GRAMMAR INDUCTION BY MDL-BASED DISTRIBUTIONAL CLASSIFICATION

Yikun Guo
Dept. Of Computer Science
Fudan University
Shanghai, 200433, China
ykguo2000@yahoo.com

Fuliang Weng
Research & Tech Center
Robert Bosch Corp
Palo Alto, CA94304, USA
fuliang.weng@rtc.bosch.com

Lide Wu
Dept. Of Computer Science
Fudan University
Shanghai, 200433, China
ldwu@fudan.edu.cn

Abstract

In this paper, we introduce a novel MDL-based grammar learning algorithms, which can automatically induce a good amount of high quality parsing-oriented grammar rules from a tagged corpus with a minimal annotation. Comparing between the basic best-first MDL induction algorithm and a pseudo-grammar induction process, we identify problems associated with the current MDL-based grammar induction approaches. Based on this, we present a novel two-stage grammar induction algorithm to overcome a local-minimal problem by clustering the left hand sides of the induced grammar rules with a classifier trained through a seed grammar. Preliminary experimental results show that the resulting induction curve is very close to its upper bound and outperforms the traditional MDL-based grammar induction.

1. Introduction

With the increasing demand for natural language processing in the various Internet applications, such as automatic speech recognition and dialog systems, the acquisition of large amount of high quality grammar rules become more important. The availability of hand-annotated corpora, such as Penn Treebank Project, offers the possibilities for overcoming this knowledge-engineering bottleneck. However, the parsers based on such grammar rules have the risk of becoming too tailored to these labeled training data so as not be able to reliably process sentences from other domains. To parse sentences from a new domain, one would then try to obtain a new set of grammar rules from that domain, which often would require hand-parsed sentences for the new domain. Because to (semi-)manually parse a large corpus is both a labor-intensive and a time-consuming task, it would be beneficial to automatically derive grammar rules from raw text data from that domain with minimal annotations.

In this paper, we report our ongoing research work in automatic grammar acquisition within the *minimal description length* (MDL) [Ris78, Ris89] paradigm, together with contextual distribution classification to infer the LHS of those induced rules. Particularly, we want to address three MDL-related grammar induction issues: 1) What problems do current MDL-based grammar induction approaches have? 2) What MDL values may we obtain using the basic MDL induction approach when both the grammar rules and their application order is known? This way, we can have a good idea about the upper bound for the basic MDL-based grammar induction. 3) Are there any new approaches that can lead to a performance close to that upper bound?

To answer these questions, we conducted a set of experiments that compare the induction curves under different settings using both automatically induced grammar rules and hand-annotated rules in Treebank.

The results show that the MDL principle alone induces reasonable phrase grammar rules at the beginning, but quickly leads to a local-minimal and most of induced rules then are not adequate. However, applying the rules from Treebank by MDL principle in bottom-up order shows monotonous and sharp decrease of MDL values, compared with the results from the basic MDL principle. We speculate that it may be due to the vagueness of the LHS symbols from the MDL principle alone, and therefore, we improve our algorithm to determine LHS using distributional classification. The experimental results show that the new approach is very close to the hand-annotated rule induction approach in term of MDL values.

The rest of paper is organized as follows: section 2 presents the MDL principle, with an emphasis on *description length gain* (DLG), described in [Kit98] following classic information theory [Shannon49, Cover&Thomas91]. The next section presents two grammar induction strategies, i.e. the basic induction algorithm, which aims to find optimal grammar rules from the scratch with the guide of MDL principle alone, and a two-stage improved induction algorithm that first explores the context distribution of five syntactic categories from some "*seed grammar*". During the next induction stage, those induced grammar rules are dynamically classified as one of the five categories to avoid the deficiency of MDL-based search strategy. Section 4 reports experimental settings, results and algorithm evaluation. Section 5 reviews previous MDL-based research on grammar learning and gives our conclusions.

2. Grammar induction by MDL Principle

Grammar induction can be viewed as a process that searches for the best grammar in a predefined grammar (or hypothesis) space. If a set of permissible rules or rule formats, e.g., context-free grammar (CFG) rules, are given, it is widely adopted to use the Baum-Welch (or forward-backward) algorithm and its extension, the inside-outside algorithm [Baker79, Lari&Young90], to estimate the probabilistic parameters for these grammars. Essentially, there are two sub-tasks in obtaining a CFG rule. One is to decide what the right-hand terminals or non-terminals should be, and the other is to decide the left-hand symbol (LHS).

