社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－NL－159　(23)
2004／1／14

Refining algorithm of pattern rule set for detecting noun phrases in English

Ekaterina Goriouchkina † and Akira Adachi ††

This paper presents a way of increasing accuracy of detecting English noun phrases with pattern matching. The present approach is based on a refining algorithm of pattern rules extracted from the Penn Treebank3 corpus. The refined rule set serves as identification of noun phrases in English sentence. The method of identifying noun phrases is explained. The algorithm to obtain the best set of pattern rules is introduced for two steps to refine pattern rule set. Some experiments are performed to evaluate the effectiveness of detecting noun phrases. Careful consideration is given to ineffective pattern. Results of future experiments are estimated.

Keywords: *refining algorithm, noun phrase detection, automaton, corpus*

1. Introduction

Noun Phrase is a group of words in a sentence that behaves in the same way as a noun, that is as a subject, an object of a preposition. Detection of noun phrase is an essential procedure in syntax analysis. This procedure is a significant factor to determine the accuracy of syntactic analysis.

A lot of research activities have been done based on stochastic approach. Presented Super-tagger[1)] extracts probabilistic syntax rules from the Penn Treebank3 corpus. This research emphasizes syntactic structure in noun phrases. The present research needs learning stochastic approach as a base, because a number of syntax rules for noun phrases explosively increases. Additional problems are verbs or adverbs, which can be included in noun phrase. In this case, identification of noun phrase is getting even more difficult.

Chunking algorithm[2)] is also a relation approach. Chunking algorithm of noun phrase is based on Support Vector Machines (SVMs). SVMs can carry out the training in high-dimensional space with smaller computational cost, that is independent of their dimensionality.

This paper deals with a refining algorithm for the pattern rule set extracted from a tagged corpus. Pattern rules have been extracted from sentences that is a training corpus. Used corpus was hand-corrected. This is why, corpus involves ineffective patterns for successful detecting noun phases. Such ineffective patterns have to be removed from the pattern rule set. The refining algorithm removes ineffective rules from noun phrase sets of the training corpuses, which include from 10,000 to 140,000 noun phrases. As a result, the accuracy of parsing reaches 74.3% in a evaluation corpus after applying refined noun phrase pattern rule sets.

This paper presents first of all, brief introduction of the Penn Treebank3 corpus, which was used for creation training and evaluation corpora. Secondly, a noun phrase detection system for the evaluation is described. Then, the refining algorithm for a rule set extracted from the training corpus is proposed and the way of performing experiment is presented. Finally, evaluation results and consideration are discussed.

2. Used corpus

One of the important developments in linguistic over past 20 years has been the possibility to use large corpora (corpus) of language to study lexicon, semantics, structure, and discourse. With the advance of computers, it is now possible to use personal computers to analyze millions of words of language data. This research has used the Penn Treebank3's tagged data, which is a large annotated corpus of English[3)]. The corpus consists of over 4.5 million words of American English,which

† Department of Information Sciences Toho University
†† Division of Media Solution, Fujitsu Co. Limitted

means a collection about 70,000 sentences and 350,000 noun phrases. This corpus has been annotated for Part of Speech information and serving as an important research tool for investigators in natural language processing, speech recognition, and integrated spoken language system, as well as in theoretical linguistic. Example of tagged sentence:

```
[ Efforts/NNS ]
by/IN
[ the/DT Hong/NNP Kong/NNP Futures/NNPS Exchange/NNP ]
to/TO introduce/VB
[ a/DT new/JJ interest-rate/NN futures/NNS contract/NN ]
continue/VBP to/TO hit/VB
[ snags/NNS ]
,/, despite/IN

[ the/DT support/NN ]
[ the/DT proposed/VBN instrument/NN ]
enjoys/VBZ in/IN
[ the/DT colony/NN 's/POS financial/JJ community/NN ]
./.
```

Every word and punctuation mark are followed by tag of Part of Speech. More than 36 tags have been used. Noun phrases are shown in brackets [ ] .

The Penn Treebank3 corpus has been used to create the training corpus and the evaluation corpus for present experiments. Volume of the training corpus is from 10,000 noun phrases to 140,000 noun phrases. Volume of the evaluation corpus is about 100,000 noun phrases from about 20,000 sentences. Sentences for the training corpus were taken from beginning of the Penn Treebank3 corpus, which means files Penn Treebank3/tagged/pos/wsj 00 - 17 were used. Sentences for the evaluation corpus were taken from ending of the Penn Treebank3 - files Penn Treebank3/tagged/pos/wsj 18 - 24.

