Annotating the functional chunks in Chinese sentences

Qiang Zhou*, Elliott Franco Drabek*, Fuji Ren†

* State Key Laboratory of Intelligent Technology and Systems
Dept. of Computer Science and Technology,
Tsinghua University, Beijing 100084, P. R. China
zhouq@s1000e.cs.tsinghua.edu.cn, elliott_drabek@ACM.org[1]

†Dept. of Information Science and Intelligent Systems
Faculty of Engineering, The University of Tokushima
2-1 Minamijosanjima, Tokushima 770-8506, Japan
ren@is.tokushima-u.ac.jp

Abstract

The paper proposed a new syntactic annotation scheme --- functional chunk, which tried to represent information about grammatical relations between sentence-level predicates and their arguments. Under this scheme, we built a Chinese chunk bank with about two million Chinese characters, and developed some learned models for automatically annotating fresh text with functional chunks. We also proposed a two-stages approach to build Chinese tree bank on the top of chunk bank, and gave some experimental results of chunk-based syntactic parser to show the advantage of functional chunk for parsing performance increase. All these work lays good foundations for further research project to build a large scale Chinese tree bank.

1. Introduction

Corpus annotation is a good way to acquire the knowledge of language performance in real texts. How to represent that knowledge in the most informative and efficient way is its fundamental question. There are two extremes among the syntactic annotating schemes now. The simplest way is to assign the words with part-of-speech tags, i.e. categories designed to reflect the syntactic behavior of words. Two typical examples are Brown and LOB corpus in English. This is a good start, and many efficient algorithms have been proposed to automatically assign the correct POS tags with very high precision and recall. But intuition and plentiful experiments tell us that this is not enough, and the idiosyncrasies of individual words and their syntactic relations must be considered. The most complex way is to tag the complete syntactic trees in sentences. A typical example is Penn tree bank (Marcus et al., 1993). Although they can give detailed syntactic information descriptions of the real text sentences, the cost to build a large-scale tree bank is very expansive.

In the resent years, many researchers have devoted to design some better annotation schemes with suitable trade-off between annotation cost and description capability. Abney's chunk parsing scheme is a good example among them. He defined a chunk essentially as the continuous set of words belonging to a single s-projection, or the domain of a single semantic head. (Abney, 1991). Based on this scheme, CONLL'2000 shared chunk task built an unified training and test set for partial parsing in English (Sang and Buchholz, 2000). The text of the corpus was taken from the Wall Street Journal portion of the Penn Treebank, and the chunks were extracted automatically from the original syntax trees, using straightforward heuristic rules. The quantity of the chunk annotated text amounts to approximately three hundred thousand words, with an average chunk length of two words. Many automatic partial parsers proposed in the conference showed good parsing performance. But it's still very difficult to grasp the overall structure of a sentence only based on these chunk information.

The paper proposed a new syntactic annotation scheme --- functional chunk, which tried to represent information about grammatical relations between sentence-level predicates and their arguments. Among this scheme, each sentence can be exhaustively partitioned into a series of non-nested, non-overlapping units, labeled with functional tags, such as subject, object, predicate, complement and so on, while any structural relations within these chunks are left implicit. Compared with Abney's chunk, the novelty of functional chunk appears in the definition of what constitutes a chunk. Abney's chunks are defined strictly from the bottom up; a unit qualifies as a chunk based on its internal make-up, regardless of any changes in the larger context, and its category is primarily determined by the category of its head word. By contrast, the functional chunks are defined strictly from the top down; a unit qualifies as a chunk based on its position in the larger context, regardless of its internal make-up, and its category is primarily determined by the grammatical relation between it and the predicate. This top-down characteristic gives more detailed information to grasp the overall structure of a sentence than Abney's chunk.

Under this scheme, we built a Chinese chunk bank with about two million Chinese characters, and developed some learned models for automatically annotating fresh text with functional chunks. We also proposed a two-stages approach to build Chinese tree bank on the top of chunk bank, and gave some experimental results of chunk-based syntactic parser to show the advantage of functional chunk for parsing performance increase. All these work lays good foundations for further research project to build a large scale Chinese tree bank.

[1] Elliott F. Drabek has been a Ph.D. candidate in John Hopkins University since Sept. 2001.