2.1. MDL Principle

Researchers have proposed various techniques and criteria to constrain the grammar space and to guide the search process. For example, [Hol75] used *genetic* algorithm and [KVG83] applied *stimulated annealing* algorithm to facilitate the search process. However, at the core of the search process, the goodness criterion for search is a critical issue, this is because it tells which grammar rule is better. Among those approaches, the *minimal description length* (MDL) principle, which is based on the classic and algorithmic information theory [Shannon49, Solo64, and Kol65], has received a wide attention.

For any given set of data, i.e. legal sentences, there are usually multiple theories (i.e. a set of grammar rules) that can account for the data, and we need to decide which one to select. An often used principle is the *Occam's razor* principle, which states that given a choice of the theories, the simplest is best. There are two aspects associated with the simplicity. One is that how simple is the theory describes the data, and the other is that how simple is the description of the theory itself. There is clearly a tradeoff between these two aspects. [Ris78] formalized this as follows: given some data D, we should pick that theory T which minimizes. That is:

$$L(T) + L(D|T) \qquad (1)$$

where L(T) is the number of bits needed to minimally encode the theory T, and L(D|T) is the number of bits needed to minimally encode the data D given the theory T.

From Shannon's information theory [Shannon49], we know that if we have a discrete set X of items with a probability distribution P(x), then to send a message identifying x∈ X, we need approximately L(x) = $-\log_2(P(x))$ bits. In other words

$P(x) = 2^{-L(x)}$ (2)

This enables us to interpret the MDL principle in Bayesian framework. From the equation it can easily be seen that minimizing L(T) + L(D|T) corresponds to maximizing P(T)*P(D|T) and hence P(T|D). This shows, theoretically, searching for the most likely theory for a given data in a Bayesian modeling framework is equivalent to searching for a model with the minimal description length.

It should be noted that the MDL principle enables us to assign prior probabilities to items in a meaningful way, even if we do not really have enough prior knowledge. We can do this through minimal length encoding for the items.

2.2. Description Length Gain

The application of MDL is independent of encoding scheme [Ris89]. To calculate the description length L(T) +L(D|T), what we need is an ideal encoding scheme, instead of a real compression program. This can be formulated in terms of token counts in the corpus as below for empirical calculation [Kit 98], following classic information theory [Shannon49, Cover&Thomas91]:

$$DL(X) = n\hat{H}(X) = -n\sum_{x\in V}\hat{p}(x)\log\hat{p}(x) = -\sum_{x\in V}c(x)\log\frac{c(x)}{|X|} \qquad (3)$$

where V is the set of distinct tokens (i.e. the vocabulary) in corpus X and c(x) is the count of x in X.

Accordingly, the *description length gain* of selecting the substring $x_i x_{i+1}\ldots x_j$ (denoted as $x_{i\ldots j}$, i < j) as possible RHS candidate of a grammar in a given corpus is defined as

$$DLG(\chi_{i\ldots j}\in X) = DL(X) - DL(X[r\rightarrow\chi_{x\ldots j}]\otimes\chi_{i\ldots j}) \qquad (4)$$

where $X[r\rightarrow\chi_{x\ldots j}]$ represents a resultant corpus from the operation of replacing all instances of $x_{i\ldots j}$ with r throughout X, and $\otimes$ denotes a string concatenation operation with a delimiter inserted between its two operands, the current corpus and newly learned phrase $x_{i\ldots j}$.

It is worth to note that we can choose the substring with maximum *DLG* value at each iteration without carrying out the real string substitution throughout the original corpus. The calculation is based on the token count change involved in the substitution to derive the new corpus. After finding the substring with maximum *DLG*, we replace the substring with a new string r in the original corpus.

Another problem is that we need to derive the count of x, for all possible sub-strings x in the corpus X, because during the induction process, it is necessary to consider all segments (i.e. all n-gram) in the corpus in order to select a set of good candidates. For one thing, MDL principle itself prefers short grammar rules over long grammar rules and long rules normally occur less frequent than short rules and hence less possible to become good grammar rules in the induction. In addition, it is too computationally expensive to consider each possible of these n-grams at every point in the search. Therefore, we use only bi-gram and tri-gram in the induction process. However, we will consider two cases: using bi-gram and tri-gram RHS with automatic MDL principle alone, and using the same number of RHS hand-annotated rules.

