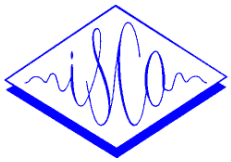

International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)
Taipei, Taiwan
August 23-24, 2002

Developing Chinese TAK for Computer Directly

Guo-Ping HU, Ben-Feng CHEN, Ren-Hua WANG

University of Science and Technology of China, Hefei

rhw@ustc.edu.cn

ABSTRACT

With the development of text analysis, the quality of the computer-used knowledge is more and more crucial to the analysis accuracy, and the text analysis knowledge (TAK) has also developed by many researchers. But so far, except the lexicon, TAK for computer (such as phrase structure grammar, unregistered word recognition rule, etc) is done on a small scale. Although large scale corpus with word segmentation annotation and even treebank has been developed, all these projects contribute limitedly to the text parser compared with the huge workload of the annotation, especially in Chinese domain. Considering the disadvantages of the data-mining and training technology used in text analysis field, aiming at one TTS system, this paper demonstrates a complete set of solutions to develop Chinese TAK for computer, including lexicon tree, nesting phrase structure grammar, combination-bigram, developing flow with computer's aid, and checking and improving the quality of the TAK automatically with the treebank (the treebank is the by-product of this development). This paper also shows that a text analysis system based on the construction result hits an accuracy rate of 80% in a close testing set of 24700 sentences, and approximately 50% tested on an open corpus. It is thus deduced that directly developing Chinese TAK for computer is more effective than other approaches under same workload.

Keywords:

Text Analysis Knowledge, Chinese Parser, Lexicon, Treebank, Syntactic tree, Rhythm tree

1. INTRODUCTION

With the development of text analysis technology, the quality of the knowledge used by computer undertakes more and more responsibility to the analysis accuracy, and the text analysis knowledge (TAK) is also developed by many researchers. In this research, the English research adopts the approach of building a large scale of treebank first, and then tries to find the TAK for computer by data-mining and training technologies. A large scale of tagged corpus has been developed, such as Lancaster-Leeds treebank project [LG91] in England, and Penn treebank project [MSM93] in America [1]. Both projects achieve the scale of two millions words. As to Chinese, not only scales, but also speed and consistency are far behind those of English. *Zhou Qiang* has put forward and constructed a Chinese treebank of about 78 thousand words in 1997[2], *Zhu Jingbo* built a Neu Chinese Semantic Treebank on 3000 sentences [3], and the Penn Chinese Treebank project developed a corpus of 200,000-words[4]. All these treebanks adopted the approach of the English treebanks, and to build a Chinese treebank whose scale can drive training is a very difficult and hard work, *Zhou Qiang* believes that the minimum size of treebank should be based on 2~5 millions words [1], yet to our opinion more is needed

We prefer to develop Chinese TAK for computer directly, and the reasons are: 1) the status of Chinese Treebank is poor nowadays; 2) the particularity of Chinese itself, the knowledge needed for many Chinese sentences to be analyzed correctly is too particular and too trivial to be achieved by training; 3) We can depend less on training technology which often can not perform good enough in Chinese domain; 4) Developing TAK for computer firstly and then building treebank with computer automatically ensures more consistency of the treebank 5)Efficiency and speed comparison: to our experience and statistics result, of course, teaching the computer to analyze one sentence accurately is more painstaking than annotating one sentence at the beginning, but with the accumulating and perfecting of the TAK for computer, the time needed in our method will be cut down. And to our current status, the time needed in the two different approaches almost same. Prof. *Yu Shiwen*, a pioneer in the developing Chinese TAK for computer field, has developed the Lexicon of Modern Chinese Syntactic Information, which is fundamental to many Chinese parsers [5]. And his paper "*About the Construction of synthetically language knowledge database*" [6] also shows his intent to directly build Chinese TAK for computer.

But isolated development will encounter a lot of problems too, and the most serious one is its robustness. The TAK concluded on a small corpus by hand can't achieve high robustness. Besides probability is always used in text analysis systems to solve the confliction among the units of TAK, but the globally best

probability is also an impossible achievement just by hand.

Considering all the problems, we present a complete set of solutions to develop Chinese TAK for computer directly, and demonstrate our TAK. This paper is structured as follows: In Section 2, we describe the three major components of the Chinese TAK, and show how they are adapted to be realizable. In Section 3, an overall flow of developing TAK with computer is described. In Section 4, the result of our experiment is presented. Finally, we give our conclusions in Section 5.