The paper is organized as follows. Section 2 gives detailed definition and description of our functional chunk scheme, and compares it with other annotating systems. Section 3 introduces the functional chunk annotation experiment on a Chinese balanced corpus with two million Chinese characters, including the annotation stylebook, processing procedure, and some detailed statistics of the functional chunk bank. Section 4 describes some of our work in developing learned models for automatically annotating fresh text according to this scheme. Section 5 proposes a two-stages approach to build Chinese tree bank on the top of chunk bank, and gives some experimental results of chunk-based syntactic parser to show the advantage of functional chunk for parsing performance increase. The final section 6 is a conclusion.

2. Functional Chunk Scheme

The functional chunk scheme we proposed in this section tries to represent information about grammatical relations between sentence-level predicates and their arguments. Under the annotation framework, each sentence is exhaustively partitioned into a series of non-nested, non-overlapping labeled units, or functional chunks, while any structural relations within or between these chunks are left implicit.

In the typical case, each clause is divided into one chunk for the main verb, one chunk for each of its argument constituents and several chunks for the adjuncts of the main verb or some elements present but not dependent on the main verb, such as exclamations or vocative phrases. In the Chinese language, the arguments of a verbal predicate can be expressed as subject, object, complements, or some special adverbial adjuncts(most of them are prepositional phrases with case flags). They are formed as the basic syntactic patterns of Chinese sentences. Table 1 outlined all eight functional chunks used in our tagset, where chunk 'J', which we call as 'JianYu' in Chinese, describes the special linguistic phenomenon in Chinese serial verb constructions, where the object of a pre-verb can be combined with the subject of its post-verb. Chunk 'T' describes some elements present but not dependent to any other constituent in the sentence, such as exclamations or vocative phrases. An annotated example sentence is shown in as follows:

[S 一/m 段/q 难以/v 忘怀/v 的/u 往事/n] [D 又/d] [P 浮现/v 在/p] [我/r 的/n 眼前/s] 。/w[2]

Some unforgettable events from the past once again floated in front of my eyes.

In most cases, the set of possible chunk sequences is tightly constrained, so that heuristic rules which work on the most common cases should be able to recover almost all grammatical relations. The following regular expression embraces about ninety percent of the observed sequences in the Chinese sentences (ignoring independent constituents):

```
D* (S D*)? P O? (D* ([JS] D*)? P [CO]?)* Y?
```

In general, this means that each sentence is made up of a series of clauses, possibly followed by a modal particle. Each clause contains one predicate (P) chunk. Preceding the predicate there may be some number of adverbial (D) chunks, possibly with one subject (S) chunk among them. Following the predicate, there may be one direct object (O) chunk or one raised object (J) chunk, one indirect object (O) chunk or one complement (C) chunk.

Table 1. The tagset of our functional chunks

Chunk Tag	Basic Function Description
S	Subject
P	Predicate
O	Object
J	Raised object
D	Adverbial adjunct
C	Complement
T	Independent constituent
Y	Modal particle

In summary, the functional chunk annotation builds basic links between functional structure (Kaplan and Bresnan, 1982) and argument structure (Alsina, 1996). Although there is no explicit annotation of connections between specific predicates and specific arguments, that information should be largely recoverable from the sequence of chunk categories. As we know, it is the first research work to describe the argument structure through syntactic function annotation.

In fact, there are several other annotation scheme to describe argument structure. After finishing the large-scale corpus annotation of skeleton syntactic tree in English texts (Marcus et al., 1993), the Penn Treebank has implemented a new syntactic annotation scheme (Marcus et al., 1994), designed to highlight aspects of predicate-argument structure. Through a four-steps editing approach to the original Penn Treebank, a bank of predicate-argument structures can be automatically extracted and built for parser evaluation and other NLP applications. It is the research work to describe the argument structure through predicate-argument annotation itself.

Another interested project is FrameNet developed in UC, Berkeley (Baker et al., 1998). Its primary aim is to produce frame-semantic descriptions of lexical items through the construction of a large-scale semantically tagged corpus. Based on frame semantics proposed by Fillmore(1982), they associated a word with a frame, chose a list of frame elements (FEs) for the frame, looked at sentences which showed a use of the word in the appropriate sense, and selected and labeled those constituents in those sentences which instantiated concepts from the frame. The constituents identified as FEs were then to be classified (automatically if possible) as to their phrase type (PP, etc.) and in respect to their grammatical function (Object, etc.). Therefore, a lexicon with full descriptions of the frame-semantic and syntactic combinational properties of the words can be constructed automatically from the set of annotations (Fillmore et al., 2001). It is the research work to describe the argument structure through semantic role annotation.