3. Method of identifying noun phrases

In present experiment, finding noun phrases have been done by search for the longest pattern. This search has requested usage of acyclic deterministic finite automaton[4]. Structure of deterministic finite automaton is this: each node is knotted by arcs, where every arc is labeled Part of Speech from a node to another node. Figure 1 shows the structure of deterministic finite automaton.

Information of input data has been compared with the internal structure.

For example, input string [CD-NNP-POS-NN] into automaton from Figure 1, where "-" is a parting line between every Part of Speech. Automaton reacted positively to [CD-NNP-POS], until state [NN] was appeared. The arc carrying label "NN" does not exist. This is why

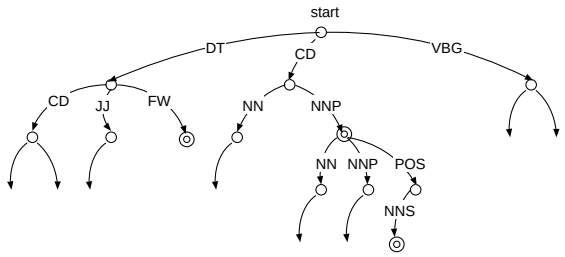

Fig.1 Example of deterministic finite automaton

transition to the next state was the failure. The last transition is finding final state, but, in this case, final state does not exist either. Coming back to the previous state means transferring to state [CD-NNP]. Happening to be in state [CD-NNP] of internal structure signifies that final state exists. For this case, [CD-NNP] is the longest pattern for given noun phrase.

4. Refining algorithm

A huge amount of data makes difficult to check the effectiveness of a pattern rule set. The pattern rule set is the Part of Speeches multitude, which was extracted from the training corpus. Unfortunately, the pattern rule set includes some patterns that decrease the result. Ineffective patterns must be excepted from pattern rule set. The proposed algorithm(Figure 2) can examine the effectiveness of every pattern rule from the set one by one and make a decision about the necessity of a given pattern for the result. An examination can be done by masking one pattern from the pattern rules set and applying left pattern set to the training corpus. The result of treatment shows the effectiveness of each pattern rule.

Training data contains n of noun phrases.

$P = r_1, r_2, ..., r_i, ..., r_n$

is pattern rule set, where n is the number of pattern rules. Each pattern rule r_i has to be removed and an experiment has to be done with remained $n-1$ rule set. The experiment is reiterated n times. Score of accuracy Q_{Mask_i} of covering training corpus's sentences by rules of left pattern set will show the necessity of removed pattern. If the score of accuracy Q_{Mask_i} is less, than before, when the pattern was in the set, it is necessary to return a previously removed pattern r_i. It means that the previously removed pattern is effective. If the score of accuracy Q_{Mask_i} after removing one pattern from pattern rule set is getting up, the removed pattern is the "disturbance pattern" and must be excepted from a pattern rule set.

In a pattern rule set, extracted patterns are following each other in strictly decided order, which is defined by

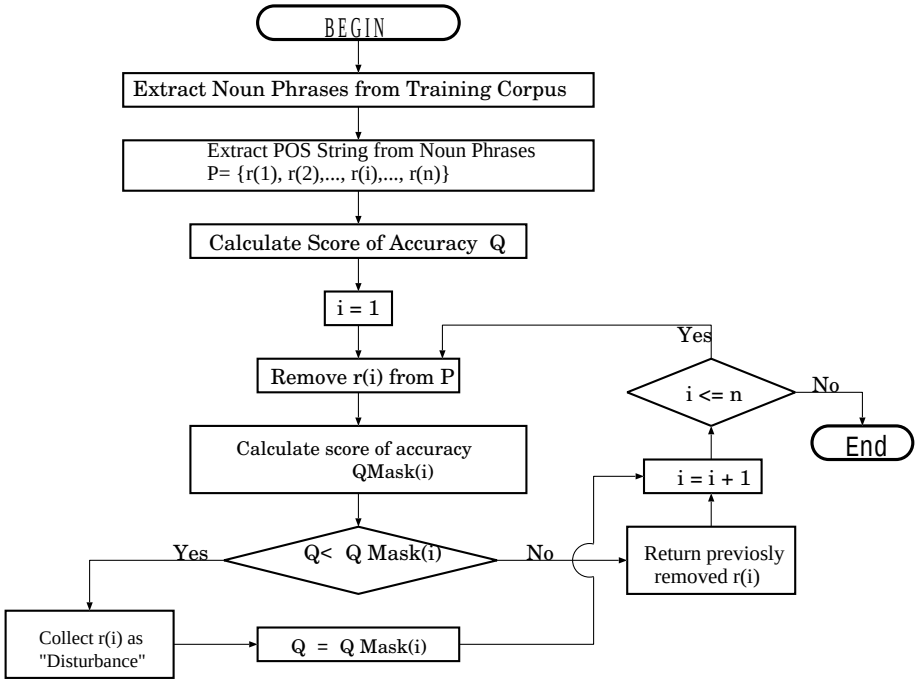

Fig.2 Block - chart of refining algorithm for "forward scan"

following existence of noun phrases in sentences of the Penn Treebank3 corpus. The following order may have some dependency between patterns. This factor is very important for the final result.