2. CHINESE TAK FOR COMPUTER

We construct the Chinese TAK orienting to the front-end (text processing) of a Chinese TTS system, and the processing flow is: word segmentation, special marks processing, unregistered word recognition, polyphone adjustment, syntactic tree building and rhythm tree generation. The rhythm tree is presented to the back-end of the system to speech synthesis. Almost each step needs the TAK's support, and each component of the TAK should be carefully analyzed and specially designed. Here we present three major components, including lexicon tree, phrase structure grammar, and combination-bigram (not co-occurrence). The adaptation on the three components is also demonstrated.

2.1. Lexicon Tree: POS hierarchical definition and word clustering

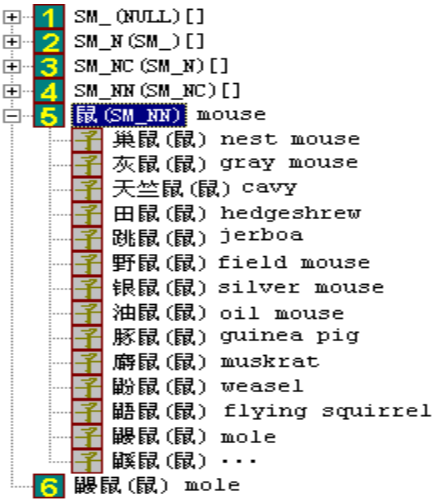

Figure 1: the "鼹鼠"(mole)'s position in the lexicon tree

Our lexicon is based on that developed by *Yu Shiwen*, but the structure of the lexicon is re-designed. We extend the POS concept by POS hierarchical definition. Usually, each lexical entry is tagged with one POS, and this kind of POS is similar to one layer of non-terminal symbols. Construction on original defined POS, it is comprehensive that the granularity of POS is too big while the granularity of word is too small. So to settle this problem, we present POS hierarchical definition of multi-layers of POS. The lexicon is structured not like a list or a table as usual, but a tree! Figure 1 shows the position of "鼹鼠"(mole) in the lexicon tree. "鼹鼠"(mole)'s POS (or meaning to our definition) is "鼠"(mouse), and "鼠"(mouse)'s meaning is SM_NN(noun), …, SM_N's meaning is SM_. So we can select different layer in the lexicon tree for TAK development and achieve suitable granularity

3. SM_V: Verb
   - 3.1 SM_VI: intransitive verb
   - 3.2 SM_VO: transitive verb that can only take one direct object
   - 3.3 SM_VT: transitive verb that can take double objects
   - 3.4 SM_VS: the pivot verb that can take one object, which also acts as the subject of a clause
   - 3.5 SM_VV: verb that can take verb as its object
   - 3.6 SM_VX: verb that can take adjective as its object
   - 3.7 SM_VJ: the verb that can take clause as its object
   - 3.8 SM_VP: the verb with a preposition postfix
4. SM_N: Noun
   - 4.1 SM_NT: the noun denoting time concept
   - 4.2 SM_NM: the numeral
   - 4.3 SM_NQ: the quantifier
     - 4.3.1 SM_NQJ: the quantifier used with noun
     - 4.3.2 SM_NQD: the quantifier used with verb
   - 4.4 SM_NF: the noun denoting direction
   - 4.5 SM_NC: the common noun
     - 4.5.1 SM_NP: the noun denoting person
     - 4.5.2 SM_NN: the simple noun
       - 4.5.2.1 SM_NS: the noun denoting location

Figure 2: POS hierarchical definition (partly)

Figure 2 shows our POS hierarchical definition. We distinguish the usage of traditional POS (such as verb and noun) and divide them into several sub-POS, and this work contributes a lot to the phrase structure grammar by providing suitable granularity selection. Meanwhile, we cluster the original 200,000 lemmas by hand. Two kinds of clustering are adopted: horizontal clustering and vertical clustering. Horizontal clustering is to cluster one lemma to its brother lemma which does not cover the means of this lemma but has similar usage, such as "上海"(*Shanghai*) to "北京"(*Beijing*) and "周恩来"(*Zhou Enlai*) to "毛泽东"(*Mao Zedong*); and when a lemma is clustered to its parent lemma which covers the means of this lemma, we call it vertical clustering, such as "科学技术" (Science and technology)to "技术"(technology), "淹死"(drown) to "死"(dead). With these two clustering methods, we obtain more non-terminal symbols, such as "北京"(*Beijing*), "技术"(technology) and the "鼠"(mouse) in figure 1. These non-terminal symbols ensure that the selection of proper TAK granularity is more easily. Of course, the clustering will lose some useful information inevitably, but to our opinions, just because of this losing constructing computer-used TAK becomes realizable.

