AN EFFICIENT SYNTACTIC TAGGING TOOL FOR CORPORA[1]

Ming Zhou   Changning Huang

Dept. of Computer Science, Tsinghua University,
Beijing, 100084, China.

ABSTRACT

The tree bank is an important resources for MT and linguistics researches, but it requires that large number of sentences be annotated with syntactic information. It is time consuming and troublesome, and difficult to keep consistency, if annotation is done manually. In this paper, we presented a new technique for the semi–automatic tagging of Chinese text. The system takes as input Chinese text, and outputs the syntactically tagged sentence(dependency tree). We use dependency grammar and employ a stack based shift / reduce context–dependent parser as the tagging mechanism. The system works in human–machine cooperative way, in which the machine can acquire tagging rules from human intervention. The automation level can be improved step by step by accumulating rules during annotation. In addition, good consistency of tagging is guaranteed.

KEYWORDS: syntactic tagging, tree bank

1. INTRODUCTION

In recent years, the corpora, either monolingual or bilingual,plays an important role in MT and linguistics researches(Komatsu, jin & Yasuhara, 1993; Sato, 1993; Isabelle & Dymetman,1993). This is because the corpora with large amount of running text is considered as an ideal resources of linguistic knowledge. However, to acquire knowledge from the corpora(Watenabe, 1993; Mitamura, Nyberg, Carbonell, 1993), or effectively use the sentences as examples, as in example based approach(Nagao, 1984, O. Furuse & H.Iida, 1992), the corpora has to be annotated with certain information which may be of morphological information, syntactic information and semantic information.

Take Chinese monolingual corpora, for instance, the raw corpora, i.e. the text which has not been segmented into word strings, can only be used for statistics of Chinese character, however, if you want to work out the frequency of words, the corpora has to be segmented into word strings, i.e., it has to be annotated with word boundary information. Further more, if you want to obtain the co–occurrence frequency of each two adjacent part of speeches, which is helpful to the study of part of speech (POS) tagging, you must annotate the corpora with POS information. And if you want to extract the syntactic knowledge from corpus, the corpus must be attached with syntactic information such as dependency relation and phrase structure etc., and such a corpora is called tree bank which is used as the resources for knowledge acquisition and examples in EBMT research.

There are usually five levels of annotation for a corpora, which includes word boundary tagging, POS tagging, sense tagging, syntactic relation tagging and semantic relation tagging, with the depth of tagging increases. To improve the tagging automation and keep good consistency, a mechanism is required at each level of tagging to acquire knowledge from human intervention and the annotated corpus. The knowledge acquired should be fed back to the tagging model to improve the tagging automation and correctness.

Our group has been doing the research on Chinese corpus for many years, and has done successful experiments on word boundary tagging, POS tagging(Bai & Xia, 1992), sense tagging(Tong, Huang & Guo, 1993). The syntactic relation tagging, however, has not been resolved well because of some reasons. First, there is no clear answer about which grammar formalism, such as phrase structure grammar, or dependency grammar or any other grammar is suitable for large scale running text syntactic tagging? Second, how to save human's labor from tagging, and keep good

[1] supported by National Foundation of Natural Science of China.

consistency?

For the first question, some researchers adopt phrase structure grammar (PSG) as the tagging formalisms(Leech & Garside 1991), and some adopt dependency grammar(DG) 1993, Komatsu, Jin, & Yasuhara, 1993). In comparison with PSG, the authors think, DG has some advantages. First, it is economical and convenient to use DG for the syntactic relation tagging of corpus because there is no non–terminal node in the parse tree of DG; Second, DG stresses relations among individual words, the acquisition of collocation knowledge and syntactic relation among words is straight; Third, there is relatively straight map between dependency tree and case representation.

Based on the above discussion, the authors chosen dependency grammar as the syntactic formalism for corpora, and defined 44 kinds of dependency relation for Chinese(Zhou & Huang 1993).

For the second question, we must develop an efficient tagging tool, for which we need take account of two factors: (1) the power of acquiring tagging knowledge from the human intervention, in order to improve the automation level; (2) the ability of keeping good consistency.