3. Learning Strategies

3.1. Basic Induction Algorithm

Accordingly, the best first learning algorithm using the goodness criterion is illustrated in figure 1. Given an utterance $U = t_0 t_1 \ldots t_n$ as input string of some linguistic token, e.g. part-of-speech tags, the unsupervised grammar induction looks for the substring with maximum description length decrease, i.e. maximum *DLG*, at each iteration and then replaces the n-gram (bi-gram or the tri-gram in our work) with a random symbol in the whole corpus, at the same time, output the learned rules in this iteration. It loops until description length value does not decrease, or *DLG* have a zero or negative value.

1. set $k = 0$ and extract all 2-gram and 3-gram in X_k with their counts;
2. for every n-gram ($n = 2,3$) in X_k, examine:
   (a) If no more x_{ij} ($2 \leq j\text{-}i \leq 3$) that DL ($X_k[r\text{->} x_{ij}]$) < DL(X_k), output the phrase and exit;
   (b) Else r_k = argmax ΔDL ($X_k[r\text{->} x_{ij}]$);
3. $X_{k+1} = X_k[r_k \text{->} x_{ij}]$, output the r_k, go to step 2;

Figure 1 Basic MDL Induction Algorithm

3.2. Pseudo-Grammar Learning by MDL Principle

One may see that the learning algorithm may not reach the real shortest description length, since it is a best first strategy that stops at local minima. To evaluate if the MDL principle is applicable for those hand-annotated rules in Treebank corpus, we implement a pseudo-grammar induction algorithm to gain insights. Figure 2 outlines this pseudo-grammar induction algorithm.

1. Extract all hand-annotated grammar rules from Treebank corpus, sort those rules according to bottom-up parsing order, mark grammar rules as “hidden”, except for leaf grammars in the every sentence tree and add them to the rule pool;
2. For all rules in the rule pool, apply step 2 and 3 of basic induction algorithm in figure 1 in each run to choose the rule with maximum *DLG*;
3. Output the learned grammar and apply it in order to involve more higher level grammars, that is , if all the children of one grammar are applicable, the grammar can be marked as “visible” and hence add to the rule pool;
4. Go to step2 if there are rules in the rule poor.

Figure 2 pseudo-grammar induction algorithm

In the algorithm, by simple extracting all hand-annotated rules from the corpus, the rule form (RHS and LHS) and the rule application order are predetermined based on the parse trees. We only use MDL principle to pick a rule in the current step so to get maximum description length gain. When all the child rules are selected and applied, their parent rule will be considered subsequently. Thus, we apply these hand-annotated rules roughly in the bottom-up parsing order, guided by the MDL criterion. From this experiment, we want to figure out how the basic induction algorithm differs from the pseudo-induction, where the rule and the order of the application are already known, under the same criterion. In addition, through this experiment, we try to find an upper bound for MDL-based grammar induction.

NP->DT JJ NN	NP->DT JJ NNS	NP->DT NN	NP->DT NNS	NP->PRP	NP->JJ NNS
NP->JJ NN	NP->NNP POS	NP->DT NN POS	NP->JJ JJ NN	NP->JJ JJ NNS	NP->DT NN NN
NP->DT CD NNS	NP->NP NP	NP->NP CC NP	ADJP->RB JJ	ADJP->RB JJR	ADJP->RB JJS
ADJP->RBS JJ	ADJP->RBR JJ	PP->IN NP	PP->TO NP	PP->IN S	VP->VB NP
VP->TO VP	VP->MD VP	VP->VBD NP	VP->VBZ NP	S->NP VP	S->PP NP VP

Table 1 the Seed Grammar Rules

3.3. MDL Induction by Dynamic Distributional Classification

Comparing the results of the two experiments, we discover that the basic MDL induction algorithm alone does infer reasonable phrasal grammar rules at the beginning, but after getting about a hundred of rules, it quickly reaches a local-minimum and most of the induced rules are not adequate.

We suspect that it may be due to the random labeling of LHS for those induced rules, because if all the LHS symbols of the induced rules are different, the repetition of certain patterns becomes less and therefore its MDL value decreases less dramatically.

