Parsing Chinese by Examples

Oliver Streiter

Academia Sinica, Institute of Information Science

Page 41 ~ 65

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Parsing Chinese by Examples

Oliver Streiter
Academia Sinica, Institute of Information Science,
Nankang, Taipei, Taiwan 115
http://rockey.iis.sinica.edu.tw/oliver

Abstract

This paper presents a Chinese parser which has been derived from the Chinese treebank developed at CKIP, Academia Sinica. Contrary to previous approaches which aim at the conversion of a treebank into a parsers, we do not derive phrase structure rules of any type. Instead, the approach chosen relies on a fuzzy pattern matching strategy in order to extract relevant examples from the treebank. Via a set of adaptation mechanism, these examples are merged and modified so as to produce the best parse for the given set of examples. A detailed description of the parser is provided. The different modules of this parser are evaluated. It is shown that the parser is not only efficient and robust but provides a reasonable level of linguistic adequacy which can be improved upon by restricting the application domain or increasing the number of examples. Competitive approaches are presented and compared to the proposed approach.

1 Introduction

1.1 From a Treebank to a Parser

The Academia Sinica (AS) disposes of rich resources for the automatic treatment of Modern Mandarin Chinese, among them the manually tagged AS-corpus of about 5 Million words (Huang and Chen, 1992) and a lexicon containing about 80.000 words described with respect to their main semantic and syntactic properties (Huang et al., 1995). With the help of a rule-based parser (Chen, 1996) a treebank of manually corrected sentences has been create recently, containing about 40.000 trees (Chen et al., 1999).

This paper describes the attempt to reshape these resources into an example-based parser which ascribes as detailed information to a sentence as can be found in the treebank.

The annotation guidelines for the treebank and a sample of 1000 trees can be found at `http://godel.iis.sinica.edu.tw/CKIP/`. One example tree is reproduced in Fig.1. A BNF of the tree structure is added in Fig.2. While the semantic role labels are almost self-explaining, POS tags are more complex: Tags starting with `N` refer to nouns, starting with `V` refer to verbs, starting with `P` refer to prepositions etc. Additional characters develop a finer classification e.g. `VK1` is a subset of `VK` which is a subset of `V`. The current specifications comprise almost 200 POS tags, 45 phrasal labels and 46 semantic role labels.

```
S(experiencer:Nep:"ta1"|
  Head:VE2:"xia3ng"|
  goal:S(agent:Nap:"ge1ge"|
         epistemics:Dbaa:"yi2di4ng"|
         epistemics:Dbaa:"hui4"|
         Head:VE2:"shuo1"))
```

Figure 1: Example from the treebank: *he think older_brother certainly can speak*

```
<top> ::= <ident> " " <cat> "(" <tree> *( "|" <tree>) ")"
<tree> ::= <role> ":" <cat> ( "(" <tree> *( "|" <tree>) ")" | ":" <Word>)
<ident> ::= {00001, 00002, 00003, ...}
<role> ::= {agent, theme, goal, Head, head, epistemics,  ...}
<cat> ::=  {S, NP, VP, PP, GP, Nep, VE2, Nap, ...}
<Word> ::= {word1, word2, ....}
```

Figure 2: The Backus-Naur Form of the tree structure.

Comparing this treebank to the Chinese Penn-Treebank currently under development (Xia et al., 2000; Xue et al., 2000), we observe the following main differences .

	Penn-Treebank	CKIP Treebank
Chinese character	simplified	traditional (BIG5)
size	3.289 sentences	40.000 sentences
domain	mainly economy	balanced, mixed
average sentence length	27 words	6.3 words (cf. Fig.10)
word to character ratio	1.72	1.87
word to POS ratio	1.10	1.14
POS tags	33	200
syntactic functions	not for every constituent	0
semantic roles	0	46
underlying linguistic theory	GB	idiosyncratic
empty categories	PRO, pro, T(race) ...	not used
branching	deep (almost binary)	flat

Figure 3: The CKIP Treebank compared to the Chinese Penn-Treebank.

1.2 From the "Ultimate Parser" to the Nearest Neighbor (NN)

An example-based parser, in its most simple form, consists of a storing and retrieval function that returns for every learned sentence the associated tree-structure. If for every sentence such a tree-structure can be found this parser would be the "Ultimate Parser" (Sekine and

Grisman, 1995). It would be easy to implement, efficient and easy to maintain. Unfortunately, with large or open subject domains, such an idealized parser cannot perform well, an insight which let Sekine and Grisman return to more conventional parsing approaches. We think, however, that the main idea of the "Ultimate Parser" may be retained if we do not require exact matches of the Input Sentence (ISs) with stored Example Trees (ETs) but content us to retrieve similar ETs and to adapt them according to the kind of mismatch which has occurred.

For this purpose we employ the so-called k-nearest neighbor classifier (k-NN classifier), an approach underlying paradigms as *Example-Based Machine Translation*, *Translation Memories* and *Case-Based Reasoning* (Collins and Cunningham, 1996). According to the latter approach, a new problem is approached by retrieving similar problem formulation from a data-base together with their associated solutions. In a consecutive step the old solutions are *adapted* to the new problem formulation. This approach is considered to result in efficient and qualified problem solutions.

Transferring this approach to the task of parsing, we can identify the problem formulation with an IS to be parsed and the solution with the stored ETs. The adaptation consists in modifying the stored ETs there where the IS does not match the ET.

1.3 From Generalizations to Fuzzy Match

If we accept differences between IS and ET, we may try to find out automatically, or via human reflections what this mismatch can and should be. For example, we could allow pronouns to be matched on proper nouns and vice versa. Predictions of such allowed mismatches are called *generalizations* and are commonly used within memory-based NLP-systems, e.g (Brown, 1999b; Brown, 1999a; Carl, 1999; Streiter, 1999). Due to the pre-definition of such generalizations, they can form part of the indexing system and, as a consequence, matches can be performed efficiently.

In previous experiments (Streiter, 1999) we could show that a still greater degree of flexibility, which cannot or should not be pre-determined may improve the performance. It could be shown that a retrieval requiring strict or generalized matches cannot compete with a fuzzy retrieval that allows for *substitutions* (an incompatible tree slot, e.g. a pronoun matches an adverb) or *deletions* (a tree slot is missing, e.g. no time adverb in the ET) if *adaptations* are applied in order to handle the mismatch. If, for example, an adverb substitutes a subject pronoun, simple frequencies collected during the training of the treebank allow to overwrite the labels associated to the pronoun by the most probable labels associated to this adverb. If a temporal adverb has been deleted, the adaptation inserts this word together with its most likely POS and semantic role.

1.4 From Generalizations to Exact Matches

Generalizations are useful if the set of examples is not large enough to *cover* the input: They help to increase the *coverage*. As could be shown in (Streiter, 2000) however, they may threaten the *reliability*, i.e. the capacity to correctly retrieve ETs once learned in the case the generalizations lead to an ambiguous matching of input and output. In such a case we speak of an *over-generalization*. Unfortunately, *over-generalization* can hardly be avoided as each word behaves differently. Therefore, in many memory-based approaches, generalizations are stored in addition to the original ungeneralized form. If the match with the generalization is ambiguous, the system may resort to the original encoding in order to resolve the ambiguity (Bod and Kaplan, 1998; Daelemans, 1998; Daelemans et al., 1999; Carl, 1999).

2 The parser from a bird eye's view

2.1 Training

The training phase consists of two runs through the treebank. The first run aims at the statistical acquisition of weights which describe how strong a word or its POS is related to a syntactic pattern it occurs in. In a second run, the trees of the treebank including all subtrees are indexed. As indeces we use the words and their POS. Each index is associated with a weight which has been calculated in the first run and points to all trees in which it occurs in. The indexing technique is called an inverted index (Grandy, 1999), a strategy used otherwise for full-text indexing. This technique in not only fast but allows also for the required fuzziness.

2.2 Parsing

Parsing starts with the extraction of k ETs, summing up the weights for all ETs referred to by the words and POSs of the IS. k ETs which accumulate the highest sum of weights are retained. As not every position in an ET has to be matched by an index (word or POS), the match between the IS and the ET may be inexact. A mismatch does not block an already retrieved ET; it only does not increase the total score for the ET.[1]

In the following step the k ETs and the IS are aligned: If, an ET is smaller than the IS (we face a *deletion*), the best mapping of the words of the IS and the positions in the ET has to be found. Words which are not aligned (deleted words) are inserted later on in order to obtain a complete parse.

One way to insert deleted words is the combination (mixing) of the k aligned ETs. For this purpose the aligned trees are segmented into opening phrases, e.g. "$\mathrm{S}_{level=1,head=VE2}$(" ,

[1] This and the following steps are illustrated in detail in (Streiter and Hsueh, 2000) for one parsing example.

words, e.g. "Head$_{level=1,head=VE2}$:VE2:xiang3", and closing phrases, e.g. ")S$_{level=1,head=VE2}$". Such fragments of all ETs are transformed into nodes of a common lattice. Those nodes which are neighbors in a sentence are linked via a transition. The best path through the lattice is generated in the backward pass of a forward-backward two-pass search. As for the alignment mentioned above, the use of the Viterbi-algorithm allows for an efficient implementation of this task. This adaptation strategy will be referred to as *combinatorial adaptation.*

The next adaptation step, referred to as *derivational adaptations* identifies awkward subtrees and replaces them with subtrees obtained by the recursive application of the hole parsing procedure to the words of the subtree. As the phrase to be parsed is shorter than the original IS, we are likely to obtain a better match, unless the chunking into phrases is wrong (as illustrated in Streiter and Hsueh (2000)). The purpose of this recursive call is, similar to the previous adaptation step, to correct badly matched trees and to insert deleted words.

