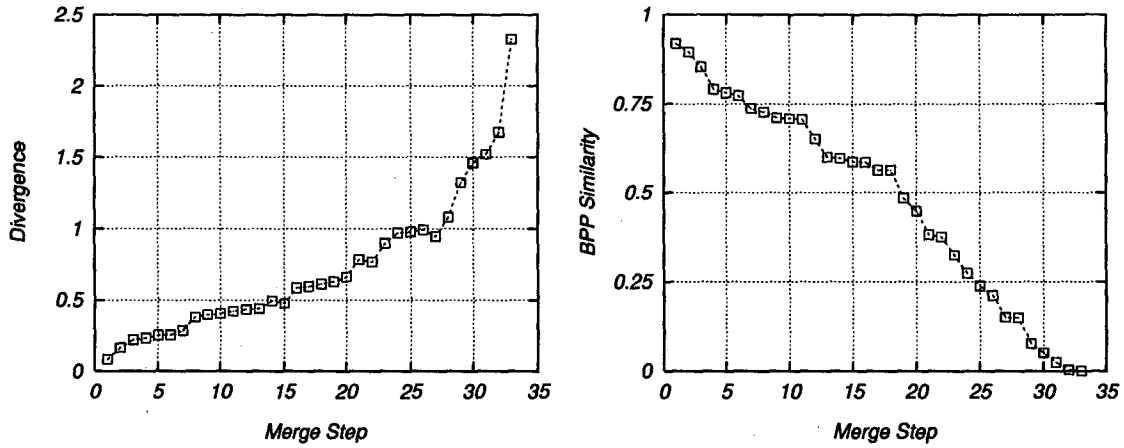

Figure 3: The minimum divergence (left) and the maximum Bayesian posterior probability (right) of each merge step

Two graphs in Figure 4 indicate the results of differential entropy (ΔE calculated by the formula in section 4) when the merging process advanced with divergence and BPP as its similarity measures. There are some sharp peaks indicating the rapid fluctuation of entropy in the graphs. In this work, we use these peaks as a clue to find the timing we should to terminate the merging process. As the result, we halt up the process at the 22nd step and the 27th step for the cases of divergence and BPP, respectively. Table 2 shows the obtained grouping results. In these tables, there are 13 groups for divergence and 8 groups for Bayesian posterior probability. To clarify the result in the tables, some sample words of each label are given in the appendix.

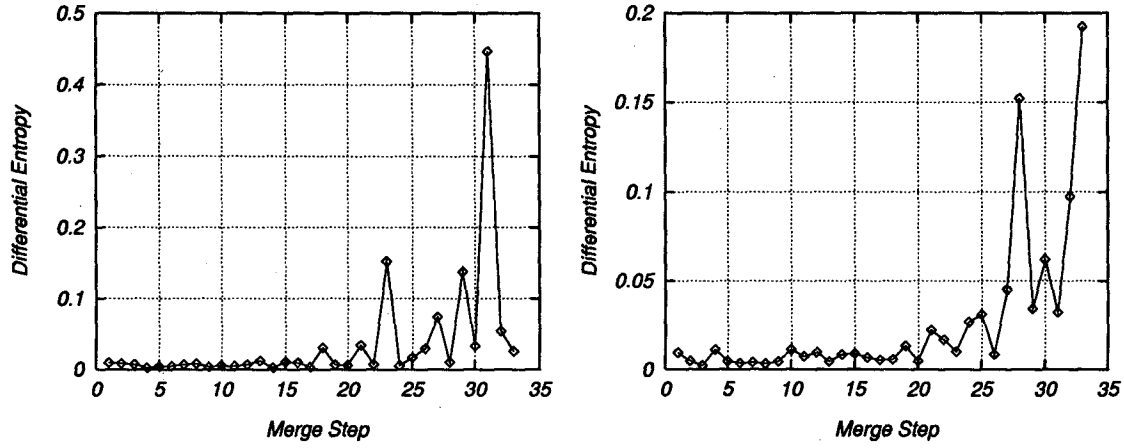

Figure 4: Differential entropy during the merging processes using divergence (left) and BPP (right)

We also made an experiment to evaluate these results with the solution given by three human evaluators (later called A, B and C) who are non-native but high-educated with more than 20 years of English education. The evaluators were told to construct 7-15 groups from the 35 initial rules, based on the grammatical similarity as they thought. As the result, the evaluators A, B and C classified the rules into 14, 13 and 14 groups, respectively.