Simmons & Yu (1992) introduced the context–dependent grammar for English parsing, in which the context–dependent rules can be acquired through an interactive mechanism, the phrase structure analysis and case analysis were conducted through a stack based shift / shift parser, with success ratio reached as high as 99%. Inspired by their work, we designed a dependency relation tagging tool for Chinese corpus, called CSTT. CSTT takes the context–dependent grammar as well. It can learn the human's knowledge of tagging. In the initial stage, the tagging is mainly done by human, the system records the operation of human and forms tagging rules, when the rules are accumulated to some number, the system can help human to tag, such as provides human with annotation operations which human did before in the same context, or even do some annotation itself in some cases. The annotation automation gets higher and higher and good consistency is thus guaranteed. It should be mentioned that since PSG non–terminal symbols are used in shift / reduce tagging process, CSTT can produce syntactically tagge d sentences of PSG version as well. In addition, both versions of tree can be mapped into each other by providing with a set of transfer rules.

A small corpora of 1300 sentences of daily life is used for experiment, with the average length of 20 Chinese characters per sentence,For the first 300 sentences, 1455 rules were obtained, and for the whole corpora,totally 6521 rules was obtained. The tagging automation was improved continually with the rules increased, and the automatic tagging ratio is above 50% after 1200 sentences were tagged.

2 DESIGN OF CSTT

2.1 The context–dependent shift / reduce tagging mechanism

The process of context–dependent tagging is that when a sentence is input(the input string is the sequence of part of speech), we look up the rule base with the top two elements of the stack to see whether there exist rules coinciding with the current context. If not, human operation is required to determine whether reduce or shift. If reduce, then further decides what phrase structure will be constructed, and what dependency relation will be constructed between these top two elements. The system records the current context and the operations to forms a new rule, and put it into rule base. Formally, context dependent rule is represented as:

$$\alpha xy\beta \rightarrow s \qquad \text{(Shift)}$$

$$\alpha xy\beta \rightarrow (z,\gamma,h) \qquad \text{(Reduce)}$$

Where x, y are the top two elements in the stack, and α,β are the context on the left hand of x and the context on the right hand of y respectively.The context is represented as a sequence of part of speeches. There are two actions on the right hand of a rule, shift action denoted as s, and reduce action denoted as(z,γ,h).For reduce action, z denotes the phrase structure after reduction, and γ denotes the dependency relation between x and y, h denotes which element is the head of the phrase structure and dependency relation. By $h='A'$means the top element is the head, $h='B'$ means that the second top element of the stack is the head. Now let's see the tagging process for a simple sentence:

我	是	她	的	好	朋友	。
R	VY	R	USDE	A	NG	。

(where, R: pronoun, VY: verb "是", USDE: "的", A: adj., NG: general noun.)

Table 1 The context-dependent shift / reduce tagging process

<Stack>#<Input string>	Action	Phrase structure	Dependency relation
———#<R><VY><R><USDE><A><NG><。>	shift		
———<R>#<VY><R><USDE><A><NG><。>	shift		
——<R><VY>#<R><USDE><A><NG><。>	reduce	SV	SUB
———<SV>#<R><USDE><A><NG><。>	shift		
——<SV><R>#<USDE><A><NG><。>	shift		
—<SV><R><USDE>#<A><NG><。>	reduce	DE	DEP
——<SV><DE>#<A><NG><。>	shift		
—<SV><DE><A>#<NG><。>	shift		
-<SV><DE><A><NG>#<。>	reduce	NP	ATTA
—<SV><DE><NP>#<。>	reduce	NP	ATTA
——<SV><NP>#<。>	reduce	SS	OBJ
———<SS>#<。>	shift		
——<SS><。>#	reduce	SP	MARK
———<SP>#	pop		GOV

(where, SV: subject-verb phrase, DE: "的" structure, NP: noun phrase, SS: sub-sentence, SP: sentence. SUB: subject, DEP: "的" structure, ATTA: modifier, OBJ: object, MARK: punctuation mark, GOV: the predicate of sentence.)