As the tree structure is adopted for lexicon, the lemma of multi-POS is added to the lexicon tree's different branches (positions) according to its POS (usage) kinds. Maybe the unique

direction graph model can replace the tree structure in future.

2.2 Nesting grammar with reliability and nesting combination-bigram

To be more flexible, our phrase structure grammar supports from two to four sub-nodes combining together, and a headword is assigned to the combined structure, which can join the further combination. According the lexicon tree definition, the rule has a concept of nesting too. For example, rule A:"SM_VS+SM_NP+SM_VI=SM_VI" is the sub rule of rule B: "SM_VS+SM_NC+SM_VI=SM_VI" because every combination according rule B also conform to rule A. This nesting design is useful to solve the confliction in rules selection. Each grammar rule has a reliability (probability) as the parameter of the evaluating function of syntactic tree. Rule A and rule B may have different reliabilities though they are nested, so when rule A conflicts with another rule in TAK, we can cut down the rule A's reliability and add up that of rule B to ensure validity.

We have also defined five kinds of combination-bigram (i.e. relationship between two words):coordinate(并 列), modifier-modified(偏正), verb-object(动宾), subject-verb(主谓), and headword-complement(述补). There are none, one or several kinds of relationship between two words. For example, it is parallelity relationship between " 老 师 "(teacher) and " 同 学"(classmate) in the sentence "老师和同学都回家了"(Teacher and classmates have gone home), but qualifier-headword relationship in the sentence "老师的同学来了"(the teacher's classmate is here). To distinguish the five kinds of relationship is very valuable. When we draw one grammar rule, we annotate what kind of relationship is needed or should be checked between the sub-nodes joining the combination. For instance, predicate-objective relationship should be detected between the "SM_VS" and "SM_NP" nodes in the rule "SM_VS+SM_NP+SM_VI=SM_VI", and in the same rule, between the "SM_NP" and "SM_VI" nodes, the subjective-predicate relationship should be detected. And all these messages are useful to minimize the confliction between rules.

According to the lexicon tree, the roughness and complexity of nature language can be fitted easily: the child-lemma inherits his meaning-lemma's usage defined in TAK by default, such as phrase structure grammar, combination-bigram; and when one child-lemma does not inherit one usage of his meaning-lemma, we can draw more trivial rules referring to the child-lemma. We can achieve high accuracy and efficiency in TAK construction by choosing proper granularity.

2.3 Training the reliability from treebank automatically

Drawing grammar rule is not difficult to native people, while achieving best reliability distribution to solve most confliction is a difficult work even to the experts. But this is just the strongpoint of computer. So we combine the advantages of handwork and computer, we first draw the grammar rules, and then train the rule's best reliability distribution using some training algorithm within the treebank, which is the by-product of the TAK construction.

We can deal with the combination-bigram's reliability in the same way, but it is not valuable because the treebank needed to train bigram is much larger than that to train grammar rule's reliability; and to combination-bigram, qualitative analysis is already enough, there is no need to train reliability of bigram from treebank.