[2] The part-of-speech tags used in this sentence are briefly describes as follows: m--numeral, q--classifier, v--verb, u-auxiliary word, n--noun, d-adverb, p-preposition, s—situation noun, w--punctuation.

3. Chunk Annotation Experiments

In March 2000, a chunk bank project work began to start. It aims to build a large-scale Chinese chunk bank (THChunk), conceived as a broadly useful linguistic resource for Chinese linguistics and language engineering. The project ended in June 2001. About two million Chinese characters of text were manually annotated with functional chunks. We believe that the particular combination form and content of the annotation used in the THChunk is relatively novel.

3.1. Basic corpus

The THChunk forms a layer of chunk annotation on top of the original ThCorp, a corpus containing two million Chinese characters of text drawn from a balanced collection of journalistic, literary, academic, and other documents. All of the material has been hand corrected after processing of sentence-split, word segmentation and part-of-speech tagging by automatic tools. Table 2 lists some basis statistics of ThCorp.

Table 2 Basic statistics of ThCorp

Text Type	File Sum	Sent. Sum	Word Sum	Char. Sum
Academic	29	9846	273017	447288
Journal	376	16921	427649	674566
Literary	295	38258	740445	1018839
Others	258	4302	88452	144027
Total	958	69327	1529563	2284720

3.2. Annotation stylebook

To design a detailed annotation stylebook is a prerequisite for high levels of inter-annotator agreement. But the linguistic phenomena in real texts are complicated and ticklish, it is very difficult to comprise all of them completely in a stylebook at first. Substantially, it is an incremental process for improvement. At first, we summarized some basic principles, such as how to separate a complete sentence into several clauses, how to identify and annotate different functional chunks in a clause, etc. Then, some detailed rules and examples can be added into the stylebook to describe the special linguistic phenomena encountered during the annotation task. At last, a better stylebook covering almost all linguistic phenomena in our current ThCorp corpus has been completed.

Some detailed information of the stylebook can found in (Zhou, 2000). It is our hope that such a stylebook will also alleviate much of the need for extensive cross-talk between different annotators, thereby increasing throughput as well.

3.3. Annotating procedure

Our current chunk bank was built through manual annotation and proofreading. To improve the annotating efficiency, we designed some special macro commands in Microsoft's WORD so that each functional chunk can be tagged through one-key input. Our tentative count shows that original annotating speed for an annotator is about 1200 words per work hour. As they are familiar with the annotation stylebook and processing procedure deeper and deeper, the annotation speed will gradually increase and reach about 2400 words per work hour after 1 or 2 months.

We also designed two-levels checking system to guarantee the quality of the final annotating results. Firstly, we developed an automatic checking program based on the basic principles and rules listed in chunk stylebook. Most wrong chunks can be checked out and provided to annotator for further conformation or modification. Secondly, We checked the final annotating results through random sampling, found and modified the chunk errors left, until the error ratio is below 0.5%.

3.4. Basic statistics of chunk bank

Table 3 lists some basic statistics of current chunk bank, including the sum of different functional chunks, the sum of words among them, and their average chunk length (ACL). We divided these chunks into 4 types, according to the different word sum(WSum) among them: 1) WSum < 5; 2) 5<=WSum<10; 3) 10<=WSum<15; 4) 15<=WSum. Table 4 shows the distributional data for these 4 types of different chunks.

Table 3 Basic statistics of different functional chunks

Chunk Type	Chunk Sum	Word Sum	Average Chunk Length
S	99121	251041	2.53
P	179605	236104	1.31
O	109362	452211	4.13
J	5715	12338	2.16
D	156000	321254	2.06
C	3113	6431	2.07
T	5649	14414	2.55
Y	12111	12225	1.01
Total	570676	1306018	2.29

From these two tables, some distributional characteristics of functional chunks in Chinese text can be summarized as follows:

1) The 'Y' chunk was defined as one or more modal particles at the end of a sentence. But few sentences in Chinese text have two or more final modal particles. So it has the minimal ACL (1.01) among all chunks.