The final *structural adaptations* operate on single words. They handle accidental word mismatches, unknown words, phenomena of type shifting and metonymical extentions of words. They compare the encoding of a word in the retrieved ET with what is know about this word in the lexicon and the learned tree structures. In the case of a mismatch, either the mismatch is maintained (type shifting and metonymies) or the mismatch is attempted to be corrected by assigning the words most likely POS and semantic role (given the POS of the head-word).

3 The Parser in Detail

3.1 Training

3.1.1 Deriving Weights

In a first run we calculate the statistical association between a specific index and a specific tree structure, where an index ($\mathcal{I}$) is a) a word-form ($\mathcal{L}$) (which is Chinese is almost identical to the lexeme), b) its POS ($\mathcal{C}$), c) an abbreviated POS for verbs and nouns ($\mathcal{A}$), and d) a semantic feature ($\mathcal{S}$).

$$\mathcal{I}_x \in \{\mathcal{L}_x, \mathcal{C}_x, \mathcal{A}_x, \mathcal{S}_x\} \quad e.g. \quad \mathcal{I}_{"ge1ge"} \in \{ge1ge, Nap, N, human\} \tag{1}$$

The weight $W^s_{I_{pos}}$ we assign to an index $\mathcal{I}_x$ is its paradigmatic weight, i.e. the relation between the index and the sentence structure. For $\mathcal{L}_x$ and only for $\mathcal{L}_x$ this weight is completed by the syntagmatic weight.

$$W^s_{L_{pos}} = Wsyntag^s_{L_{pos}} + Wparadig^s_{L_{pos}} \tag{2}$$

Syntagmatic Weights The syntagmatic weight describes the contribution of $\mathcal{L}_x$ to the string of a sentence, comparable to the power of a word to trigger a poem or a song in humans. The syntagmatic weight for $\mathcal{L}_x$ in the position *pos* of a sentence s calculated as follows:

$$Wsyntag^{s}_{L_{pos}} = \frac{log(10 + length(s))}{length(s)} \tag{3}$$

The aim of the syntagmatic weight is to enhance the reliability of the parser, i.e. to correctly retrieve learned examples. Without this syntagmatic weight the parser max prefer similar matches with a high probability over exact but unlikely matches. This may already happen with a training corpus as small as 100 sentences. Using this syntagmatic weight, the reliability can be maintained even with very large training corpora.

Paradigmatic Weights The paradigmatic weight of an index $\mathcal{I}$ for a sentence s is calculated indirectly by braking down sentence s in a set of paradigms $\mathcal{P}^s_p$.

$$Wparadig^{s}_{I_{pos}} \sim= Wparadig^{p}_{I_{pos}} \tag{4}$$

How to obtain $Wparadig^{s}_{I_{pos}}$ from $Wparadig^{p}_{I_{pos}}$ will be shown below in Section 3.1.2. What a paradigm is and how it is related to a tree is illustrated in Fig.4.

```
S(experiencer,Head,goal)
S(agent,epistemics,epistemics,Head)
```

Figure 4: Paradigms derived from $sentence_{Fig.1}$.

There is more than one way to calculate the association between an index and the pattern it occurs in. Many association measures are reported to be equivalent when their outcome is transformed onto an ordinal scale (Rijsbergen, 1979) (they are said to be monotone with respect to each other). However, within the current setting we use a ratio scale which has to represent the fact that, for example, two bad matches are better than one good match or not. Whether the different association measures can provide valid information of this type cannot be concluded given the definition of the association measure. We therefore conducted a sequence of experiments reported on in Section 4.2 in which we identified association measures which are more adequate for the task at hand than others.

In order to allow for a better understanding of the weights tested, we develop here, as an example, two weights. The first is derived from the Mutual Information of the paradigm ($\mathcal{P}_p$) and the index in that position of the paradigm ($\mathcal{I}_{pos}$) (with pos = 1,2,3,...) and the second is derived from the conditional probability to have the paradigm ($\mathcal{P}_p$) given the index in that position of the paradigm ($\mathcal{I}_{pos}$).

Be fq_t the total number of observations we make in the training corpus during which we observe a) the joint occurrence of position pos with an index $\mathcal{I}$ ($fq_{i,pos}$), b) a paradigm $\mathcal{P}_p$ (fq_p) and c) the joint occurrence of $\mathcal{I}_{pos}$ in $\mathcal{P}_p$ ($fq_{p,i,pos}$). We can calculate the $MI_{P_p,I_{pos}}$ and $P(p|I_{pos})$ as:

$$Wparadig1^p_{I_{pos}} \sim= MI_{P_p,I_{pos}} = log\frac{P(\mathcal{P}_p \cap \mathcal{I}_{pos})}{P(\mathcal{I}_{pos})\cdot P(\mathcal{P}_p)} = log\frac{\frac{fq_{p,i,pos}}{fq_t}}{\frac{fq_{i,pos}}{fq_t}\cdot\frac{fq_p}{fq_t}} = log\frac{fq_{p,i,pos}\cdot fq_t}{fq_p\cdot fq_{i,pos}} \tag{5}$$

$$Wparadig2^p_{I_{pos}} \sim= P(p|I_{pos}) = \frac{P(\mathcal{P}_p \cap \mathcal{I}_{pos})}{P(\mathcal{I}_{pos})} = \frac{\frac{fq_{p,i,pos}}{fq_t}}{\frac{fq_{i,pos}}{fq_t}} = \frac{fq_{p,i,pos}\cdot fq_t}{fq_{i,pos}} \tag{6}$$

By removing the constant fq_t we simplify the calculus and obtain scores between 0 and 1, so that the logarithmic transformation is no longer necessary.

$$Wparadigm1^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p\cdot fq_{i,pos}} \tag{7}$$

$$Wparadigm2^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_{i,pos}} \tag{8}$$

As can be seen, these two values and those we shall test below differ with respect to the normalization of the joint occurrence $fq_{p,i,pos}$. The measure derived from the Mutual Information provides for a maximal normalization, while the measure derived from the conditional probability does not normalize for fq_p and thus reproduces frequent structures more often than infrequent structures.[2] Both normalize for $fq_{i,pos}$, i.e. reduce the score if the index $\mathcal{I}$ occurred also in different patterns $\overline{p}$. This is may be questionable, especially if $\mathcal{I}$ nevertheless occurred in all or most patterns p. After all, none of the scores tested below seems to be perfect, as none of them obtains the best scores for frequent and infrequent structures.

3.1.2 Indexing

During the indexing of the ETs, we transform the weights we have obtained for the patterns into weights for ETs. In general, the weight of an index $\mathcal{I}$ in sentence s is the weight we calculated for $\mathcal{I}$ in $\mathcal{P}^s_p$. For head-words of embedded phrases (patters), which have received two scorings, one as dependent of the upper level and one as the head of the lower level (e.g.

[2] The conditional probability is frequently used in studies which try to model the human language performance (Hoogweg, 1999; Kaplan, 1996; Bod and Kaplan, 1998). In this light, it seems reasonable to assume that studies using the Mutual Information are competence studies - but do the authors agree upon that?

"shuo1" in Fig.1 is scored once as "Head" at level 2 and once as "goal" at level 1) only the better score is retained.

Inverted indices as used here to retrieve ETs are position independent and as such an optimal indexing mechanism for free word order languages like Russian (although the words of different phrasal levels should not be confused). Word order in Chinese is less free and therefore Chinese may not be well suited for a position-less indexing. In addition, position-less matching requires complex adaptation strategies which have not been investigated until now. Therefore we have to assign the index and its score to a position in the ET. As we intend to match ISs onto ETs which are smaller than the IS, which is not possible if we retain absolute position values (e.g. 3th word of a sentence of 12 words), we map the position onto what we call an index-position *ipos* by transforming the position onto a scale of 10 (e.g. 3/12 = index-position 2). The resulting tuple $< \mathcal{I}, ipos >$ serves as index to the tuple $< trees_s, Wparadig^s_{I_{ipos}} >$.

$$Wparadig^s_{I_{ipos}} = max\ Wparadig^s_{I_{integer(10 \cdot \frac{pos(I)}{length(s)})}} \tag{9}$$

In order to parse fast, even with tens of thousand trees learned, we let the system automatically extend the index with key-words. A $key - word$ is a word which occurs more than 100 times at a given index-position. If a sentence to be indexed contains a key-word, all indices of the sentence are extended by the keyword and its index position. More than one keyword are allowed, extending the index to $< \mathcal{I}, ipos * (, keyword, ipos_{keyword}) >$. More sentences are learned, more words are used as key-word. The additional indexing may improve the performance (not only in time), if the search-space is limited correctly (similar to document clustering in Information Retrieval (Rijsbergen, 1979), or the parsing experiments reported in (Kim and Kim, 1995)), however, if the search-space becomes to small, or, in the worst case, no intersection of the key-words can be found, also negative effects may be expected. It goes without saying that most key-words are high frequent function words like *de* and *le* together with their index-position, but also *shì... de*, *bú shì*, *zài... de... xià* or *zài... de... zhōng* constructions are indexed when training data becomes larger.