Group	Members
1	(INDEF)(NOUN), (ART)(NOUN), (PRON)(NOUN), (DEMO)(NOUN), (NUM)(NOUN), (NUM)(UNIT), (NOUN)(NUM)
2	(ADJ)(NOUN), (NOUN)(NOUN), (NOUN)(CONJ)(NOUN)
3	(AUX)(VT)
4	(PREP)(NOUN), (PREP)(NUM), (PREP)(PRON), (ADV)(ADV), (PTCL)(VI)
5	(PTCL)(VT)
6	(VT)(NOUN), (VI)(ADV), (VT)(PRON), (AUX)(VI), (BE)(VI), (BE)(VT), (BE)(ADJ), (ADV)(VI), (VI)(PTCL)
7	(ADV)(ADJ)
8	(AUX)(ADV)
9	(ADV)(VT), (VT)(PTCL), (VI)(PREP)
10	(AUX)(BE)
11	(BE)(ADV)
12	(ADV)(BE)
13	(PRON)(VT)

Group	Members
1	(INDEF)(NOUN), (ART)(NOUN), (PRON)(NOUN), (DEMO)(NOUN), (NOUN)(CONJ)(NOUN)
2	(ADJ)(NOUN), (NOUN)(NOUN)
3	(AUX)(VT), (AUX)(BE), (BE)(ADV), (PTCL)(VT), (AUX)(ADV)
4	(PREP)(NOUN), (ADV)(ADV), (PREP)(PRON), (PTCL)(VI), (PREP)(NUM)
5	(VT)(NOUN), (VI)(ADV), (VT)(PRON), (AUX)(VI), (BE)(VI), (BE)(ADJ), (BE)(VT), (ADV)(VI), (VI)(PTCL)
6	(ADV)(ADJ), (NUM)(NOUN), (NUM)(UNIT)
7	(ADV)(VT), (VT)(PTCL), (VI)(PREP)
8	(NOUN)(NUM), (ADV)(BE), (PRON)(VT)

Table 2: The grouping result using divergence (left) and BPP (right)

	The Evaluator's Answer	
	Yes	No
The system says Yes	a	b
The system says No	c	d

Table 3: The number of entry pairs for evaluating accuracy

To evaluate the system with the model solutions, we applied a contingency table model as one shown in Table 3. This table model was introduced in [Swe69] and widely used in Information Retrieval and Psychology. In the table, a is the number of the label pairs which an evaluator assigned in the same group and so did the system, b is the number of the pairs which an evaluator did not assign in the same group but the system did, c is the number of the pairs which an evaluator assigned but the system did not, and d is the number of the pairs which both an evaluator and the system did not assign in the same group. From this table, we define seven measures, as shown below, for evaluating performance of the proposed methods. This evaluation technique was also applied partly for computing "closeness" between a system's answer and an evaluator's answer in [Hat93][Aga95][Iwa95]

- **Positive Recall (PR)** : $\frac{a}{a+c}$
- **Positive Precision (PP)** : $\frac{a}{a+b}$
- **Negative Recall (NR)** : $\frac{d}{b+d}$
- **Negative Precision (NP)** : $\frac{d}{c+d}$
- **Averaged Recall (AR)** : $\frac{PR+NR}{2}$
- **Averaged Precision (AP)** : $\frac{PP+NP}{2}$
- **F-measure (FM)** : $\frac{(\beta^2+1)\times PP\times PR}{\beta^2\times PP+PR}$

The F-measure is used as a combined measure of recall and precision, where β is the weight of recall relative to precision. Here, we use $\beta = 1.0$, which corresponds to equal weighting of the two measures. The results compared with three human evaluators are shown in Table 4.

	Similarity	Measures						
		PR	PP	NR	NP	AR	AP	FM
Evaluator A	Divergence	0.91	0.70	0.96	0.99	0.93	0.84	0.79
	BPP	0.63	0.46	0.92	0.96	0.77	0.71	0.53
Evaluator B	Divergence	0.73	0.66	0.95	0.97	0.84	0.81	0.69
	BPP	0.68	0.59	0.94	0.96	0.81	0.78	0.63
Evaluator C	Divergence	0.89	0.66	0.95	0.99	0.92	0.82	0.76
	BPP	0.80	0.57	0.94	0.98	0.87	0.77	0.66
Averaged	Divergence	0.84	0.67	0.95	0.98	0.90	0.82	0.75
	BPP	0.70	0.54	0.93	0.97	0.82	0.75	0.61