Dependency relation is represented as a triple of the form <modifier, head,the dependency relation>.The tagging result is represented as a set of triples: {<我,是,SUB>, <是,Nil,GOV>, <他,的,DEP>, <的,朋友,ATTA>, <好,朋友,ATTA>, <朋友,是,OBJ>}.At each step, we can obtain a rule by recording the content of stack and input string, and the operation(shift or reduce) given by user. If the operation is a reduction, the phrase structure and dependency relation are to be decided by user. Here are two rules obtained:

——<R><VY>#<R><USDE><A><NG><。>→(SV,SUB,A)
—<SV><R><USDE>#<A><NG><。>→s

After the reduction, the phrase structure formed replaces the top two elements in the stack. And the head will represent this phrase in later process. Since sentences varies with its length, we use three elements on the left side of the top two elements in the stack and the top five elements in the input string as the context.

2.2 The tagging algorithm

The input is a sequence of the part of speech of a sentence, and the output is the dependency tree denoted as a set of triple of the form (modifier, head, the dependency relation), and as a by-product, context-dependent rules are acquired. It is obviously that we can work out the phrase structure tree as well by modifying the algorithm (not detailed in this paper).

Let CDG be the context-dependent rule base which were acquired before,CDG is empty if the system is just put into use. NUMBER-OF-ACTION records the number of total actions(either shift or reduce) during tagging, NUMBER-OF-AUTOMATION is the number of actions(given by the system itself) which are confirmed to be right by human. The automatic tagging ratio is therefore set as NUMBER-OF-AUTOMATION / NUMBER-OF-ACTIONS.

At present, the system is under supervision, human intervention is applied at each step either to confirm the actions given by the system or to append new actions. Ideally, the tagging process should be nearly full automatic with minimum human intervention. But it is a long term process. We believed that with the size of corpora tagged increases, the automatic tagging ratio will be improved, and when it reaches to a degree of high enough, human intervention may be removed, or it may only be needed in the case that no rule is matched.

Table 2 The supervised tagging algorithm

```
BEGIN
  STACK = EMPTY
  NUMBER–OF–AUTOMATION = 0
  NUMBER–OF–ACTION = 0
  DO UNTIL (INPUT = EMPTY AND STACK = EMPTY))
        CONTEXT = APPEND(TOP–FIVE(STACK),FIRST–FIVE(INPUT)) / * get the context * /
        RULE–LIST = CONSULT–TO–CDG(CONTEXT) / * match with CDG * /
        RULE = CONSULT–TO–HUMAN(RULE–LIST) / * human intervention * /
        IF(RULE = FIRST(RULE–LIST)) / * the default operation is right * /
          NUMBER–OF–AUTOMATION++
        NUMBER–OF–ACTION++
        IF RHS(RULE) = 'S'
          STACK = PUSH(FIRST(INPUT),STACK)
        ELSE
          {
          LET (Z,γ, h)BE RHS OF THE RULE
          LET X = FIRST(STACK)   Y= SECOND(STACK)
          BUILD A PHRASE STRUCTURE Z FROM X AND Y
          STACK = PUSH(Z,POP(POP(STACK)))
  / * the phrase structure replace the top two elements of the stack * /
          IF h = 'A'
              BUILD–DEPENDENCY–RELATION(HEAD(Y),HEAD(X),γ)
          / * build the dependency triple * /
          ELSE
          IF h = 'B'
              BUILD–DEPENDENCY–RELATION(HEAD(X),HEAD(Y),γ)
              / * build the dependency triple * /
          }
          IF(INPUT = EMPTY AND NUMBER(STACK) = 1)  STACK = POP(STACK)
  ENDDO
END
```

Function TOP–FIVE, FIRST–FIVE return the first five elements of the stack and input string respectively. If there are less than five elements in the stack or in the input string, then fills with blanks. APPEND merges two lists to obtain the current context. CONSULT–TO–CDG looks up the rule base and returns a list of rules matching with the current context. The list is empty when no rule is matched. If the list is not empty, rules are sorted in descending order of their usage frequency. If human's intervention is default(this may be available when the automatic tagging ratio reaches to some high degree), the system will take a action according to the rule of the highest frequency. CONSULT–TO–HUMAN returns only one rule by human's inspection. In this interactive process, human is asked to determine what action should be taken, he first inspect the rule–list to see if there is already a rule correctly confirming with current context, if not, he should tell the system whether "shift" or "reduce", if "reduce", he is requested to tell the system what phrase structure and what dependency relation is to be built, and which element, the top element of the stack, or the second is the head. A new rule will be acquired when human makes a different operation from existing rules, by recording the current context and the operation. NUMBER–OF–AUTOMATION records the times that the rule with the highest frequency coincides with human's decision, which means that if the system works in automatic way, the rule with the highest frequency is right. NUMBER–OF–ACTIONS records the total times of operation(shift or reduce) during tagging. The