This is why the refining algorithm must be applied to pattern rule set twice. The first time is for checking patterns from the first pattern to the last pattern and the second time is for checking patterns from the last pattern to the first pattern. The first time of applying the refining algorithm is called forward scan. The second time of using the refining algorithm is called backward scan. In case of backward scan, checking patten rule set starts from the last pattern and goes to the first pattern. This way of applying the refining algorithm to pattern set can improve it until disturbances patterns will be gone. Combining usage forward scan and backward scan makes the refining algorithm very effective for detecting disturbances patterns.

The proposed refining algorithm considers the fact that pattern rules might have some mutual dependency, which influences the result. Therefore, the refining algorithm is checking pattern rule set of noun phrases not only from the beginning to the end, but also in opposite order, from the end to the beginning.

5. Experiments

Outline of the experiment is shown in Figure 3. Set of noun phrases is extracted from the training corpus, than the refining algorithm is applied to previously extracted set of noun phrases to except disturbances patterns and pattern rule set is done. The last step is getting result of detecting noun phrases of the evaluation corpus by present pattern rule set. The last step was done for all 14 already refined pattern rule sets.

Experiments have been done separately with the training corpus and the evaluation corpus. The first series of experiments have given refined rule sets for the training corpus, the second series of experiments have given the accuracy for the evaluation corpus. A size of the training corpus was not fixed, because a volume of pattern rule set for the best covering corpus was unknown. Starting size of the training corpus was 10,000 noun phrases. Every time the size of the training corpus was expanded by adding next 10,000 noun phrases. The final experiment was done with 140,000 noun phrases. Each time set of noun phrases was extracted from the training corpus. The refining algorithm was used for each set and ineffective patterns have been excepted from all pattern rules sets. Figure 4 shows the result of applying the refining algorithm to set of noun phrases from the training corpus of 10,000noun phrases. Figure 4 has a horizontal axis, that is the number of noun phrases in the pattern rule set and a vertical axis, which is accuracy (%).

The result was calculated for every size of the training corpus. Figure 5 shows results. A horizontal axis of the figure is size of the training corpus, which is the number of noun phrases, and a vertical axis is accuracy (%). The accuracy is a stable covering the size of the training

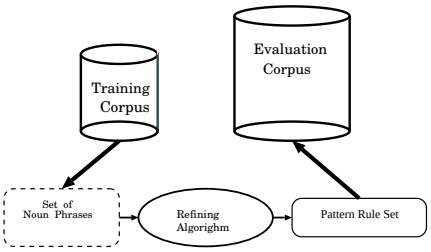

Fig.3 The way of performing experiment

corpus.

After getting results of experiments with the training corpus, refined pattern rule sets have been applied to the evaluation corpus. The evaluation corpus has fixed size. It is about 100,000 noun phrases from about 20,000 sentences. Every pattern rule set was created for each size of the training corpus. This is why every pattern rule set has different quantity of patterns. It means, that every pattern rule set has different ability to identify noun phrases in a sentence. The second series of experiments have shown results for applying 14 already refined pattern rule sets to the evaluation corpus. Figure 6 shows those results. Figure 6 is the graph with a horizontal axis, which indicates size of the training corpus, and a vertical axis, which is accuracy (%).

6. Evaluation

Very important parameters for evaluation are accuracy of treatment and volume of training data.

Experiments showed, that increasing the size of the training corpus can raise the accuracy of treatment of large evaluation corpus. After applying refined pattern rule set from the training corpus of 30,000 noun phrases to the evaluation corpus, the accuracy was over 70%. After applying the training corpus of 50,000 noun phrases and more, accuracy was linearly increasing. After applying refined pattern rule set extracted from the largest training corpus of 140,000 noun phrases to the evaluation corpus, the accuracy was 74.3%. In case of continuance of experiment with the training corpus of over 140,000 noun phrases, the result of accuracy for the evaluation corpus will be incresed.