3. CONSTRUCT CHINESE TAK BY AID OF COMPUTER

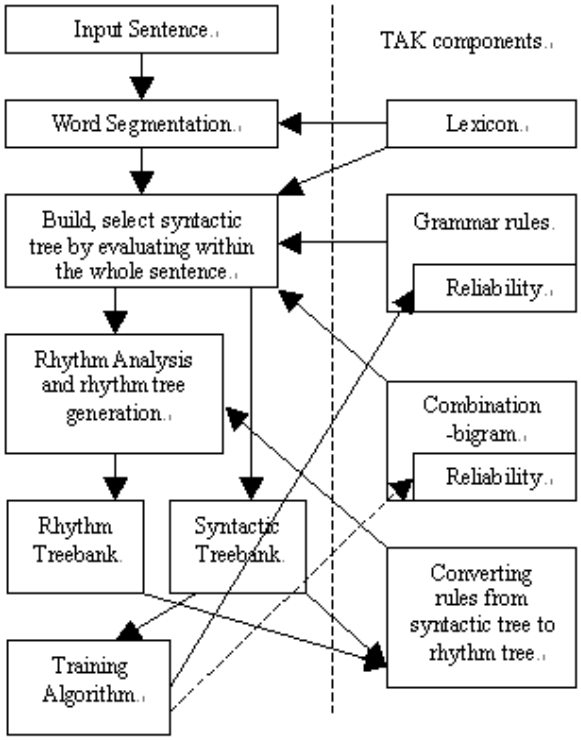

Figure 3: the flow of developing Chinese TAK

Figure 3 shows the flow of developing Chinese TAK. It is based on a syntactic and rhythm analysis text processing system, and the core process is to build and select syntactic tree by evaluating within the whole sentence. The construction generates two kinds of treebank, syntactic treebank and rhythm treebank, and the treebanks are utilized to check the accuracy of the system, and

also input into the training algorithm to train the reliability automatically.

4. EXPERIMENT

The following is the outline of our experiments:

1) Firstly, we construct lexicon with 200,000 lemmas, and for each lemma, we checked its POS and the meaning lemma should be clustered to. Now there are 15738 non-terminal symbols in the lexicon tree.

2) We performed TAK construction within a corpus size of 10000+2000+5000+7700=24700 sentences (about 500K bytes). These sentences come from *RenMinRiBao*, *GuangMingRiBao* and Chinese textbook of Singapore's primary school. When the syntactic treebank of the 24700 sentences is completed, we totally drew 1226 grammar rules and 11594 combination-bigrams. And the system with the TAK approach 75.2% accuracy when tested in the treebank.

3) We recorded the original data whether one sentence can be analyzed accurately before changing or adding any TAK according to this sentence, and we call this probability "first accuracy". The first accuracy's curve is shown in table 1:

Table 1:First Accuracy Rate

Term	1	2	3	4
Sentence Count	10000	2000	5000	7700
First Accuracy	12.43%	8.2%	32.4%	50.2%

The curve demonstrates that first accuracy (and the open testing accuracy) is improving, and also indicates that the TAK's construction is speeded up.

4) We trained the reliability of the grammar rule using iterative algorithm within the treebank, and the accuracy is promoted to 80.1%.

5. CONCLUSION

Although many kinds of tagged corpus have been constructed, they don't contribute much to the text analysis system. What's more, though a lot of TAK for computer has been done, still many TAK such as grammar rules, combination-bigrams are not believed to be realizable. In this paper, we made a series of adaptation on TAK structure, including lexicon tree, nesting rules with reliability and nesting combination-bigram, and our experiments demonstrate that after this adaptation, these kinds of TAK's development becomes realizable, and also shows that developing TAK directly can be more efficient, worthful and more economical.

6. ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (69975018), and China National Hi-Tech Project under N0.2001AA110333.

7. REFERENCE

[1] Building a large annotated corpus of English: the Penn Treebank, Mitchell Marcus, *Computational Linguistics* 19(2): 313-330. 4

[2] Construction of Chinese Treebank, *Zhou Qiang, Zhang Wei, Yu ShiWen*, Journal of Chinese Information Processing, 1997

[3] Neu Chinese Semantic Treebank Manual, *Zhu JingBo, Yao TianShun,* Northeastern University, China, 2000

[4] Developing Guidelines and Ensuring consistency for Chinese text annotation, *Fei Xia*, etc, Proceedings of the second International Conference on Language Resources and Evaluation (LREC-2000), Athens, Greece, 2000.

[5] Annotation for the Lexicon of Modern Chinese Syntactic Information, *Yu Shiwen*, TsingHua University Press, 1998

[6] About the Construction of synthetically language knowledge database, *Yu Shiwen*, Institute of Computational Linguistics of Peking University, 2001

[7] The PENN Treebank: Annotating Predicate Argument structure, Mitchell Marcus, *Proceedings of the Human Language Technology Workshop*, Morgan Kaufmanns, San Francisco.

[8] A Study of Constructing Rules of Phrases in Contemporary Chinese for Chinese Information Processing, *Zhan Weidong*, the thesis of doctor, 1999

[9] Leaning to Parse Natural Language with Maximum Entropy Models, Adwait Rathnaparkhi, Dept. of Computer and Information Science University of Pennsylvania, 1998

[10] Hierarchical Network of Concepts, *Huang Cengyang*, TsingHua University Press, 1998

[11] A Block-Based Robust Dependency Parser for Unrestricted Chinese Text, *Zhou Ming*, Microsoft Research China, 2000