3.2 Parsing

3.2.1 NN-Retrieval

Parsing starts with a lexicon look-up which transforms the word of an IS into indices ($\mathcal{I}_x$). The positions of the words are transformed into the index-position as described above. With the help of the resulting index $< \mathcal{I}, ipos * (, keyword, ipos_{keyword} >)$ we access the database and retrieve tuples of $< trees_s, Wparadig^s_{I_{ipos}} >$. One matching index is sufficient in order to

```
sub NN-retrieval {
    KEY-WORD=mkKEY-WORD(WORD);
    for each (WORD,POSITION) {
        INDEX=mkINDEX(WORDi);                    # make the index
        IND-POS=mkINDEX-POS(POSITION);           # make the index-position
        for each(INDEX) {                        # database is looked up
            (TREE,SCORE)=$treeDB{INDEXi,IND-POS,KEY-WORD}
            $NNscore{TREE}+=SCORE }              # accumulate scores per tree
    return best_NN(%NNscore) }                   # return best NN
```

Figure 5: Fuzzy NN-retrieval algorithm.

retrieve an ET. In order to distinguish this match from a better match we sum up the scores for every tree.

3.2.2 Alignment

After the NN-Retrieval k ETs and IS are aligned. As IS may have more words than ET, we have to determine which words of IS matches best with which word in ET and which words are not matched (i.e. deleted during the fuzzy match). In order to solve this task efficiently, dynamic programming strategies can be used: Imagine the words of IS to be plotted on the x-axis and the slots in ET to be plotted on the y-axis. We first determine the "envelope", i.e. all possible combinations of x and y. Within this envelop every cell is filled with a score.

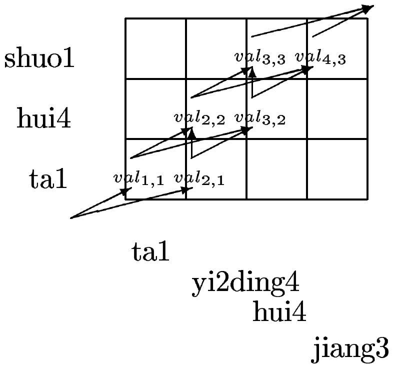


Figure 6: The alignment of IS and ET.

This score consists of the sum of three sub-scores, each being the product of $W_{P_p,I_{pos}}$ defined above and the similarity between $\mathcal{I}_x$ and $\mathcal{I}_y$. For the moment these similarity measures are quite rudimentary.[3]

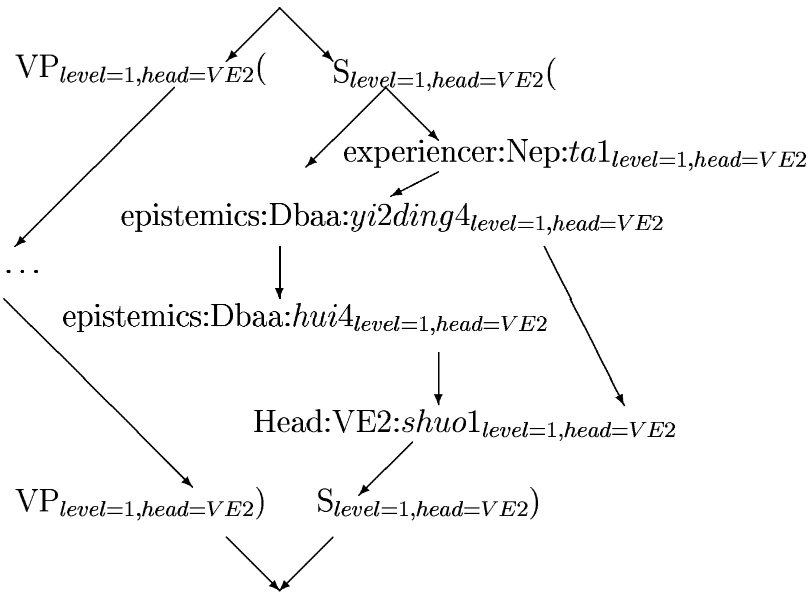
[3] For $\mathcal{L}$, they yield binary values (0,1), for $\mathcal{C}$, the surface similarity of the features is used (e.g. $\mathcal{C}_x$=Nha and $\mathcal{C}_y$=Nhb yields 0.66). For $\mathcal{S}$, some hand-coded rough estimations are used.

$$val_{x,y} = \sum_{\mathcal{I}\in\{\mathcal{L},\mathcal{C},\mathcal{S}\}} (W_{P_p,I_{pos}} \cdot similarity(\mathcal{I}_x,\mathcal{I}_y)) \tag{10}$$

The alignment of the IS and ET in Fig.6 consists of finding a path through this lattice which accumulates most scores. By storing in a hash table partial best paths (e.g. the best path starting from (3,2) to the end), not all possible paths have to be run through, but can be calculated by summing up these partial results (Viterbi-Algorithm cf Ryan and Nudd (1993)). The last step of the alignment consists of the replacement of the words in the ET by the aligned words of the IS. The POS found in the ET is still retained for a while, as this coupling of word and POS allows to identify mismatches.

3.2.3 Combinatorial Adaptation

An almost identical algorithm is used for the *combinatorial adaptation.* The lattice we see in Fig.7 consists not of words as in Fig.6 but of all segments of the k aligned trees. The scores for each segment are those which have been calculated during the alignment procedure and are not reproduced.



VP$_{level=1,head=VE2}$(
S$_{level=1,head=VE2}$(
experiencer:Nep:$ta1_{level=1,head=VE2}$
epistemics:Dbaa:$yi2ding4_{level=1,head=VE2}$
...
epistemics:Dbaa:$hui4_{level=1,head=VE2}$
Head:VE2:$shuo1_{level=1,head=VE2}$
VP$_{level=1,head=VE2}$)
S$_{level=1,head=VE2}$)

Figure 7: A lattice of tree-segments for the *combinatorial adaptation*, merging two partial matches: *yídìng huì shuō* and *tā yídìng shuō* into *tā yídìng huì shuō.*

Looking for the best path trough this lattice allows for the combination of different trees: the insertion of the analysis of one word or phrase of tree A into tree B. In order to obtain coherent tree structures, the level (= the depth) and the head-POS of each segment are annotated on the nodes of the lattice. In addition, the sequence of words of the resulting tree must not contradict that of the IS.

3.2.4 Derivational Adaptations

The next adaptation step identifies awkward subtrees and replaces such subtrees with the re-parse of the words of the awkward subtrees. A phrase is re-parsed if there is a relation between a word and its POS which is not attested in the learned corpus nor in the lexicon. This adaptation allows, similar to the previous adaptation, to correct badly matched trees and to insert deleted words. The largest possible sub-tree is chosen for the re-parsing in order to have the largest possible context for the unmatched word and, secondly, to correct possible errors in the surrounding which may have been caused by this mismatch.

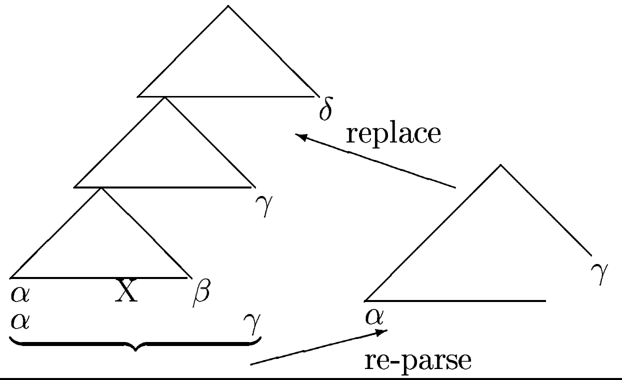


Figure 8: Re-parsing of the subtree form α to γ, triggered by a mismatch in X.

It goes without saying that mismatches at the sentence top level cannot be corrected by this adaptation strategy. The last adaptation, the structural adaptation, is applied to these words.

3.2.5 Structural Adaptations

Structural Adaptations are triggered by the same unattested relations which also trigger the re-parsing, i.e an unknown combination of a word and its POS given the POS of the head-word. If the word is unknown, the assigned POS is maintained. In the future we intend to combine this top-down unknown word guessing with a bottom-up unknown word guessing, so that at this stage two kinds of analysis would have to be conciliated.

If the POS found in the current ET and that found in the lexicon or in the learned trees are very similar, the POS is replaced by the attested known similar POS, however the semantic role of the ET is maintained (assuming that it is compatible with the new POS). If the POS is very different, a new POS and a new semantic role are searched for. POS and semantic role should combine with the word in question and the POS of the head-word. Such information has been collected during the training of the treebank, is however not sensitive to the context (e.g. the role "theme" might be assigned, although already present

in this phrase, cf. Appendix B, sentence 11).
In the future we hope to be able to handle metaphorical extentions of words at this level also. Given the statistical relatedness of a semantic feature and a pattern (sentence) expressed by the association measure, we may retain the semantic feature found in the ET if it is very strongly related to the ET and add a POS of the word of the IS, which is normally not compatible with this semantic feature.

4 Evaluation of Modules

4.1 Main Approach

The Evaluated Unit We have chosen the semantic role relation between a head-word and its dependent words as the entity to be evaluated. The semantic role relation can be though of as turning the phrase structure tree into a dependency tree the arcs of which are labeled with semantic roles. Thus, given the tree in Fig.1, we evaluate the correctness of the triples $< head - word, relation, dependent - word >$, (e.g. $< xia3ng, experiencer, ta1 >$, $< xia3ng, goal, shuo1 >$, $< shuo1, agent, ge1ge >$ etc.

Assuming a hierarchy of increasingly hard evaluation measures (Fig.9), the correct assignment of a semantic role between α and β implies the correct syntactic function, which requires the correct identification of the dependency relation which again requires the bracketing of α and β into one phrase.