HEAD returns the head word of a phrase. The function PUSH means push an element into stack, and POP pops top element out of stack, FIRST and SECOND return the first element and second element of a list respectively.

In matching process, weighted matching approach (Simmons & Yu, 1992) is used. Assume the set of CDG rules is R= $\{R_1, R_2, .., R_m\}$, where the left hand of each rule is $R_i=\{r_{i1}, r_{i2}.., r_{i10}\}$, assume the context of the top two elements of the stack is $C=\{c_1, c_2, .., c_{10}\}$, where c_4 and c_5 are the top two elements in the stack, we set up a match function:

$$\mu(c_j, r_{ij})=1, \qquad \text{if } c_j=r_{ij},$$
$$\mu(c_j, r_{ij})=0, \qquad \text{if } c_{j!}=r_{ij}.$$

The score function is

$$SCORE=\sum_{i=1}^{3}\mu(c_i,r_i)\cdot i+\sum_{i=6}^{10}\mu(c_i,r_i)(11-i)$$

A rule is preferred if and only if SCORE is greater than a threshold ζ set in advance. $\zeta=21$ means full matching. In the beginning of the system, the full matching is recommended in order to deduce the conflict. And after certain period of tagging, we may set the threshold smaller than 21 to overcome the shortage of rules in some cases. CDG base is controlled dynamically so that to keep high efficiency of matching. A rule will be removed from the CDG base if it is seldom used.

3 EXPERIMENT AND ANALYSIS

3.1 The experiment

A small corpora of 1300 sentences of daily life is prepared for experiment, with the average length as 20 Chinese characters per sentence, the corpora covers main classes of Chinese simple declarative sentences.The experiments is conducted in the following steps:

(1) input a sentence;

(2) word segmentation;

(3) part of speech tagging.

The tagging model is a bi-gram model(Bai & Xia, 1991), and the correct ratio is about 94%, so human confirmation is needed.

(4) tagging the dependency relation by CSTT.

As shown in Table 3, 1455 rules was obtained from the first 300 sentences. In the whole experiment, totally 6521 rules was obtained. The more sentences tagged, the higher automatic tagging ratio may be. After 1200 sentences have been tagged, the ratio of automatic operation is above 50%.

Table 3 The experiment result

Sentence	1–300	400	500	600	700	800	900	1000	1100	1200	1300
No. of rules accquired	1455	447	384	455	486	628	565	572	564	483	492
No. of operation	2072	768	776	792	851	834	846	837	1153	1164	1111
No. of auto operation	487	291	336	281	317	121	237	210	572	641	580
automatic ratio	23.5	37.8	43.3	35.5	37.3	14.5	30.0	25.1	49.6	55.1	52.2

3.2 Discussion

(1) The rule conflict

Although this system has some power for disambiguation due to the context–dependent rules, it is difficult to resolve some ambiguities.Therefore it is easy to understand that a conflict will occur if some ambiguity is encountered. For example, the sequence of VG A NG may be {(A, VG, COMPLEMENT),(NG, VG, OBJ)} or {(A, NG, ATTA), (NG, VG, OBJ)}, and the sequence NG1 NG2 may be {(NG2, NG1, COORDINATE)} or {(NG1, NG2, ATTA)} as the following two pairs of sentence demonstrate:

(i) VG A NG
处理 好 关系 (A, VG, Complement)
treat well relation

养成 好 习惯 (A, NG, ATTA)
form good habit
(ii) NG NG
飞机 大炮 (NG, NG COORDINATE)
plane gun
木头 桌子 (NG, NG, ATTA)
wood table