7. Consideration

The constituents of the English noun phrase can be analyzed into functional constituents and formal constituents. From a function point of view, the noun phrase has four major components occurring in a fixed order:

- the determination, that is consituent, which determines the reference of the noun phrase in its linguistic or situational context;
- premodification, which comprises all the modifying or describing constituents before the head;
- the head, aroud which the other constituents cluster;
- postmodification, which comprise all the modifying constituents placed after the head.

Unfortunately, some patterns of noun phrases from the Penn Treebank3 corpus do not satisfy those definitions. Those kind of patterns are called disturbances patterns. A disturbance pattern is a pattern, that decreases percent of accuracy by existing in pattern rule set. During experiments some disturbances patterns have been discovered. All patterns

- with preposition in the beginning, in the middle or in the end;
- with verb in the beginning, in the middle or in the end of noun phrase;
- a lot of patterns with adjective in the end;
- all patterns which include adverbs in the beginning, in the middle or in the end of noun phrase

basically decrease accuracy.

Patterns, which include same punctuation mark and brackets ( ), have decreased result of treatment. Therefore, those patterns were also excepted from pattern rule set by the refining algorithm.

The effectiveness of every pattern in each training corpus has been checked by the refining algorithm. Figure 7 shows the number of excepted patterns and left patterns in pattern rule sets for every size of the training corpus. A horizontal axis of Figure 7 is size of the training corpus and a vertical axis is the number of pattern rules.

Although the refining algotithm extracts a pattern rule set which can identify noun phrases with high accuracy, the accuracy linearly increases with increased size of the training corpus. This problem needs an abstraction of the refined pattern rule set, as a solution. In this experiment , an acyclic finite automaton has been used to detect a noun phrase. The abstraction of the pattern rule set requires an abstraction method and a cyclic finite automaton[5]. This problem is remained to be solved in the future.

8. Conclusions

This paper presented the approach to detect noun phrase in English sentence, based on using the refining algorithm. Rising accuracy of the treatment has been succeeded by using the refining algorithm to detect and except patterns, which were ineffective. 140,000 noun

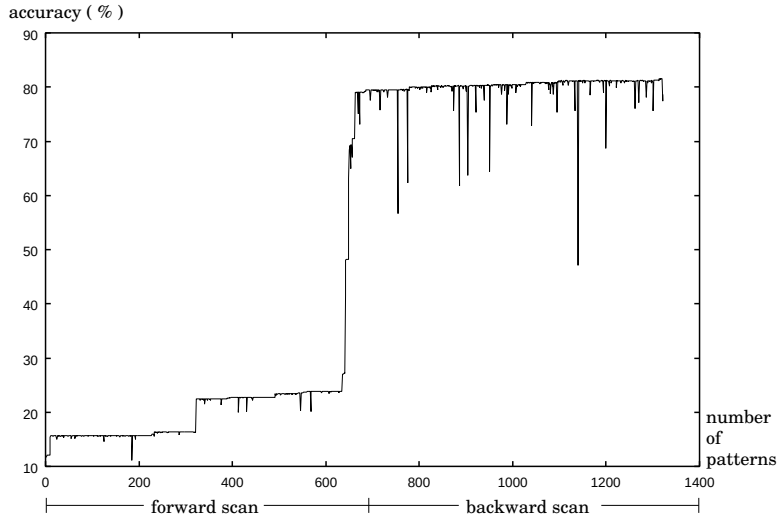

Fig.4 Result of applying the refining algorithm to pattern rule set from the training corpus of 10,000 noun phrases