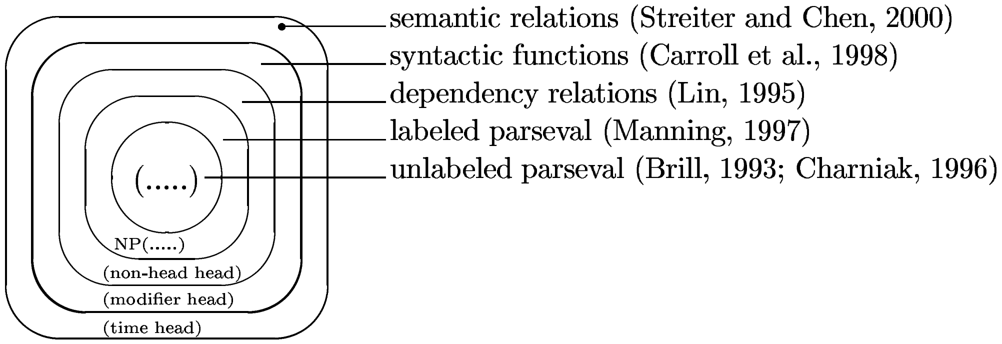

semantic relations (Streiter and Chen, 2000)
syntactic functions (Carroll et al., 1998)
dependency relations (Lin, 1995)
labeled parseval (Manning, 1997)
unlabeled parseval (Brill, 1993; Charniak, 1996)
(.....)
NP(.....)
(non-head head)
(modifier head)
(time head)

Figure 9: The chosen evaluated unit in a hierarchy of increasing hard evaluation units.

The Basic Measures Dividing the number of correctly identified semantic role relations by the number of semantic role relations in the reference corpus, we obtain the *recall*. Dividing the number of correctly identified semantic role relations by the number of semantic role relations in the parsing output we obtain the *precision*. Both scores are combined into the *f-score* via the following formula.

$$f\text{-}score = 2\cdot\frac{precision\cdot recall}{recall + precision} \tag{11}$$

Beside specifying the *recall*, *precision* and *f-score* over the whole test corpus we specify the *f-score* for each sentence length. This allows us to estimate not only the contribution of a specific measure to the overall *f-score* but shows whether long or short sentences take more or less advantage of them.

Derived Measures From the basic measures (recall, precision and f-score) we derive the *coverage* and the *reliability*. The *coverage* describes the performance (in terms of the f-score) on untrained ISs. The *reliability* as defined in (Streiter, 2000; Streiter et al., 2000) is measured using the *f-score* obtained with trained + untrained items. Contrary to the *coverage*, the *reliability* quantifies the ability of the parser to correctly retrieve learned items. To correctly retrieve learned items is not evident for approaches which decompose during training and re-compose during learning, including probabilistic phrase structure grammars, hand-written grammars or even Translation Memories (Carl and Hansen, 1999). *Reliability* values which are not 1 or close to 1 may explain bad performances with large training data. In addition, such systems cannot be trained satisfyingly for a closed domain application which requires 100% correctness. In order to determine the *reliability* we train the training corpus together with reference corpus and test with the same test corpus (hide-and-seek).

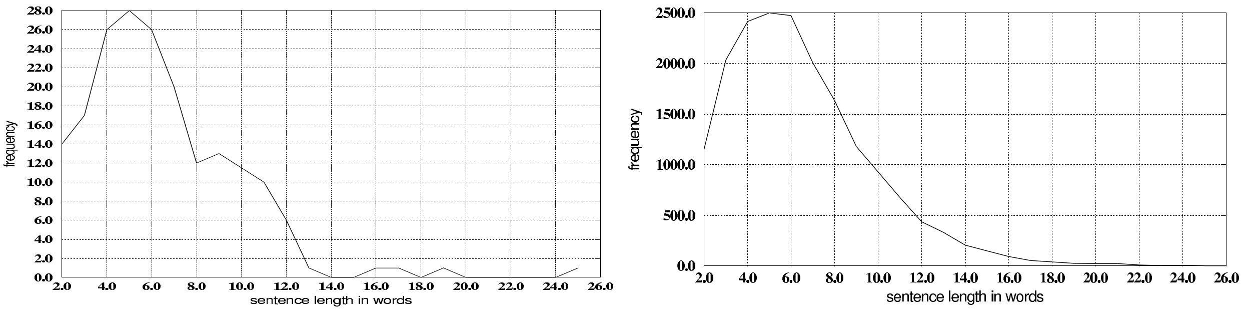


Figure 10: The frequency distribution for the test corpus (left) and the training corpus (right).

Test and Reference Corpus Our point of departure are 20.000 entries of the treebank. As the treebank consists of different articles representing different subject domains and different speech styles, we first have to shuffle the 20.000 sentences into a random order if we do not want to learn the whole set of trees at once, otherwise the system would be trained on text type A and tested on text type A, B and C. After shuffling the treebank using the Fisher-Yates shuffle, we randomly selected 1% of the sentences for testing purposes while the rest is used for training.[4] By this procedure we obtained 197 reference trees with an average

[4] Training and test corpora are kept small due to time constraints in the completion of this contribution.

length of 6.12 words used for automatic evaluation.

In addition we derive two kinds of test corpora (sentences to be parsed) from the reference corpus, one containing the lexical tags and one which does not. The first 50 sentences of the reference corpus are reproduced in the appendix A. The frequency distribution of the test/reference and training corpus are shown in Fig.10.

4.2 Evaluating different Weights

In a first set of experiments we evaluated the NN-retrieval. More precisely, we turned off all adaptation measures (except for the alignment which cannot be dispensed with) and compared the *coverage* with different measures attached to the indices. 3.000 trees have been learned for this experiment. The results in Table 4.2 reveal great differences between the association measures. Neither the competence nor the performance measure yield the best results. Instead those measures which normalize **moderately** for fq_p and $fq_{i,pos}$ perform best. However, no "ultimate" measure can be established as some of them perform better on frequent items and some perform better on infrequent items. The Cosine Coefficient will be used throughout the following experiments.

style	weight	derived from	f-score obtained
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p \cdot fq_{i,pos}}$	mutual information (MI)	0.240
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_{i,pos}}$	conditional probability (CP)	0.280
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p + fq_{i,pos}}$	Dice's coefficient	0.295
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{\sqrt{(fq_p)} \cdot \sqrt{(fq_{i,pos})}}$	Cosine Coefficient	0.300

Figure 11: Comparison of Different Weights for NN-retrieval with 3.000 Training Sentences.

4.3 Contribution of Adaptation: 4.000

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	4.000	alignment only	0.335	0.316	0.325	0.47
coverage	4.000	+ struct. adapt.	0.356	0.336	0.346	0.43
coverage	4.000	+ comb. adapt.	0.340	0.321	0.331	0.51
coverage	4.000	+ recurs. adapt.	0.343	0.324	0.333	0.68
coverage	4.000	all adapt.	0.371	0.349	0.360	0.89
coverage	4.000	tagged input	0.398	0.3376	0.387	0.75
reliability	4.000+197	all adapt.	1	1	1	0.49

Figure 12: The Coverage, reliability and mean parsing time for 4.000 trained sentences with and without adaptation measures.

In order to evaluate the contribution of adaptation strategies we trained 4.000 trees and established the coverage, reliability and the speed of the processing with and without adaptation steps. Results are presented in Fig.4.3 and Fig.13.

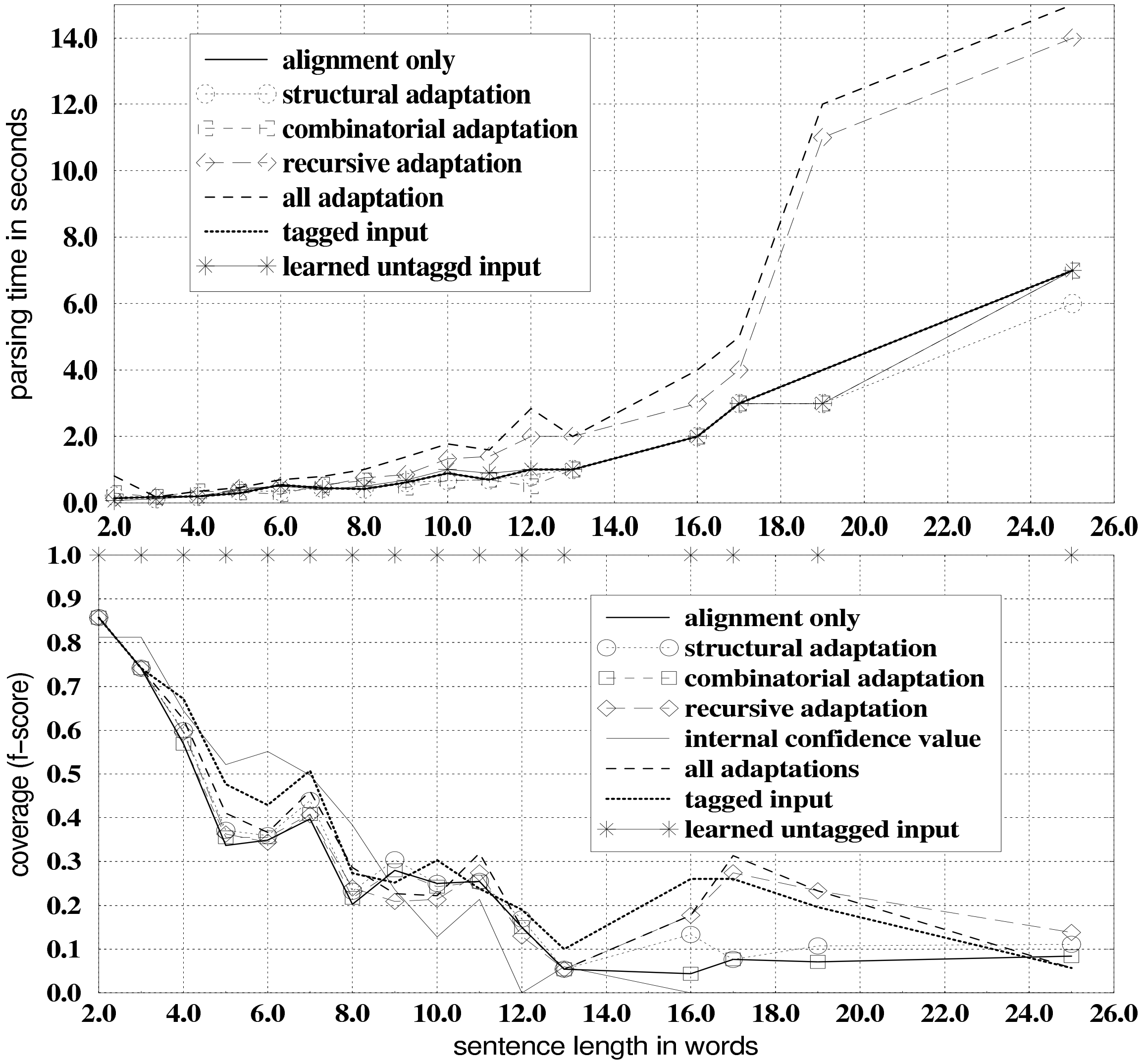


Figure 13: The Impact of Adaptation: Parsing time in seconds and the coverage and reliability measured as f-score for 4.000 training sentences.