There are two kinds of ambiguities, one is contextual dependent ambiguity, another is contextual independent ambiguity. For the former, CSTT can resole some of them. For example, 咬死(VG)猎人(NG1)的 (USDE)狗 (NG2)is an ambiguous phrase(which may be {(VG, nil, GOV), (NG1, USDE, DEP), (USDE, NG2, ATTA), (NG2, VG, OBJ)} which means "killed the hunter's dog",or {(VG, USDE, DEP), (NG1, VG, OBJ), (USDE, NG2, ATTA), (NG2, nil, GOV)} which means the dog which killed the hunter. However, if the context is considered, the ambiguity may be resolved:

咬死 猎人 的 狗 死 了
VG NG USDE NG VG Y

一 只 咬死 猎人 的 狗
M Q VG NG USDE NG

Unfortunately, CSTT can't resolve the ambiguity of the later, human–interventionis necessary.

(2) The convergence of the CDG rule

According to the analysis of (Simmons & Yu 1992), 25,000 CDG rules will be sufficient to cover the 99% phenomenon of English common sentences. In this sense, the CDG rule is convergent. If we are only for syntactic tagging, the convergence issues can be avoided temporally, if the automatic ratio reaches above 80%, we can stop acquisition, at this time the tagging can already provide lots help to the users. Of course, if we make some effective attempts to CSTT, it may be developed into an efficient dependency parser as well.

4. CONCLUDING REMARK

In this paper, we presented that dependency grammar is a suitable formalism for syntactic tagging and presented a new technique for developing a syntactic tagging tool for large corpora, in which a simple shift / reduce mechanism was employed and context dependent rules were accumulated during tagging. The supervised tagging algorithm is described. The experiment shows that automatic tagging ratio rises up continually with the number of sentence increases, and good consistency is kept. This idea may be helpful for POS tagging and case tagging of corpora as well.

We hope the automatic tagging ratio will raise above 80% in the future by enlarging the size of rule base, so that it can be practically used for syntactic tagging of running text.

REFERENCES

Bai, Shuan–hu, Yin Xia(1992). A Scheme For Tagging Chinese Running Text. *Proc. of NLPRS, p25–26,* 1991, Singapore.

Furuse O, H. Iida(1992). An example–based method for transfer–driven machine translation. *Proc. 4th TMI–92.* Montreal, 1992.

Isabelle, Pierre, Marc Dymetman et al.(1993). Translation Analysis and translation automation. *Proc. of TMI–93, p201–217.*

Komatsu, Eiji, Cui Jin, and Hiroshi Yasuhara(1983). A mono–lingual corpus–based machine translation of the inter lingua method. *Proc. of TMI–93, p24–46.*

Leech, Geofferey and Roger Garside(1991). Running a grammar factory, the production of syntactically analyzed corpora or "tree banks". In: *English Computer Corpora, p15–32,* Mouton de Gruyter, 1991.

Mitamura, Terko, Eric h. Nyberg, 3rd and

Jaime G. Carbonell(1993). Automated corpus analysis and the acquisition of large, multi–lingual knowledge bases for MT. *Proc. of TMI–93, p292–301,* Kyoto, Japan, July 1993.

Nagao, M.(1984). A framework of a mechanical translation between Japanese and English by analogy example, In: A. Elithorn, R. Benerji, (Ed.), *Artificial and Human Intelligence,*Elsevier: Amsterdam.

Sato, Satoshi(1993). Example–based translation of technical terms. *Proc. of TMI–93, p58–68.*

Simmons, F. Robert, Yeong–Ho Yu(1992). The Acquisition and Use of Context– Dependent Grammars for English. *Computational Linguistics, Vol.* 18, *No.*4, 1992.

Tong, Xiang, Changning Huang, and Chengming Guo(1993). Example–Based Sense Tagging of Running Chinese Text. *Proc. of the workshop on very large corpus,* Academic and Industrial Perspectives, *p102–112,* Columbus, Ohio, USA,June 22, 1993.

Watanabe, Hideo(1993). A method for extracting translation patterns from translation examples. *Proc. of TMI–93, p292–301,* Kyoto, Japan, July 1993.

Zhou, Ming, and Changning Huang(1993). Viewing the Dependency parsing as a statistically based tagging process. *Proc. NLPRS'93,* Japan, Dec. 6–7, 1993.