Based on this observation, we come up with a new algorithm. We decide to integrate some linguistic information into the search strategy, which tries to classify the LHS symbols of the induced rules, using distributional analysis, and to help the search process to infer more syntactic plausible rules.

The algorithm is divided into two stages, one is the context vector training stage and the other is an improved MDL induction process.

The context vector training algorithm is based on the assumption that similar grammar rules tend to occur in similar contexts. The contexts of the rules from “VP” category, for instance, differ greatly from those of the “NP” rules. If the context is restricted to a fixed sliding window (e.g., three part-of-speech tags in our work, on either side of rules), then we can define the context distribution over all rules in that syntactic category. The context distribution of one category can be estimated from the observed contexts of sample sentences in category.

In the next MDL induction stage, we measure the similarity of each MDL induced rule to the center of context vectors of each syntactic category using Kullback-Leibler (KL) divergence as the distance function and assign to the LHS of the rule the category with the shortest distance. At each iteration, we also dynamically update the contexts and their centers of the induced rules for every syntactic category.

For simplicity, the syntactic categories are limited to five non-terminals, i.e., “NP”, “VP”, “S”, “ADJP”, and “PP”, which are main syntactic components in the syntactic parsing. We take, as “a *seed grammar*”, a set of most frequently used grammar rules for each of 5 syntactic categories and analyze the contexts (3 left and 3 right part-of-speech tags) for each of those rules in the sample corpus. The thirty seed rules we used in the algorithm are illustrated in Table 1. Note that we not only use the base grammar rules, i.e. the rules at the leaves of the parsing tree, but also the ones at the upper levels, since exploiting the context of these rules on the fly will make the search process more robust. Another critical issue concerning the selection of the seed grammar rules is to decide the number of rules for each category. The ratio we choose is roughly the same ratio for these five categories in the training corpus. In addition, we discover that “NP” rules alone account for two thirds of all rules in the hand-annotated Treebank corpus and they dominate other kind of rules especially in the bottom level of parsing trees. Therefore, we choose many “NP” rules as *seed grammar rules*.

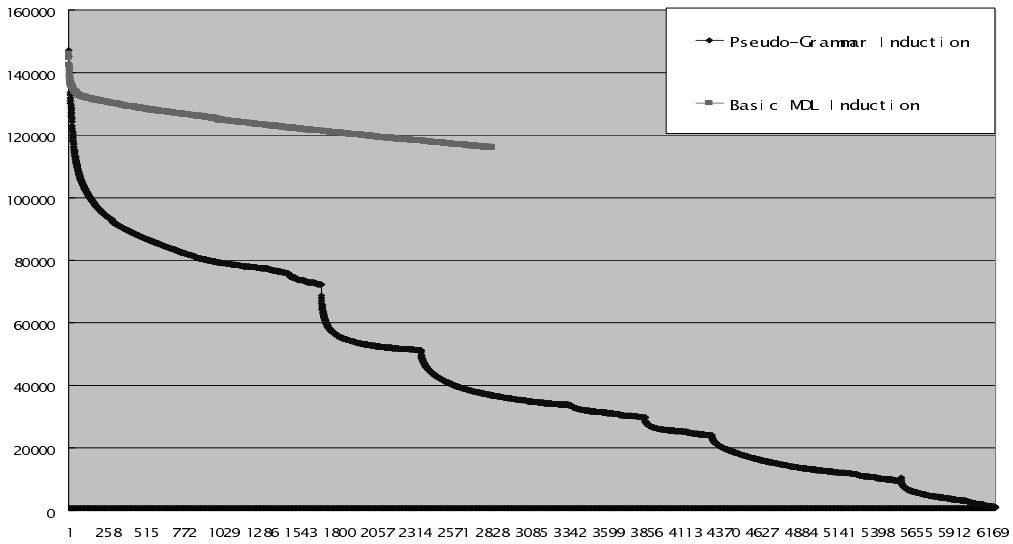

Figure3 The Two Grammar Induction Curves (MDL Value vs. Rules Induced)

4. Experiments and Results

A number of preliminary experiments on unsupervised phrase and lexical learning have been conducted on parts of Treebank corpus. These experiments show promising results by *DLG* measure [Kit 98]. It shows certain ability to capture the regularities in the data. Since it takes a learning-via-compression approach, i.e. MDL principle approach, the result is a set of deterministic CFG rules.