Table 4: Evaluation results using three human evaluators' solutions

From these results, we observe some features as follows. The divergence gives a better solution than Bayesian posterior probability does. Normally, the positive measures (PR and PP) have smaller values than the negative ones (NR and NP) do. This means that it is difficult to judge two labels to be in a same group rather than to judge them to be in a separate group. Using divergence as a similarity measure, we get, on average, 84 % positive recall and 67 % positive precision and up to 90 % and 82 % when considering both positive and negative measures. Even for the worst result(Evaluator B), we can get up to 84 % and 81 % for averaged recall and precision. In order to confirm the performance of the system, the evaluators' results are compared with each other. This comparison is useful for investigating the difficulty of the grouping problem. The comparison result is shown in Table 5. At this point, we can observe that the label grouping process is a hard problem that may make an evaluator's solution inconsistent with the others' solutions. However, our proposed method seem to give a reconciliation solution between those solutions. Especially, the method which applies divergence as the similarity measure, has a good performance in grouping brackets in the bracketed corpus.

	Measures						
	PR	PP	NR	NP	AR	AP	FM
A + B	0.47	0.55	0.95	0.94	0.71	0.74	0.51
B + A	0.55	0.47	0.94	0.95	0.74	0.71	0.51
B + C	0.83	0.68	0.96	0.98	0.90	0.83	0.75
C + B	0.68	0.83	0.98	0.96	0.83	0.90	0.75
C + A	0.55	0.57	0.96	0.95	0.76	0.76	0.56
A + C	0.57	0.55	0.95	0.96	0.76	0.76	0.56
Averaged	0.61	0.61	0.96	0.96	0.78	0.78	0.61

Table 5: Comparing the grouping results obtained by the evaluators(A,B,C)

We also make an experiment to evaluate whether divergence is a better measure than BPP, and whether the application of differential entropy to cut off the merging process is appropriate. This examination can be held by plotting values of recall, precision and F-measure during each step of merging process. Figure 5 shows the fluctuation of positive recall(PR), positive precision(AP), averaged recall(AR), averaged precision and F-measure (FM).

From the graphs, we found out that the maximum value of F-measure is 0.75 in the case of divergence while it is only 0.65 in the case of BPP. That is, divergence provides a better solution than BPP. Moreover, the 22nd an 25th merge steps were the most suitable points to terminate the merging process for divergence and BPP, respectively. This result is consistent with the grouping result of our system (13 groups) in the case of divergence. Although differential entropy leads us to terminate the merging process at the 27th merge step in the case of BPP, we observe that there is just a little difference between the F-measure value of the 25th merge step and that of the 27th merge step. From this result, we conclude that differential entropy can be used a good measure to predict the cut-off timing of the merging process.

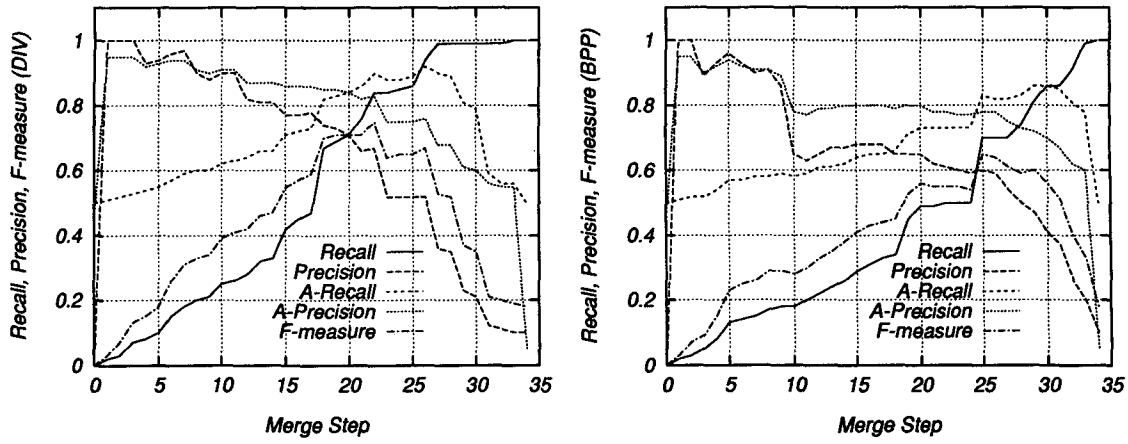

Figure 5: The transition of PR, PP, AR, AP, FM during the merging process using divergence(left) and BPP(right) as the similarity measures

6 Conclusion

There has been an increase of the construction of many types of corpora, including bracketed corpora. In this work, we attempt to use a bracketed (tagged) corpus for grammar inference. Towards the automatic acquisition of a context-free grammar, we proposed some statistical techniques to group brackets in a bracketed corpus (with lexical tags), according to their local contextual information. Two measures, divergence and Bayesian posterior probability, were introduced to express the similarity between two brackets. Merging the most similar bracket pair iteratively, a set of label groups was constructed. To terminate a merging process at appropriate timing, we proposed differential entropy as a measure to represent the entropy difference before and after merging two brackets and stopped the merging process at a large fluctuation. From the experimental results compared with the model solutions given by three human evaluators, we observed that divergence gave a better solution than Bayesian posterior probability. For divergence, we obtained 84 % recall and 67 % precision, and up to 90 % and 82 % when considering both positive and negative measures. We also investigated the fitness of using differential entropy for terminating the merging process by way of experiment and confirmed it.