As can be seen from the data, the adaptation is time consuming and the gain sometimes very limited. The *structural adaptation* performs best. The run-time behavior of the *recursive adaptation* is almost uncontrollable, as in the worst case, $2+3+4+\ldots+n$ words have to be parsed instead of n words (e.g. 209 words instead of 20), however with long sentences, the parsing results may improve considerably. Only if the quality of the match increases, the re-parsing is no longer that time consuming (or no longer performed), as can be seen from the tagged or learned input. The reliability of the parser is very high, i.e. the system can be perfectly trained for a closed domain application.

4.4 Contribution of Adaptation: 12.000

The above experiments are repeated with a training corpus of 12.000 sentences in order to illustrate the impact of more training data on the behavior of the parser.

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	12.000	alignment only	0.378	0.360	0.369	0.70
coverage	12.000	+ struct. adapt.	0.402	0.383	0.392	0.70
coverage	12.000	+ comb. adapt.	0.383	0.366	0.374	0.56
coverage	12.000	+ recurs. adapt.	0.401	0.378	0.389	0.71
coverage	12.000	all adapt.	0.423	0.4	0.411	0.92
coverage	12.000	tagged input	0.454	0.433	0.443	0.98
reliability	12.000+197	all adapt.	1	1	1	0.75

Figure 14: The Impact of adaptation for 12.000 training sentences.

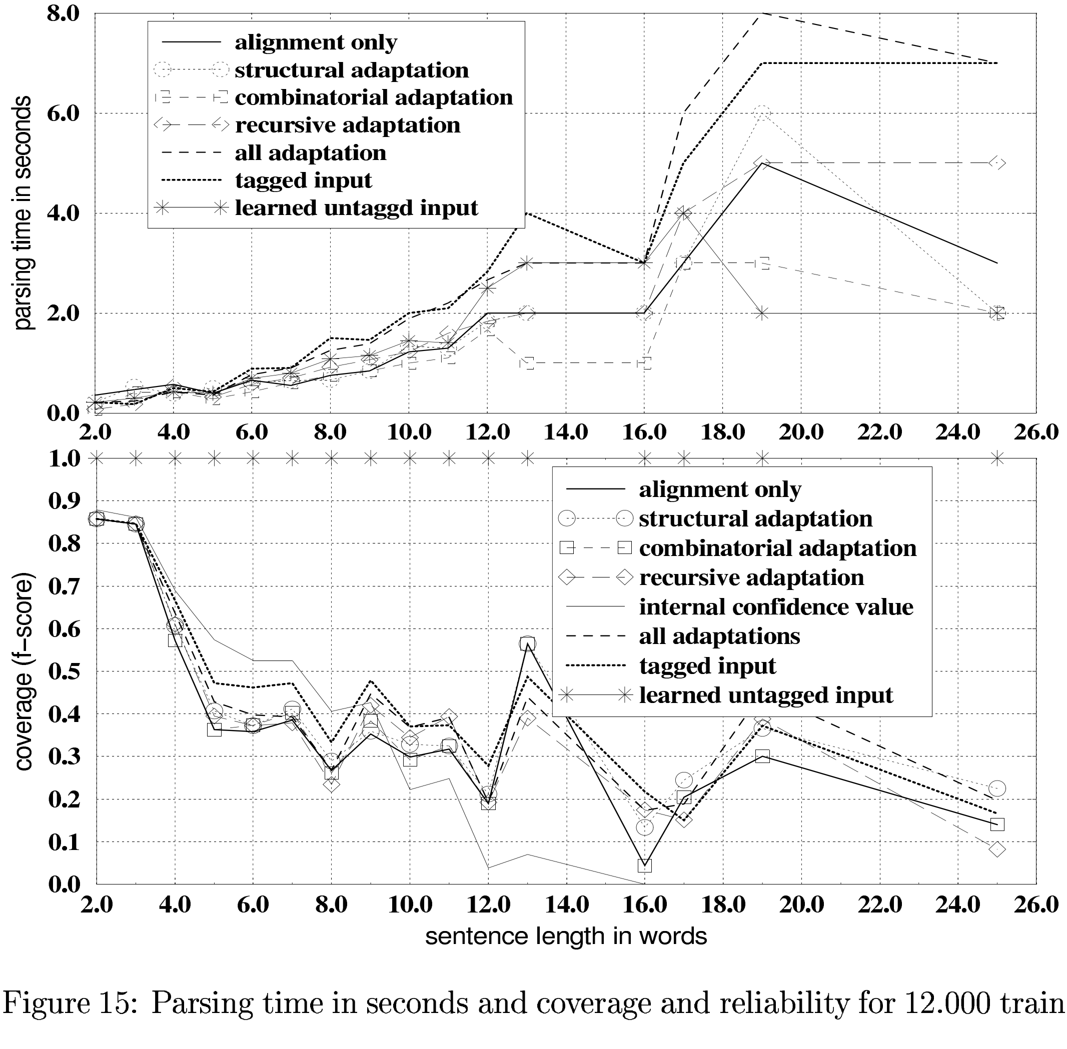


Figure 15: Parsing time in seconds and coverage and reliability for 12.000 training sentences.

Although parsing times increased with more training data, longer sentences require less time as a) the number of keywords grows and b) the recursive adaptation is applied less frequently.

4.5 All Training Data : 19.803

The above experiments are repeated with the complete training corpus. In order to judge the quality of the parse independent from the statistical deviation from the reference corpus refer to Appendix B, where parsing results are shown.

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	19.803	alignment only	0.399	0.389	0.394	1.10
coverage	19.803	+ struct. adapt.	0.423	0.413	0.418	1.13
coverage	19.803	+ comb. adapt.	0.402	0.392	0.397	1.13
coverage	19.803	+ recurs. adapt.	0.408	0.392	0.401	1.01
coverage	19.803	all adapt.	0.428	0.413	0.420	1.78
coverage	19.803	tagged input	0.424	0.413	0.419	1.57
reliability	12.000+197	all adapt.	1	1	1	1.29

Figure 16: The Coverage, reliability and mean parsing time for 19.803 training sentences.

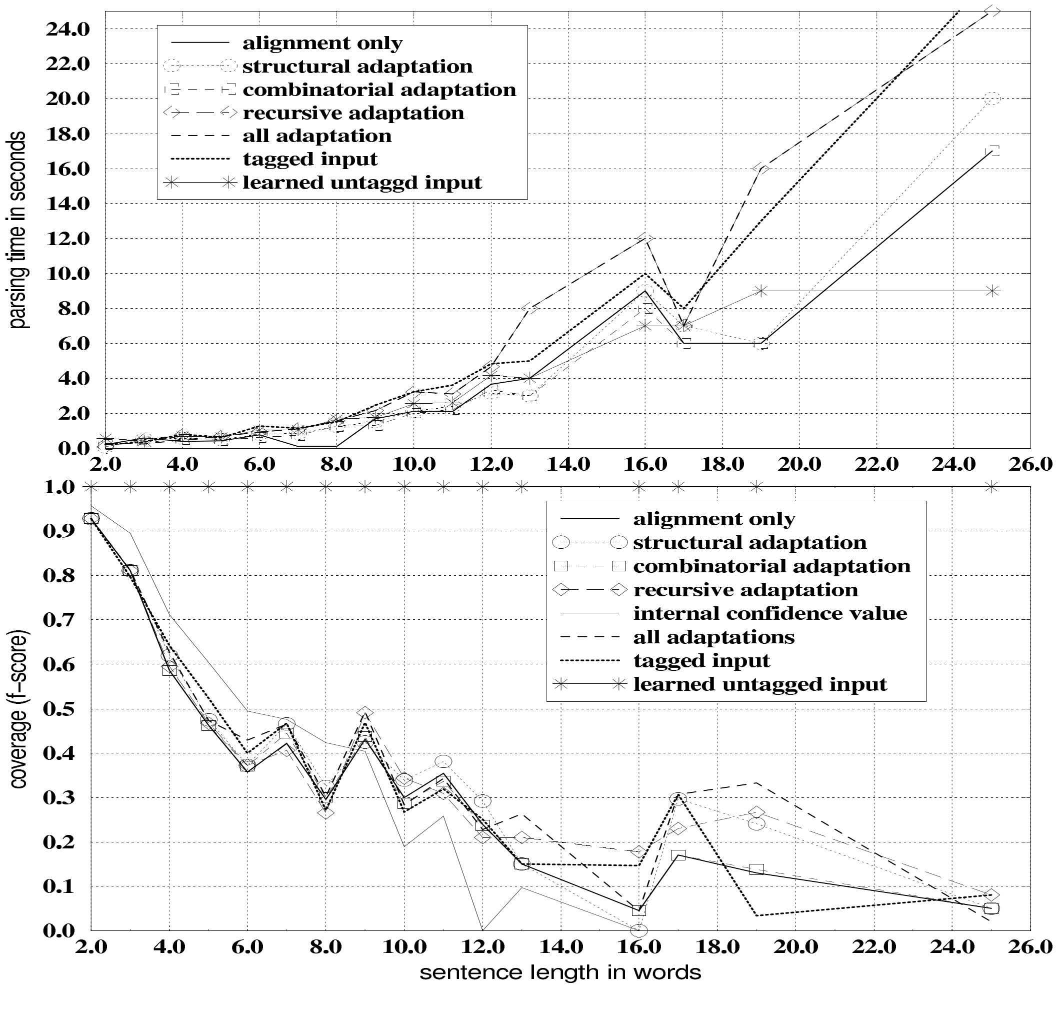


Figure 17: Parsing time in seconds and the coverage and reliability measured as f-score for 19.803 training sentences.