We perform four experiments and all of them use 2,500 sentences extracted from Treebank corpus with hand-annotated part-of-speech tag for each word as input. Backing off to POS tags is necessary because it alleviates the sparse data problem.

4.1. Experiment 1: Basic MDL Grammar Induction

The first experiment is the basic MDL principle induction. The testing corpus contains 2,500 short sentences and the vocabulary set is made up of 32 POS tags, a subset of 47 tags used in Treebank corpus. We apply MDL principle on the grammar space, where the RHS of a CFG is bi-gram or tri-gram. The first thirty of induced rules are given in appendix A.

From the appendix, we can see that most learned grammar rules are reasonable, such as [NNP NNP], [TO VB] and [MD VB]. However, some other rules seem to be quite "flat", i.e. lack of internal structures of the rule. The rule [PRP RB VBD], for instance, should be broken down into [PRP [RB VBD]]. In addition, we plot MDL value curve to show the MDL decrease trend along the search procedure. The curve is given in figure 3.

It is clear that having induced about a hundred rules, the basic MDL induction algorithm reaches the local minimal quickly. For comparison and verification of whether the MDL principle is useful for real world data, we perform another experiment using the pseudo-induction algorithm in subsection 3.2.

4.2. Experiment 2: Learning by Pseudo-Grammar Induction

In this experiment, we use the same search strategy but to the manually annotated rules found in Treebank corpus. The search process only chooses the rule with the maximum description length decrease, while the rule forms are all predetermined in advance. The MDL curve is illustrated in the figure 3. The detail of the algorithm is described in section 3.2.

From the figure, we observe that although the two curves are very closely to each other in the beginning, they differ greatly in the whole process in that: firstly, no local minimal is found in this case, while the basic induction process quickly becomes flat and has to be terminated because it is too computational expensive to infer new rules – large amount of randomly selected LHS of the induced rules lead to a very sparse distribution. Secondly, for the pseudo-grammar induction case, the curve decreases irregularly, that is, a few "critical" rules make the description length drop dramatically than the rest of them. We still work on understanding the effects.

To verify if it is just a special case for pseudo-grammar induction, we perform similar experiment on different sentence sets, illustrated in figure 4. The three curves are obtained by applying pseudo-grammar induction algorithm on different number of sentences extracted from Treebank corpus, namely 2,500, 5,000 and 10,000 sentences respectively. The chart shows the consistency among varied numbers of sentences.

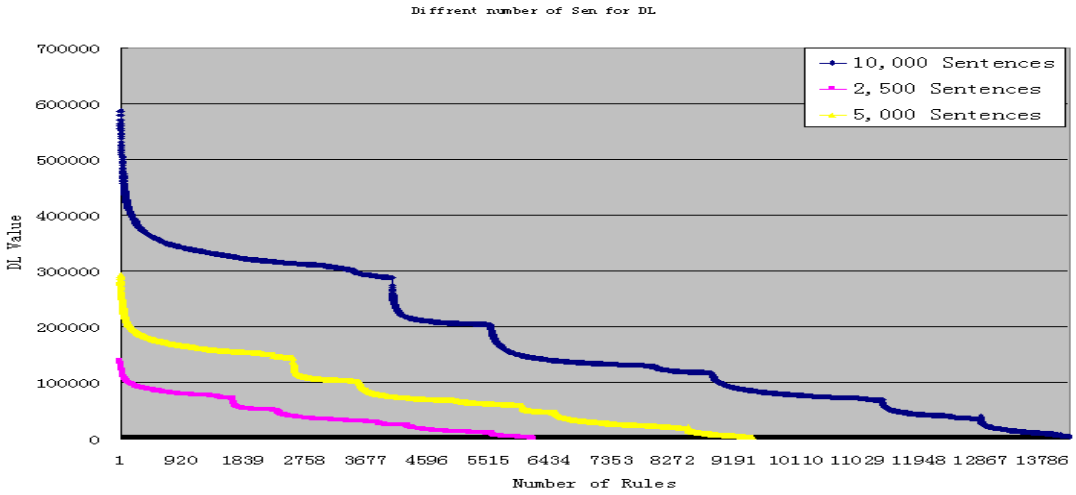

Figure 4 Different Data Set used in Pseudo-Grammar Induction Experiment

4.3. Experiment 3: MDL Induction by Dynamic Distributional Classification

In this section, we give experiment results of the two-stage induction algorithm described in section 3.3.