2) Most of the predicate (P) chunks in Chinese sentences are verbs, adjectives or phrasal verbs, whose forms have been strictly defined in our annotation stylebook. They have the more regular distributional features (ACL=1.31 and the word sum among more than 99% 'P' chunk is less than 5) for automatic identification.

3) The ACL of the 'D' chunk and 'C' chunk is about 2, and the word sum among more than 90% of these chunks is less than 5. These cases indicate that there are few complex adverbials and complements in Chinese real text. Some obvious boundary flags of them provide important heuristic features for automatic identification.

4) The 'S' and 'J' chunk have larger ACL (2.53 and 2.16), and the 'O' chunk has the maximum ACL (4.13). The main reason is that most of them contain an entire restrictive clause. They bring in the most difficulty for automatic parsing models.

Table 4 Word length distribution of different functional chunks

Chunk Type	Chunk Sum	WSum∈ [0,5)	Ratio (%)	WSum∈ [5,10)	Ratio (%)	WSum∈ [10,15)	Ratio (%)	WSum∈ [15,∞]	Ratio (%)
S	99121	85208	85.96	11023	11.12	1939	1.96	951	0.96
P	179605	178545	99.41	862	0.48	144	0.08	54	0.03
O	109362	75745	69.26	24569	22.47	5888	5.38	3160	2.89
J	5715	5134	89.83	482	8.43	70	1.22	29	0.51
D	156000	141060	90.42	11863	7.60	2151	1.38	926	0.60
C	3113	2857	91.78	219	7.04	31	1.00	6	0.19
T	5649	4984	88.23	388	6.87	136	2.41	141	2.49
Y	12111	12111	100.00	0	0.00	0	0.00	0	0.00
Total	570676	505644	88.60	49406	8.66	10359	1.82	5267	0.92

4. Parsing models

The aspect of the chunking system which likely poses the greatest difficulty for automatic annotation is the potentially unbounded length and internal complexity of the chunks. The top-down definition implies that the status of a constituent depends only on its context, and not on its content, so that a chunk may encompass arbitrarily many modifiers within itself. The annotator must recognize these as belonging to the same unit, even though under other circumstances exactly the same constructs might need to be separated and labeled as individual chunks themselves. This would seem to pose the most acute difficulty in the common circumstance that an entire subordinate clause appears within a single chunk.

The machine learning algorithm we used for almost all of our experiments was the C4.5 decision tree system (Quinlan, 1993). Although more sophisticated learners may be available, C4.5 has the advantages of familiarity, ease of use, and computational efficiency.

In the following sections, we will briefly introduce our parsing models and current experimental results. Some detailed information can be found in (Drabek, 2001).

4.1. The baseline models

Our baseline models base their judgements entirely on the part-of-speech sequences making up each sentence. We conducted our experiments in parallel within two basic parsing frameworks. The first implements a top-down parser, which uses a generative probability model to represent knowledge of what kinds of syntactic structures are likely, and what inputs these structures are likely to be associated with. The second implements a bottom-up parser, which uses a constructive probability model or decision module to represent knowledge of what parsing actions are likely to be appropriate in response to a given input.

1) The Generative Model

The generative models model the sentence generation process as a series of events generating chunk categories, and then for each chunk, a series of events generating words. Each of these series is ended by the generation of a special STOP symbol. The probability assigned to a given analysis of a sentence is the product of the probabilities for all the generation events necessary to generate the structure and the words of the sentence. Different models are specified by specifying probability models for these two events, the generation of a chunk category, and the generation of a word within a chunk. In the baseline model, probabilities for the generation of chunk categories are estimated simply, based on the preceding three chunk categories. The generation of a word means simply the generation of a part-of-speech tag, and this probability is estimated based on the preceding three words.

2) The Constructive Model

The constructive models model the parsing process as a series of construction events, alternating between creating (and labeling) chunks and then extending each, word by word, until it is completed. The probability assigned to a given analysis of a sentence is the product of the probabilities for all the construction events necessary to build up that structure. Different models are specified by specifying probability models for these two events, the choice of a chunk category, and the choice of whether to break or extend the current chunk. Because this model does not need to be as concerned about maintaining a probability distribution, decisions can be based on a wider range of conditioning information.