As can be seen in the data plots, the reliability remains high while the coverage increased slowly with more training data. Some sentences however perform worse with more training data than in the previous experiments, showing that the selection of ETs is not optimal and still requires improvements. It may also be the case that sentences are over-indexed, i.e. no intersection of the keywords can be found.

The yet unmentioned internal confidence value based on the score of the NN-retrieval allows for a quite reasonable a-priori estimation of the outcome of the parsing results. As a consequence, this internal confidence value might be used in the future in order to trigger or block specific adaptation mechanism (for sentences they seem or do not seem to be adequate for). It might be used further for the interaction with other parsers or for the automatic acquisition of parsing trees.

5 Related Research

Since the first appearance of treebanks, there have been attempts to use such resources for parsing. A standard approach is to convert the subtrees represented in the treebank into stochastic phrase structure grammars. Such grammars generally outperform hand-written grammars.

Charniak (1996) derives a probabilistic context-free grammar from a 1.000.000 word hand-annotated corpus. The parsing is performed by a probabilistic chart parser. No lexical material except the lexical POS is integrated into the phrase structure rules. Using this strategy about 16.000 rules are derived from the corpus. 10.000 of them have a frequence of 1 and proved irrelevant for the parsing results. Using unlabeled parseval (cf. Fig.9) to evaluate the recall and precision of the parser scores between 80% and 90 % are achieved, depending on the size of the sentence.

The author mentions two drawbacks related to this approach. The first is the lack of lexicalization. Such grammars express only with difficulty relations between lexemes. In most cases, the lexemes are removed during the extraction of rules. The author hopes to obtain better results if more lexical information is integrated into the phrase structure rules. In fact, this claim is confirmed in Bod (1999). A higher degree of lexicalization equally might solve the second problem, i.e. the problem of over-generations. As the parser assigns a parse to almost every combination of POSs, it has difficulties to determine the best sub-parse from the chart with which the parse has to be continued, often resulting in an incorrect or failing parse. Parsing using all sub-parses stored in the chart seems impossible given the high redundancy of the grammar. Lexicalization thus seems to be the method to overcome the over-generation. Improvements to this standard approach are suggested in (Manning, 1997; Manning and Hinrich, 1999).

Tree Adjoining Grammars (TAGs) are equally extracted from treebanks (Xia, 1999; Chen and Vijay-Shanker, 2000) and used for grammar development and testing (Sarkar, 2000; Xia and Martha, 2000). It seems however, that the grammar extraction does not take full advantage of the treebanks, as the trees are split into elementary trees without retaining, in addition to the elementary trees, the unsplit (sub)tree, e.g "eat hot soup with a spoon". That the knowledge contained in these unsplit trees is critical for high quality parsing has been shown repeatedly (Rayner and Christer, 1994; Srivinas and Joshi, 1995; Bod, 1999; Streiter, 2000).

Data-oriented parsing (DOP) (Bod, 1992; Bod and Kaplan, 1998; Hoogweg, 1999) represents a parsing approach which promises to do away with the deficiencies in lexicalization. This approach consists of breaking the learned trees into all possible sub-trees and using all these sub-trees during the parsing. That parse which is obtained most frequently is chosen as final parse. It goes without saying that such an approach, as interesting as it may be, leads to a crazy computational complexity (Manning and Hinrich, 1999). Whether or not this approach becomes tractable by building up random samples of parses (Monte Carlo Parsing) and what the effect on the performance is, is a topic of current research. Practical systems following this approach cannot be expected in the near future.

Approaches which do not involve standard parsers while converting a treebank into a parser are hard to find. An interesting approach is presented by Lepage (1999). Sentences are analyzed with the help of analogy relations. Triples of ETs are extracted from a treebank the sentences to which they belong stand in a relation of analogy to the IS. The parse of the sentence is supposed to be the analogous tree derived by this triple. Although this method is extremely elegant, the system does not know on the basis of which ETs the analogy has to be made. As a consequence, a sentence may produce, depending on the size of the treebank almost as many parses as there are ETs. The selection of the best parse and the question whether this parse is a correct parse, as well as the efficiency of the algorithm are yet unsolved problems.

5.1 Summary and Conclusions

In this paper we present a approach to the analysis of natural language which does not follow any traditional parsing approach. Differently from standard approaches we do not attempt a local identification of forms, functions and meaning. We try instead to identify large sentence patterns by comparing the input sentence with examples from the treebank and concentrate on individual word meanings after the global structure has been established. We claim that this approach can maintain the highest degree of lexicalization while remaining efficient.

In order to identify the main sentence patterns, the system makes use of a fuzzy matching strategy. The inexact matches are worked over by a set of adaptation strategies. Thus, instead of considering mismatches to be harmful exceptions, they constitute a fundamental part of our approach, resulting in an extremely robust and adaptive system (there is no sentence which cannot be parsed).

Although parsing results are not especially good for long sentences in open domains, perfect parsing results are achieved in closed domains, independent of the size of the domain and independent of the size of the tree. This is not possible for any approach which during parsing re-composes subtrees, even for small domain areas.

The evaluation of different adaptation strategies has shown that the structural and recursive adaptation should be retained, as they improve the parsing results in open domains significantly. The combinatorial adaptation does not seem to be as performing. However, the combinatorial adaptation may provide a good interface for the cooperation with other (still hypothetical) parsers running in parallel. Such parsers running in parallel could fill the lattice with (partial) results until the example-based parser has completed the lattice and starts the evaluation.

Future work is manifold and overwhelming. First of all, the coverage has to be increased by optimizing scores, parameters and thresholds. Secondly, we intend to investigate experimentally the usefulness of this parser for sublanguage applications
Our claim that metonymies and maybe metaphors may be treated in this framework still awaits an experimental confirmation. We further intend to apply this parsing approach to a free word-order language, using the Russian corpus developed at IPPI (Boguslavskij et al., 2000). The integration of a bottom-up unknown word classifier as well as the cooperation with other parsers complete the set of future tasks.

6 Resources

The parser is written in Perl and has been developed under Linux. With minor changes the parser may run also under commercial operating systems. Experiments have been performed with 200 MHz CPU. The parser is a multi-tasking server which can be accessed via the TCP/IP. A demo-system and a download of a parsing-client can be found under http:://rockey.iis.sinica.edu.tw/oliver/parser.

Acknowledgment This work would not have been possible without the group of diligent tree-bank writers of CKIP. Special thanks to Chiu Chih-Ming, Luo Chi-Ching and Tsai Pi-Fang. Chen Laoshi has been a constant source of welcome objections and suggestions. Xiao-An finally supported me mentally. Without her patience and courage this work would have been neither started nor finished.

References

Rens Bod and Ronald M. Kaplan. 1998. A probabilistic corpus-driven model for lexial-functional analysis. In *COLING-ACL'98*. URL http://www/lfg.standord.edu/lfg/lfg-dop.

Rens Bod. 1992. Data oriented parsing (dop). In *COLING'92*.

Rens Bod. 1999. Extracting stochastic grammars from treebanks. In *Journées ATALA sur les Corpus annotés pour la syntaxe*. Talana, Paris VII.

I.M. Boguslavskij, Grigoreva S.A. Grigorev, N.V., L.L. Iomdin, M.V. Kreidlin, V.Z. Sannikov, and N.E. Frid. 2000. Annotierovannyj korpus russkix tekstov: koncepcija, instrumenty razmetki, tipy informacii. In *Proceedings of the Dialogue'2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 41–47, Prodvino, Russia.

Eric Brill. 1993. Automatic grammar induction and parsing free text: A transformation-based approach. In *31rd Annual Meeting of the ACL*. URL http://www.cs.jhu.edu/ brill/acadpubs.html.

Ralf D. Brown. 1999a. Adding linguistic knowledge to a lexical example-based translation system. In *Theoretical and Methodological Issues in Machine Translation-99*. URL http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ralf/pub/WWW.

Ralf D. Brown. 1999b. Generalized example-based machine translation. URL http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ralf/pub/WWW/ebmt.html, April.

Michael Carl and Silvia Hansen. 1999. Linking translation memories with example-based machine translation. In *Proceedings of the MT-Summit'99*, Singapore. URL http://www.iai.uni-sb.de/.