The goal of the experiment is to incorporate some linguistic knowledge into the search process to get more syntactic plausible grammar and overcome the local-minimal problem.

After training the classifier on 1,000 POS tagged sentences using the seed grammar rules, we obtain the context vector centers for all the 5 syntactic categories. Then, we construct the induction sets using a different set of 2,500 sentences (which is the same as the previous experiements).

The curve labeled with "MDL Induction by DDC" in Figure 5 summarizes the outcome of the experiment. From the chart, we see that it decreases monotonically, however, with no local minimal found at this time in the curve; the search process repeats until no rules can be induced by MDL framework. In addition, the curve is very close to the pseudo-grammar induction case, i.e. the upper bound of the algorithm.

This algorithm learns not only the right-hand sequences of terminals/non-terminals, but also their LHS syntactic categories.

We also conduct another experiment to see the effect when both the classification of syntactic category and the MDL induction are accurate, that is, assume the syntactic category of every induced rule is correctly identifiable using extra knowledge, and also assume all induced rules are subset of the grammar rules found in the hand-annotated Treebank corpus.

We vary the experiment settings in this algorithm to explore the upper bound for experiment 3. Because many induced grammar rules, which reduce the description length dramatically, are not syntactic plausible rules and are not found in manual rule set, however, the algorithm assigns a syntactic category for them and updates the center of the context vector for that category in the search process. This, in turn, impacts on the classification in the later part of the process. Although we try to use several high-level grammar rules as *seed grammar* and explore their context in the induction process to compensate this effect, how to improve the robustness needs further investigation.

Another major factor to certain poor performance is the limited number of syntactic category, (five in our work, but more than fifteen in the Treebank), and the restricted number of n-gram (many 'flat' grammar rules are found in Treebank [Gai95]) that we applied in the experiment all impact the induction procedure. To

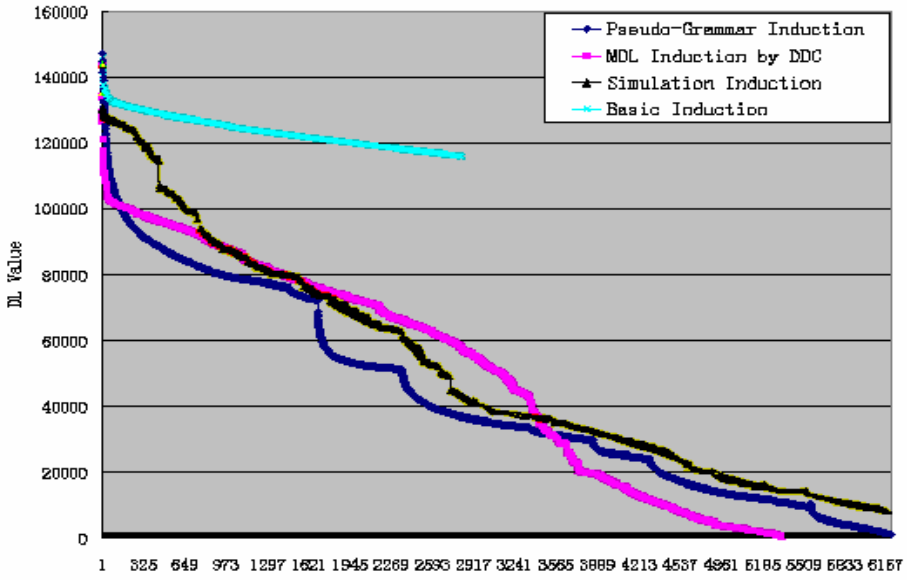

Figure 5 The MDL Curve

investigate their impact, we loose the restriction on the limited number of LHS categories. In the each induction process, as described in section 2.3, we sort the candidate rules by their *DLGs* in a descending order, choose only the first rule found in hand-annotated Treebank grammar and apply it by replacing its RHS sequence with the correct LHS.