4.2. Feature engineering

For improving the parsing performance, our tentative idea was to extend the above models with more detailed syntactic features than the POS tags. We selected Grammatical Knowledge base of Contemporary Chinese (GKBCC) (Yu et al., 1998) as our main knowledge resource. Developed specifically for natural language processing in the Chinese language, it contains information specifically about the syntactic behavior of more than fifty thousands Chinese words.

The form of the dictionary seems to give a very direct fit to the feature vector representation used by C4.5, but closer inspection revealed a number of places where this fit is less than perfect, and several adaptations were necessary. The two larger issues of feature engineering we needed to address in order to apply the dictionary to our task were feature presentation and feature selection.

Here, feature presentation refers to the mapping between dictionary features, which are functions of words, and the model features, which are functions of automaton states. We chose to restrict our models to the simplest form, simply adding the dictionary features of words in

specified positions. These positions are specified in the same way as those for part-of-speech tags in the baseline models, but because of the very large number of available features, we chose to restrict each model to using three such positions.

The greatest practical difficulty with using the dictionary was the sheer number of the features available. This made even training such a model prohibitively compute-intensive, and seemed likely to introduce more noise than information. We used a further process of combined hand selection and automatic methods to eliminate features from the models described as many terms as possible without degrading performance on a separate validation set, and approximately halved their number in the final model.

4.3. Experimental results and analysis

The 6814 sentences (142,759 words) extracted from the THChunk were used at the time of the experiments to generate a test set containing every tenth sentence, and a training set containing the remainder. Table 5 shows the experimental results. Where 'GPM' represents Generative Parsing Model, 'CPM' represents Constructive Parsing Model, 'B' represents Baseline models, 'E' represents the Enhanced models after feature engineering.

Table 5 Experimental results of different models

Models	Labelled Precision	Labelled Recall	F-measure
GPM_B	71.0%	67.8%	69.4%
CPM_B	75.4%	79.2%	76.2%
GPM_E	74.3%	74.2%	74.2%
CPM_E	78.3%	79.8%	79.0%

Our experiments have shown that the information from a rich lexical resource can be made directly useful to a simple machine-learning based shallow parser to improve its parsing performance, without significant effort in linguistic engineering. The information is sufficiently rich to provide a useful basis for generalization, and the machine-learning algorithm is able to decide how and when to use this information.

5. From chunk bank to tree bank

As an important middle product, the completed chunk bank also plays a key role in our two-stages approach to build a large scale Chinese tree bank. Figure 1 shows the overview of this method.

One of the best-known efforts to produce corpora with syntactic annotations is the Penn Treebank project (Marcus et al., 1993). It currently includes two corpora annotated with part-of-speech tags and skeletal syntactic parse trees. The effort in annotating corpora in this project included an automated first step(using a part-of-speech tagger and a parser), but relied heavily on the manual efforts of linguists to achieve high-quality linguistic annotations. Although this approach yields a high level of accuracy, it is impractical if time is relatively limited and a team of linguists dedicated to corpus-annotation is not available. In fact, a number of syntactically annotated corpora (or treebanks) have been produced in recent years (Garside et al., 1997, Skut, 1997), with varying amounts of automation, but typically with human effort playing a major role in the annotation process.

In our opinion, the manual efforts are inevitable in the construction of a good syntactically annotated corpus. The key issue is how to reduce them as far as possible through suitable human-machine collaboration. As we know, the biggest problem of many current automatic parsers lies in their poor disambiguation ability. They have difficult to process some typical ambiguous structures in real texts, such as the prepositional phrase attachment problem in English and the "v np uJDE np" structure in Chinese, especially in complex sentence. In these respects, human has their advantages. If we can separate the complex sentence into several chunks with special syntactic functions through suitable manual preprocessing, then provide them to the automatic parser for syntactic parsing, many ambiguous structures in the sentence will be eliminated or restricted in the smaller context. Therefore, the accuracy of the parsed results will be greatly improved and the man workload for post-proofreading will be greatly reduced.

Starting from the above train of thought, we conceived the plan of building a large-scale Chinese treebank through the following two-stages approaching:

At the first stage, each sentence in the corpus can be exhaustively partitioned into a series of non-nested, non-overlapping units, labeled with functional tags, such as subject, object, predicate, complement and so on, while any structural relations within these chunks are left implicit. The top-down characteristics of the functional chunk scheme guarantee its suitableness for manual annotation and proofreading. Some automatic chunking models can be also used in the future if possible.