Michael Carl. 1999. Inducing translation templates for example-based machine translation. In *Proceedings of the MT-Summit'99*, Singapore. URL http://www.iai.uni-sb.de/.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *First International Conference on Language Resources & Evaluation, Granada, Spain*. URL http://www.cogs.susx.ac.uk/lab/nlp/carroll/carroll.html#cbs98.

Eugene Charniak. 1996. Tree-bank grammars. In *13th National Conference on Artificial Intelligence, AAAI-96*, pages 1031–1036.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGS from the Penn Treebank. In *IWPT 2000, Sixth International Workshop on Parsing Technologies*, Trento, Italy, 23-25 February.

Keh-Jiann Chen, Chi-Ching Luo, Zhao-Ming Gao, Ming-Chung Chang, Feng-Yi Chen, and Chao-Jan Chen. 1999. The CKIP Chinese Treebank. In *Journées ATALA sur les Corpus annotés pour la syntaxe*. Talana, Paris VII.

Keh-Jiann Chen. 1996. A model for robust Chinese parser. *Computational Linguistics and Chinese Language*, 1(1):183–204. URL http://rocling.iis.sinica.edu.tw/ROCLING/publish/clclp/v1/a6.htm.

Bróna Collins and Pádraig Cunningham. 1996. Adaptation guided retrieval in EBMT: A case-based approach to machine translation. In *Advances in Case-Based Reasoning*, LNAI, pages 91–104. Springer.

Walter Daelemans, Antal Van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, special issue on natural language learning(34):11–43. URL http://ilk.kub.nl/papers.html.

Walter Daelemans. 1998. Abstraction is Harmful in Language Learning. In *Proceedings of NeMLaP (New Methods in Language Processing)*, pages 1–2, Sydney, January. URL http://ilk.kub.nl/papers.html.

Cheryl Grandy. 1999. The art of indexing. A white paper, Dynamic Systems Corporation, October. URL http://www.disc.com/home/artindex.html.

Lars Hoogweg. 1999. A data-oriented approach to lexicalize tree adjoining grammars. Report, University of Amsterdam, URL http://gene.wins.uva.nl/ lhoogweg/dop.html.

Chu-Ren Huang and Keh-Jiann Chen. 1992. A Chinese corpus for linguistics research. In *COLING'92*, Nantes.

Chu-Ren Huang, Keh-Jiann Chen, Li-ping Chang, and Hui-li Hsu. 1995. Zhōng yāng yán jīu yuàn pīng héng yǔ liào kù jiǎn jiè. In *Proceedings of ROCLING VIII.*

Ronald M. Kaplan. 1996. A probabilistic approach to LFG. keynote lecture held at LFG-workshop. Grenoble. ftp://ftp-lfg.stanford.edu/pub/lfg/.

Sung Dong Kim and Yung Taek Kim. 1995. Sentence analysis using pattern matching in English-Korean Machine Translation. In *International Conference on Computer Processing of Oriental Languages*, pages 199–206, Honolulu.

Yves Lepage. 1999. Open set experiments with direct analysis by analogy. In *Proceedings NLPRS'99 (Natural Language Processing Pacific Rim Symposium)*, pages 363–368, Beijing.

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the IJCAI-95*, pages 1420–1425, Montreal, Canada.

Christoher D. Manning and Schütz Hinrich. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, London. URL http://www.sultry.arts.usyd.edu.au/cmanning/.

Christoher D. Manning. 1997. Probabilistic models of language structure. URL http://www.sultry.arts.usyd.edu.au/cmanning/LAE, May.

Manny Rayner and Samuelsson Christer. 1994. Corpus-based grammar specification for fast analysis. In Agnas et al., editor, *Spoken Language Translator: First Year Report*, SRI Technical Report CRC-043, pg. 41-54. URL http://www.cam.sri.com.

C.J. Rijsbergen. 1979. *Information Retrieval.* Butterworth, London. URL http://www.dcs.ac.uk/Keith/Preface.html.

Matthew S. Ryan and Grahan R. Nudd. 1993. The Viterbi Algorithm. Research report cs-rr-238, Department of Computer Science, University of Warwick, Coventry,UK, URL http://www.dcs.warwick.ac.uk/pub/reports/rr/238.html.

Anoop Sarkar. 2000. Practical experiments in parsing using tree adjoining grammars. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 193–198, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Satoshi Sekine and Ralph Grisman. 1995. A corpus-based probabilistic grammar with only two non-terminals. In *Forth International Workshop on Parsing Technology*, Prague, September. URL http://www.cs.nyu.edu/cs/projects/proteus/sekine.

B. Srivinas and Aravind K. Joshi. 1995. Some novel applications of explanation-based learning to parsing lexicalized tree-adjoining grammars. cmp-lg archive 9505023.

Oliver Streiter and Keh-Jiann Chen. 2000. Experiments in example-based parsing. In *Proceedings of the Dialogue 2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 410–418, Prodvino, Russia. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter and Pei-Yun Hsueh. 2000. A case-study in example-based parsing. In *ICCLC2000, 2000 International Conference on Chinese Language Computing*, Chicago. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter, Michael Carl, and Leonid L. Iomdin. 2000. A virtual translation machine for hybrid machine translation. In *Proceedings of the Dialogue'2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 382–394, Prodvino, Russia. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter. 1999. Parsing Chinese with randomly generalized examples. In *NLPRS'99 Workshop on Multi-lingual Information Processing and Asian Language Processing (Natural Language Processing Pacific Rim Symposium)*, pages 114–119, Beijing, November. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

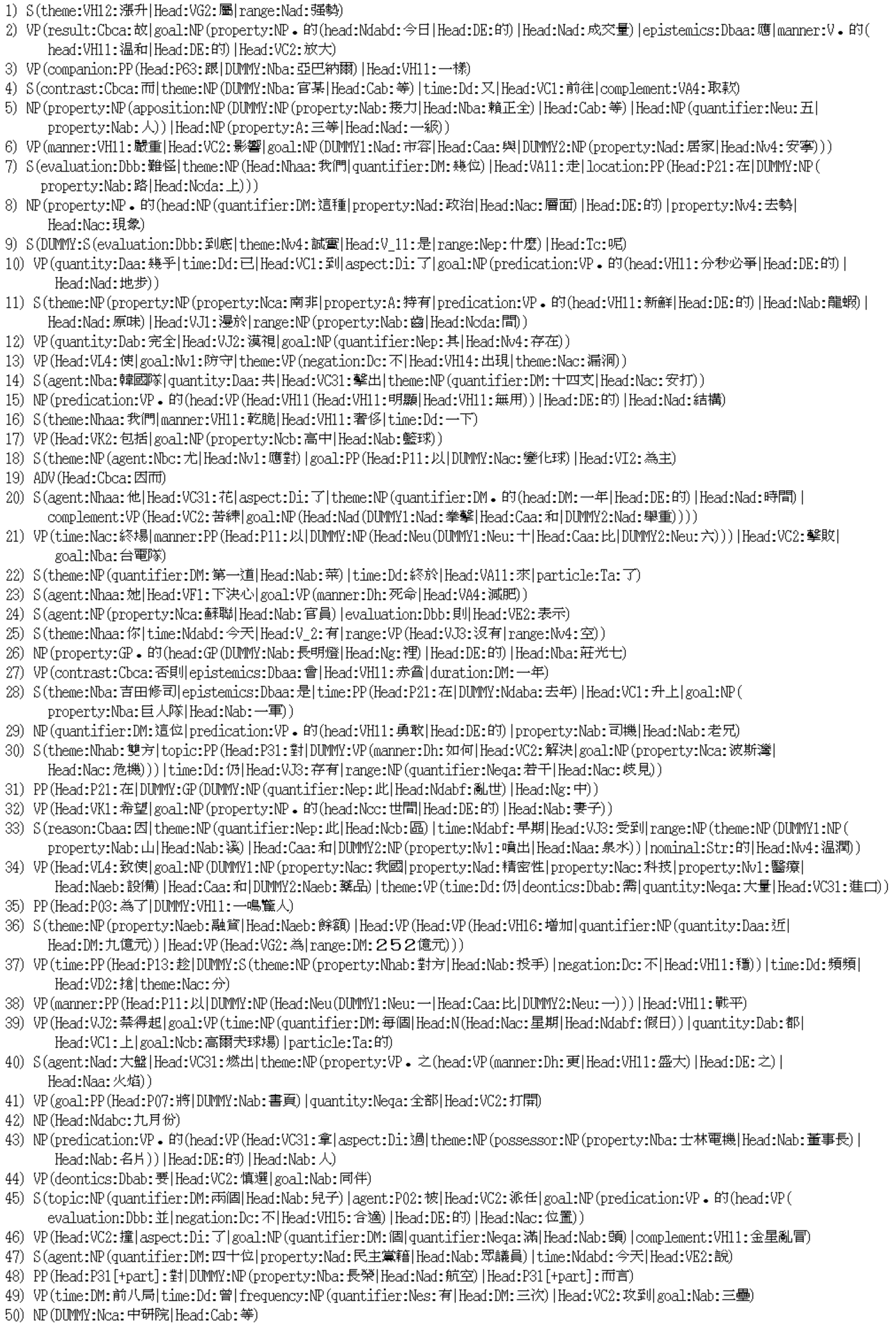
Oliver Streiter. 2000. Reliability in example-based parsing. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 257–260, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Fei Xia and Palmer Martha. 2000. Comparing and integrating tree adjoining grammars. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 265–268, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Fei Xia, Palmer Martha, Xue Nianwen, Okurowski Mary Ellen, Kovarik John, Chiou Fu-Dong, Huang Shizhe, Kroch Tony, and Marcus Mitch. 2000. Developing guidelines and ensuring consistency for Chinese text annotation. In *Second International Conference on Language Resources & Evaluation*, Athenes, Greece.