In this paper, we focus on only rules with lexical categories as their right hand side. As a further work, we are on the way to introduce the techniques introduced here to acquire the other rules(rules with nonterminal categories as their right hand side). At that time, it is also necessary for us to develop some suitable evaluation techniques for assessing the obtained grammar.

References

[Aga95] Agarwal, R.: Evaluation of Semantic Clusters, in *Proceeding of 33th Annual Meeting of the ACL*, pp. 284–286, 1995.

[Bri92] Brill, E.: Automatically Acquiring Phrase Structure using Distributional Analysis, in *Proc. of Speech and Natural Language Workshop*, pp. 155–159, 1992.

[EDR94] EDR: Japan Electronic Dictionary Research Institute: *EDR Electric Dictionary User's Manual (in Japanese)*, 2.1 edition, 1994.

[Har51] Harris, Z.: *Structural Linguistics*, Chicago: University of Chicago Press, 1951.

[Hat93] Hatzivassiloglou, V. and K. R. McKeown: Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives according to Meaning, in *Proceeding of 31st Annual Meeting of the ACL*, pp. 172–182, 1993.

[Iwa95] Iwayama, M. and T. Tokunaga: Hierarchical Bayesian Clustering for Automatic Text Classification, in *IJCAI*, pp. 1322–1327, 1995.

[Kiy94a] Kiyono, M. and J. Tsujii: Combination of Symbolic and Statistical Approaches for Grammatical Knowledge Acquisition, in *Proc. of 4th Conference on Applied Natural Language Processing(ANLP'94)*, pp. 72–77, 1994.

[Kiy94b] Kiyono, M. and J. Tsujii: Hypothesis Selection in Grammar Acquisition, in *COLING-94*, pp. 837–841, 1994.

[Lar90] Lari, K. and S. Young: "The Estimation of Stochastic Context-free Grammars Using the Inside-Outside Algorithm", *Computer speech and languages*, Vol. 4, pp. 35–56, 1990.

[Mil94] Miller, S. and H. J. Fox: Automatic Grammar Acquisition, in *Proc. of the Human Language Technology Workshop*, pp. 268–271, 1994.

[Per92] Pereira, F. and Y. Schabes: Inside-Outside reestimation from partially bracketed corpora, in *Proceeding of 30th Annual Meeting of the ACL*, pp. 128–135, 1992.

[Per93] Pereira, F., N. Tishby, and L. Lee: Distributional Clustering of English Words, in *Proceeding of 31st Annual Meeting of the ACL*, pp. 183–190, 1993.

[Ris89] Rissanen, J.: *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing, 1989.

[Shi95] Shirai, K., T. Tokunaga, and H. Tanaka: Automatic Extraction of Japanese Grammar from a Bracketed Corpus, in *Natural Language Processing Pacific Rim Symposium(NLPRS'95)*, pp. 211–216, 1995.

[Swe69] Swets, J.: Effectiveness of Information Retrieval Methods, *American Documentation*, Vol. 20, pp. 72–89, 1969.

Appendix

labels	some instances
(ADJ)	'specific' 'commercial' 'adequate' 'structural' 'old'
(ADV)	'explicitly' 'enormously' 'quite' 'not'
(ART)	'the' 'a' 'an'
(AUX)	'may' 'should' 'did' 'could' 'will' 'have'
(BE)	'be' 'is' 'are'
(CONJ)	'and' 'when' 'or'
(DEMO)	'this' 'that' 'these' 'such'
(INDEF)	'few' 'one' 'any' 'some'
(NOUN)	'member' 'Japan' 'merchant' 'tour' 'area'
(NUM)	'2' '0.5' '60 billion'
(PREP)	'with' 'in' 'to' 'of'
(PRON)	'I' 'my' 'me' 'your' 'us'
(PTCL)	'up' 'to (to V)' 'down' 'out'
(UNIT)	'centimeter' 'percent' '%' 'mm' 'dollar'
(VI)	'grow' 'delay' 'feed' 'go' 'went' 'gone'
(VT)	'give' 'gave' 'given'