At the second stage, each sentence can be automatically parser through a chunk-based syntactic parser, where the skeleton tree of the sentence can be easily built based on the implicit grammatical relations among different functional chunks, and many base phrases can be restrictively parsed among the small context of functional chunks. Therefore, the searching space for the syntactic parser will be greatly reduced and the parsing performance will be greatly increased. After further manual proofreading, the complete tree bank can be built.

Using this two-stages annotating method, we built a small-size Chinese tree bank (THTree) consisting of about 200,000 Chinese words. All the sentences of THTree was automatically extracted from THChunk based on a special sampling algorithm (Zhou and Sun, 1999). Table 6 lists some basic statistics of ThTree.

Table 6 Basic statistics of ThTree

Text Type	File Sum	Sent. Sum	Word Sum	Char. Sum
Academic	16	5864	164292	270227
Journal	14	458	11671	18605
Literary	19	1273	25678	38794
Total	49	7595	201641	327626

In the following, we designed a simple experiment for testing the efficiency of our current annotating method.

We selected two types of corpus sentences as the input of syntactic parser (Zhou, 1997):

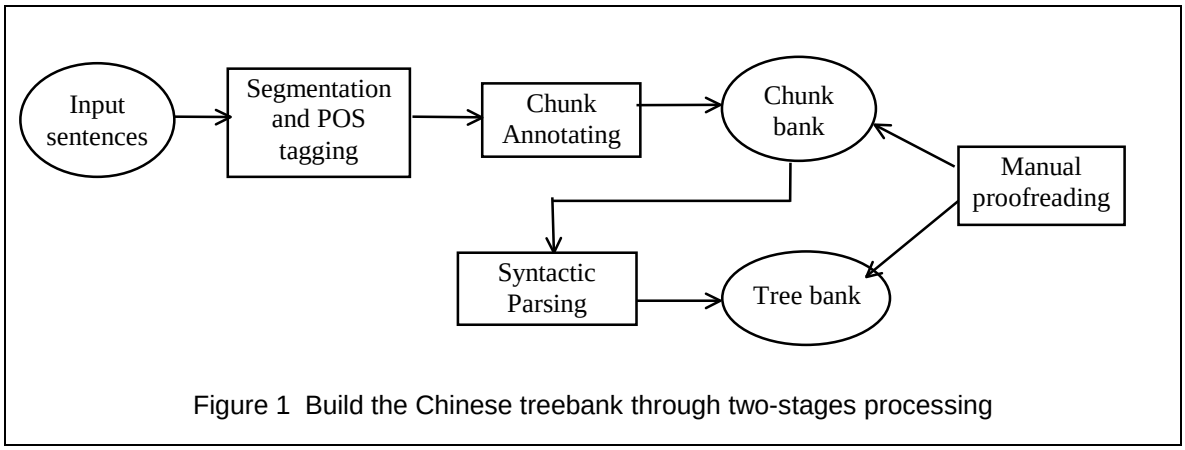

Figure 1 Build the Chinese treebank through two-stages processing

1) Word-split and part-of-speech tagged sentences (STSent);

2) Word-split, part-of-speech tagged and functional chunk annotated sentences (CHSent);

This is a close test. All the knowledge based used by the parser were automatically extracted from THTree, including:

1) Probabilistic Context-Free Grammar rules, which can be used for overall disambiguation;

2) Structure Preference Relation rules (Zhou and Ren, 2001), which can be used for local context disambiguation;

Table 7 Experimental results of parser with different input

Input	Labelled Precision	Labelled Recall	Crossing Brackets
STSent	76.7%	78.5%	3.04
CHSent	89.5%	89.2%	1.17

The experimental results in Table 7 shows that functional chunk information bring in great improvement in parsing performance: the labeled precision an recall of the syntactic parser increase about 13%, and the average number of crossing brackets in a sentence is reduced from 3.04 to 1.17. Therefore, the workload for manual proofreading will be greatly reduced.

6. Conclusions and Future Work

The construction of a large-scale linguistically annotated corpus is a long-term and arduous work, where suitable human-machine collaboration is inevitable and retains a central role.