In such procedure, no classification is performed, therefore the classifier is always assumed to be accurate, which removes the effect that grammar rules induced early influence the rules induced later. On the other hand, all rules learned are syntactic plausible ones, since they are subset of the manual-annotated grammar rules. The only difference between them is the number of the right-hand symbol and the syntactic category of the left-hand symbol.

The experiment result is illustrated in the figure 5 with the label, "Simulated induction". From the figure, we find that the different number of n-grams and the syntactic categories does affect the results, especially in

the later search process, when compared with the "Pseudo-Grammar Induction" curve. On the other hand, compared with "Grammar Induction by DDC" curve, the classifier is really distracted by the previously induced "bad" grammar rules. This is the place where future research work can be directed.

In addition, we also calculate the precision and recall for those induced rules, in contrast with the hand-annotated grammar rules extracted from the same set of 2,500 sentences. The result is illustrated in table2.

Rules after	100	200	500	1000
Precision	0.92	0.89	0.82	0.74
Recall	0.13	0.16	0.18	0.22

Table2 the precision and recall for induced rules

5. Related Work and Conclusions

The difficulty of grammar induction depends greatly on the amount of supervision provided. [Charniak96], for example, has shown that a grammar rule can be easily constructed when the examples are fully labeled parse tree. However, if the examples consist of only raw sentences with no extra structural information, grammar induction is very difficult, even theoretically impossible [Gold67]. Part of our work explores the in-between case, where the category of learned rules could be decided by the result of a supervised learning algorithm.

Second, the search criterion also impacts the induction process. Besides the MDL principle, there are other search criteria, similar to us, to guide the "guessing game". Cook et al. [Cra76] explores a hill-climbing search for a grammar of a smaller weighted sum of grammar *complexity* and the *discrepancy* between the grammar and corpus; Brill et al. [BMMS90] derive phrase structures from a bracket corpus by *generalized mutual information* approach; and Brill and Marcus [BM92] attempt to induce binary branching phrases with distribution analysis using the information-theoretical measure *divergence*, derived from *relative entropy*. de Marcken gives an in-depth discussion on the kind of issues involved in the pure distribution analysis and on the disadvantages of the inside-outside algorithm for grammar induction in his PhD thesis [deMa95]. Recently, following Cook's work, Stolcke [Sto94] worked under the Bayesian modeling framework, whereas Chen [Chen95, Chen 96] uses the universal prior probability $p(G) = 2^{-l(G)}$ for grammar induction. Their learning strategy reports to work well on small to medium size artificial corpora, using measures such as entropy, perplexity or likelihood. But, to our knowledge, no one has tried to induce all levels of syntactic grammar rules on large scale real corpora before.

In the paper, we have shown two MDL-based grammar induction algorithms. Both of them try to infer syntactic plausible grammar rules for parsing with one focusing on a best-first search strategy with minimal supervision and the other focusing on integration of language constrains into the learning model. Comparing these two approaches through experiments, we show that MDL principle alone could induce phrase-level grammar well, but fails to learn high-level grammar rules. In addition, with integrated language constraints, the MDL principle could infer not only the grammar rules, but also the categories of the LHS of the learned rules. The experiments show that the result of the second algorithm is very close to that of the pseudo-grammar induction algorithm.

To further improve the grammar learning algorithms for high performance parsing, we still need to investigate the failed instances and come up with more sophisticated learning algorithms. Evaluating learned rules for parsing and further improving learning algorithms are the two main tasks in our future work.

Reference

[Baker79] Baker, James K. "*Trainable grammars for speech recognition*". In Speech Communication Papers for the 97th Meeting of the Acoustical Society of America, edited by Jared J. Wolf and Dennis H. Klatt, 547--550, MIT, Cambridge, Mass, 1979

[BM92] Brill, E. and Marcus, M. *Automatically Acquiring Phrase Structure Using Distributional Analysis*, In proceedings of 1992 DARPA Speech and Language Workshop, Harriman, N.Y., 1992

[BMMS90] E. Brill, D. Magerman, M. Marcus, and B. Santorini. *Deducing Linguistic Structure from the Statistics of Large Corpora*. In Proceedings of DARPA Speech and Natural Language Workshop, Hidden Valley, Pennsylvania, June 1990

[Charniak96] Charniak, E. *Tree-bank grammars*. In Proceedings of the National Conference on Artificial Intelligence (AAAI), 1996

[Chen95] S.F. Chen. *Bayesian grammar induction for language modeling*. Technical Report TR-01-95, Harvard University, Center for Research in Computing Technology, January 1995.