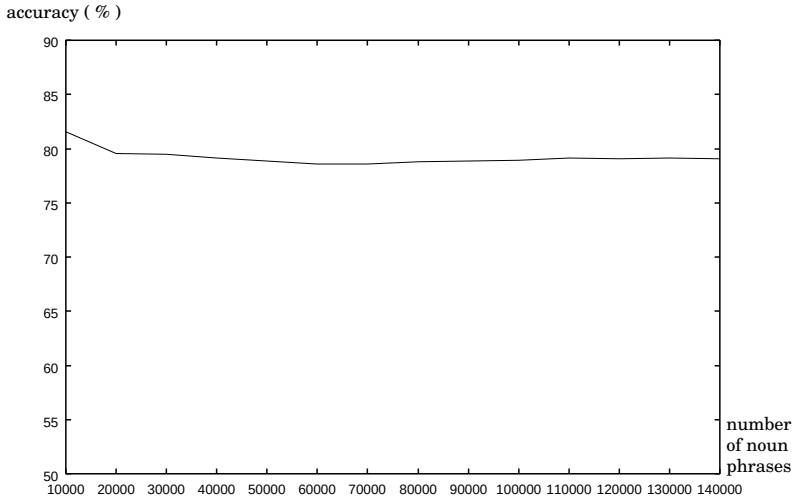

Fig.5 Result of experiments with the training corpus

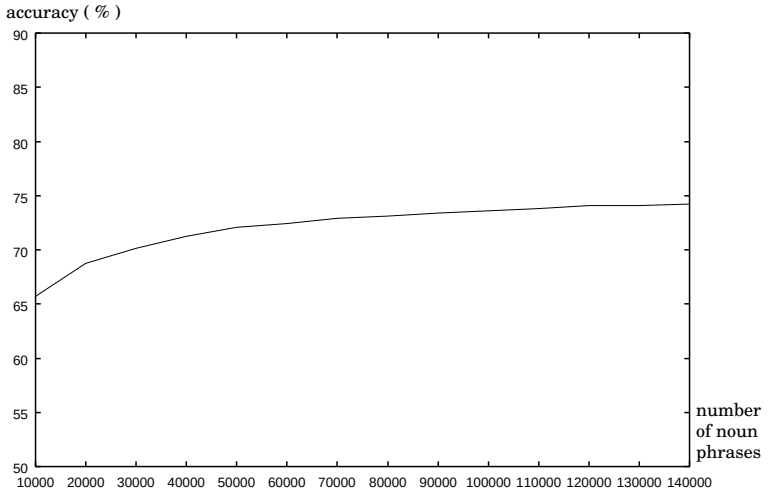

Fig.6 Accuracy of detecting noun phrases of the evaluation corpus by refined pattern rule sets

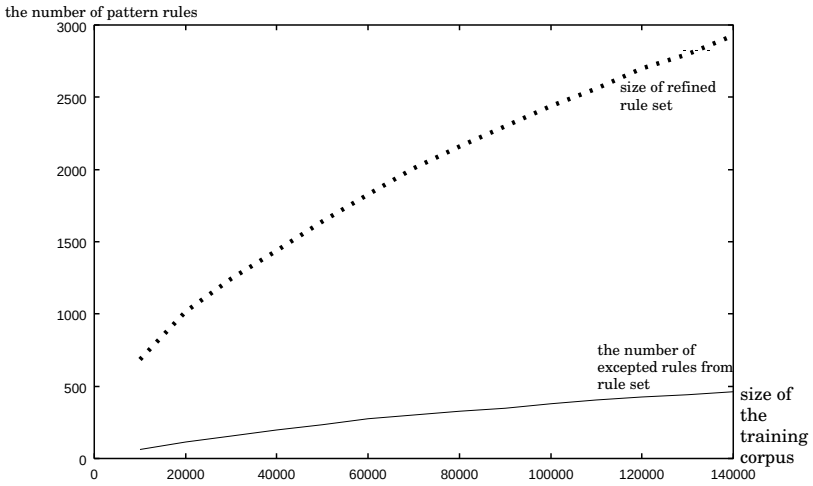

Fig.7 Size of refined pattern rule sets and the number of excepted patterns

phrases have been trained to creat refined pattern rule sets for achievement of high accuracy. The proposed approach has taken about 100,000 noun phrases from 20,000 sentences of the evaluation corpus for performing series of experiments to prove the effect of the present approach. The accuracy of treatment was rising to 74.3%. The present research can be continued with expanded size of the training corpus to increase the accuracy of the treatment.

References

1) Aravind K.Joshi and Srinivas Bangalore. Supertagging: An Approach to Almost Parsing. Computational Linguistics, Volume 25, Number 2, pages 237-264, June 1999.
2) Taku Kudoh and Yuji Matsumoto. Chunking with Support Vector Machines. Information Processing of Japan,NL 140, pages 9-16, 2000.
3) Beatrice Santorini. Part-of-Speech Tagging Guidelines for the Penn Treebank Project.(3rd revision, 2nd printing), Linguistic Data Consortium,June 1990.
4) Jan Daciuk, Stoyan Mihov, Bruce W.Watson and Richard E.Watson. Incremental Construction of Minimal Acyclic Finite-State Automata. Computational Linguistics, Volume 26, Number 1, pages 3-16, March 2000.
5) Rafael C.Carrasco and Mikel L.Forcada. Incremental Construction and Maintenance of Minimal Finite-State Automata. Computational Linguistics, Volume 28, Number 2, pages 206-217, June 2002.