The paper proposed a new syntactic annotation scheme --- functional chunk, representing information about grammatical relations between sentence-level predicates and their arguments. We made a tentative step to find suitable trade-off between annotation cost and description capability. Under this scheme, we built a Chinese chunk bank with about two million Chinese characters through manual annotation and proofreading, and developed some learned models for automatically annotating fresh text with functional chunks. Based on the top-down characteristic of functional chunk scheme, We proposed a two-stages approach to build Chinese tree bank on the top of chunk bank, and gave some experimental results of chunk-based syntactic parser to show the advantage of functional chunk for parsing performance increase. The small-size treebank building experiment on the Chinese text with about 200,000 words have shown the feasibility of this method.

In the future, we plan to give impetus to the above research work in the following directions:

1) Automatically annotate (manually proofread if possible) the constituent type (NP, VP, etc.) and head word information for each functional chunk, and build a new version of chunk bank.

2) Extract useful lexical collocations from new chunk bank and apply them in the partial parsing models to improve parsing performance.

3) Build a 1,000,000 words Chinese treebank annotated with syntactic constituent and function information on the top of current chunk bank.

7. Acknowledgements

This work was supported by the Chinese National Science Foundation (Grant No. 69903007) and National 973 Foundation (Grant No. 1998030507). We would like to thank Prof. Changning Huang for his original idea in functional chunk annotation, Dr. Weidong Zhan and Dr. Haibo Ren for their good suggests to perfect the chunk annotation stylebook, all the corpus builders for their hard works in annotating the chunk bank, and the anonymous reviewers for their insightful comments and suggestions.

8. References

Abney, S. 1991. Parsing by chunks. In *Principle-Based Parsing*, Kluwer Academic Publishers,.

Alsina. A. 1996. *The Role of Argument Structure in Grammar: Evidence from Romance*. CSLI Lecture Notes No. 62, CSLI Publications: Stanford, California, USA.

Baker, C.F., Fillmore, C.J., and Lowe, J.B. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL'98, Montreal, Canada*, 86-90.

Drabek, E.F. 2001. *Use of Machine Learning Techniques for Partial Parsing of Chinese.* Master thesis, Dept. of Computer Science, Tsinghua University.

Fillmore, C. J. 1982. Frame semantics. In *Linguistics in the Morning Calm*, Hanshin Publishing Co., Seoul, South Korea. 111-137.

Fillmore, C.J., Wooters, C., and Baker, C.F. 2001. Building a Large Lexical Databank Which Provides Deep Semantics. In *Proceedings of the Pacific Asian Conference on Language, Information and Computation. Hong Kong.*

Garside, R., Leech, G., and McEnery, A. (eds.) 1997. *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Longman, London, England

Kaplan, R. and Bresnan, J. 1982. Lexical-Functional Grammar: A Formal System of Representation, In Joan Bresnan (ed.) *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass. 173-281.

Marcus, M.P., Marcinkiewicz, M.A. and Santorini, B. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics,* 19(2), 313-330

Marcus, M., Kim, G., Marcinkiewicz, M.A. et al. 1994. The Penn treebank: annotating predicate argument structure. In *Proceedings of ACL'94 post*.

Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Sang T.K. and Buchholz, S. 2000, Introduction to CoNLL-200 shared task: chunking. In *Proceedings of CoNLL-2000 and LLL*, Lisbon, Portugal. 127-132.

Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. 1997. An Annotation Scheme for Free Word Order Language. In *Proceedings of the Fifth Conference on Applied Natural Language Processing(ANLP-97)*, Washington, DC, USA.

Yu, S.W., Zhu, X.F., Wang, H., Zhang Y.Y. 1998. *The Grammatical Knowledge base of Contemporary Chinese --- A Complete Specification*. Tsinghua University Press,

Zhou, Q. 1997. A Statistics-Based Chinese Parser. In *Proc. of the Fifth Workshop on Very Large Corpora*, 4-15. Bejing, China.

Zhou Q. 2000. *The stylebook for Chinese functional chunk annotation*. Technique report, Dept. of Computer Science, Tsinghua University.

Zhou, Q. and Ren, F.. 2000. Acquisition and applications of structure preference relations in Chinese. *Natural Language Engineering* 6(2): 163-181.

Zhou, Q. and Sun, M.S. 1999. Build a Chinese Treebank as the test suite for Chinese parser. In *Proceedings of the Workshop MAL'99 (Multi-lingual information Processing and Asian Language Processing)*, Beijing, China.