[Chen96] S. F. Chen. *Building probabilistic models for Natural Language*. PhD thesis, Harvard University, Cambridge, Massachusetts, Cambridge, Mass., 1996

[Cover&Thomas91] Cover, T. and Thomas, J. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991

[Cra76] Craig M. Cook, Azriel Rosenfeld, and Alan R. Aronson. *Grammatical inference by hill climbing*. Information Sciences, 10:59—80, 1976

[deMa95] C. de Marken. *The Unsupervised Acquisition of a Lexicon from Continuous Speech*. Technical Report A.I. Memo No. 1558, AI Lab., MIT, Cambridge, MA, November, 1995

[Gai95] Robert Gaizauskas. *Investigations into the grammar underlying the Penn Treebank II*. Research Memorandum CS-95-25, University of Sheffield, 1995

[Gold67] E. M. Gold. *Language identification in the limit*. Information and Control, 10:447--474, 1967

[Hol75] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975

[KVG83] Kirkpatrick, S., Gelatt, C. and Vecchi, M., "*Optimisation by Simulated Annealing*," Science, No. 220, pp.671-680, 1983

[Kit98] Kit, C. *A goodness measure for phrase learning via compression with the MDL principle*. In The ESSLLI-98 Student Session, Chapter 13, pp.175-187. Aug. 17-28, Saarbruken, 1998

[Kol65**]** Kolmogorov, A. N. Three approaches to the definition of the concept "*quantity of information*." Problemy Peredachi Informatsii 1, 3—11, 1965

[Lari&Young90] K. Lari and S. J. Young. *The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm*. Computer Speech and Language, 4:35--56, 1990

[Ris78] Rissanen, J. *Modeling by shortest data description*. Automatica, 14:465--471, 1978.

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

[Shannon49] C. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.

[Solo64] R. J. Solomonoff, "*A formal theory of inductive inference*," Inform. and Control, vol. 7, pp. 1--22, March 1964; pp. 224--254, June 1964.

[Sto94] A. Stolcke. *Bayesian Learning of Probabilistic Language Models*. Dissertation, U. California, Berkeley, 1994.

Appendix A

The first 30 grammar rules learned by basic MDL grammar induction algorithm from 2,500 sentences in the Treebank corpus are given below. The rules are marked as true (t), false (f) and unsure (u) respectively according to human evaluators. There are four columns in the table, namely, rule number, the current description length with model and data combined when the grammar rule is acquired, the rule and the evaluation flag. The POS tags in these rules are listed below.

1	146186	NNP NNP	t	16	136398	NNP NNPS	t
2	144823	TO VB	t	17	136274	TO CD CD	t
3	142641	MD VB	t	18	136128	NNP POS	t
4	141748	MD RB VB	t	19	135916	EX VBZ	u
5	141411	DT JJ NN	t	20	135839	WDT [MD VB]	t
6	140245	IN DT NN	t	21	135762	WDT VBD	t
7	138981	PRP VBP	u	22	135666	PRP RB VBD	u
8	138625	PRP VBD	u	23	135569	PRP [MD RB VB]	t
9	138119	PRP VBZ	u	24	135508	EX VBZ	u
10	137703	PRP [MD VB]	t	25	135456	WP VBZ	t
11	137471	NNS VBP	t	26	135369	JJR IN CD	f
12	136989	NNS WDT VBP	f	27	135271	TO DT NN	t
13	136834	WDT VBZ	t	28	135082	PRP RB VBP	u
14	136681	NNS WP VBP	f	29	135023	NNS RB VBP	f
15	136567	RB VB	t	30	134936	[NNP NNP] POS	t

CD: Cardinal number
DT: Determiner
JJ: Adjective
JJS: Adjective, superlative
IN: Preposition or subordinating conjunction
POS: Possessive ending
VB: Verb, base form
NNP: Noun, singular form
PRP: Personal pronoun
MD: Modal
RB: Adverb
To: ‘to’
VBD: Verb, past tense
VBN: Verb, past participle
VBZ: Verb, 3rd person singular present
NN: Noun, base form