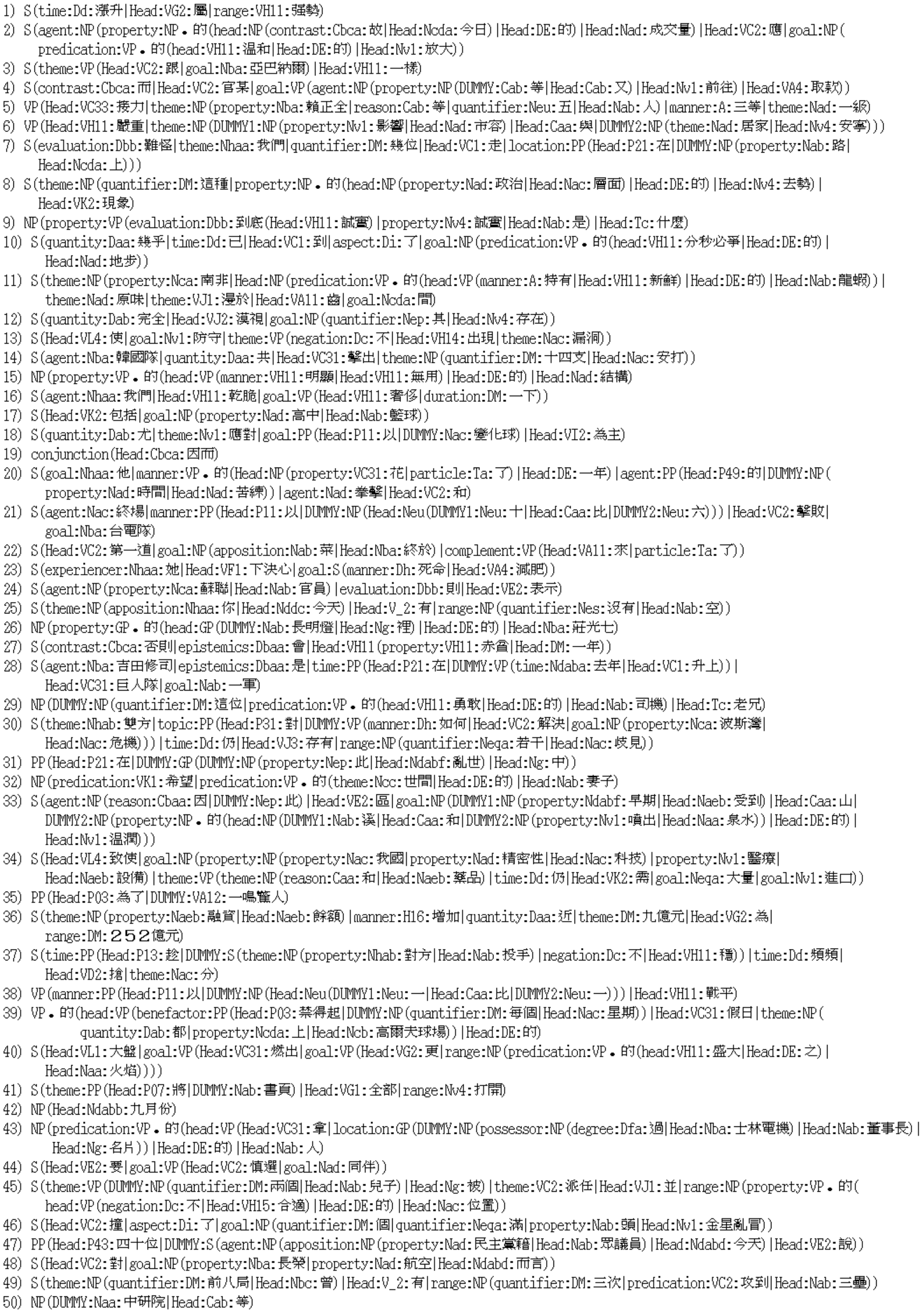
Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings NLPRS'99 (Natural Language Processing Pacific Rim Symposium)*, pages 398–403, Beijing.

Nianwen Xue, Huang Shizhe Xia, Fei, and Anthony Kroch. 2000. The bracketing guidelines for the Penn Chinese Treebank (draft II). Technical report, University of Pennsylvania, URL http://www.ldc.upenn-ctb/.



```
1) S(theme:VH12:漲升|Head:VG2:屬|range:Nad:强勢)
2) VP(result:Cbca:故|goal:NP(property:NP•的(head:Ndabd:今日|Head:DE:的)|Head:Nad:成交量)|epistemics:Dbaa:應|manner:V•的(
     head:VH11:温和|Head:DE:的)|Head:VC2:放大)
3) VP(companion:PP(Head:P63:跟|DUMMY:Nba:亞巴納爾)|Head:VH11:一樣)
4) S(contrast:Cbca:而|theme:NP(DUMMY:Nba:官某|Head:Cab:等)|time:Dd:又|Head:VC1:前往|complement:VA4:取款)
5) NP(property:NP(apposition:NP(DUMMY:NP(property:Nab:接力|Head:Nba:賴正全)|Head:Cab:等)|Head:NP(quantifier:Neu:五|
     property:Nab:人))|Head:NP(property:A:三等|Head:Nad:一級))
6) VP(manner:VH11:嚴重|Head:VC2:影響|goal:NP(DUMMY1:Nad:市容|Head:Caa:與|DUMMY2:NP(property:Nad:居家|Head:Nv4:安寧)))
7) S(evaluation:Dbb:難怪|theme:NP(Head:Nhaa:我們|quantifier:DM:幾位)|Head:VA11:走|location:PP(Head:P21:在|DUMMY:NP(
    property:Nab:路|Head:Ncda:上)))
8) NP(property:NP•的(head:NP(quantifier:DM:這種|property:Nad:政治|Head:Nac:層面)|Head:DE:的)|property:Nv4:去勢|
     Head:Nac:現象)
9) S(DUMMY:S(evaluation:Dbb:到底|theme:Nv4:誠實|Head:V_11:是|range:Nep:什麼)|Head:Tc:呢)
10) VP(quantity:Daa:幾乎|time:Dd:已|Head:VC1:到|aspect:Di:了|goal:NP(predication:VP•的(head:VH11:分秒必爭|Head:DE:的)|
     Head:Nad:地步))
11) S(theme:NP(property:NP(property:Nca:南非|property:A:特有|predication:VP•的(head:VH11:新鮮|Head:DE:的)|Head:Nab:龍蝦)|
     Head:Nad:原味)|Head:VJ1:漫於|range:NP(property:Nab:齒|Head:Ncda:間))
12) VP(quantity:Dab:完全|Head:VJ2:漠視|goal:NP(quantifier:Nep:其|Head:Nv4:存在))
13) VP(Head:VL4:使|goal:Nv1:防守|theme:VP(negation:Dc:不|Head:VH14:出現|theme:Nac:漏洞))
14) S(agent:Nba:韓國隊|quantity:Daa:共|Head:VC31:擊出|theme:NP(quantifier:DM:十四支|Head:Nac:安打))
15) NP(predication:VP•的(head:VP(Head:VH11(Head:VH11:明顯|Head:VH11:無用))|Head:DE:的)|Head:Nad:結構)
16) S(theme:Nhaa:我們|manner:VH11:乾脆|Head:VH11:奢侈|time:Dd:一下)
17) VP(Head:VK2:包括|goal:NP(property:Ncb:高中|Head:Nab:籃球))
18) S(theme:NP(agent:Nbc:尤|Head:Nv1:應對)|goal:PP(Head:P11:以|DUMMY:Nac:變化球)|Head:VI2:為主)
19) ADV(Head:Cbca:因而)
20) S(agent:Nhaa:他|Head:VC31:花|aspect:Di:了|theme:NP(quantifier:DM•的(head:DM:一年|Head:DE:的)|Head:Nad:時間)|
     complement:VP(Head:VC2:苦練|goal:NP(Head:Nad(DUMMY1:Nad:拳擊|Head:Caa:和|DUMMY2:Nad:舉重))))
21) VP(time:Nac:終場|manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:十|Head:Caa:比|DUMMY2:Neu:六)))|Head:VC2:擊敗|
     goal:Nba:台電隊)
22) S(theme:NP(quantifier:DM:第一道|Head:Nab:菜)|time:Dd:終於|Head:VA11:來|particle:Ta:了)
23) S(agent:Nhaa:她|Head:VF1:下決心|goal:VP(manner:Dh:死命|Head:VA4:減肥))
24) S(agent:NP(property:Nca:蘇聯|Head:Nab:官員)|evaluation:Dbb:則|Head:VE2:表示)
25) S(theme:Nhaa:你|time:Ndabd:今天|Head:V_2:有|range:VP(Head:VJ3:沒有|range:Nv4:空))
26) NP(property:GP•的(head:GP(DUMMY:Nab:長明燈|Head:Ng:裡)|Head:DE:的)|Head:Nba:莊光七)
27) VP(contrast:Cbca:否則|epistemics:Dbaa:會|Head:VH11:赤貧|duration:DM:一年)
28) S(theme:Nba:吉田修司|epistemics:Dbaa:是|time:PP(Head:P21:在|DUMMY:Ndaba:去年)|Head:VC1:升上|goal:NP(
     property:Nba:巨人隊|Head:Nab:一軍))
29) NP(quantifier:DM:這位|predication:VP•的(head:VH11:勇敢|Head:DE:的)|property:Nab:司機|Head:Nab:老兄)
30) S(theme:Nhab:雙方|topic:PP(Head:P31:對|DUMMY:VP(manner:Dh:如何|Head:VC2:解決|goal:NP(property:Nca:波斯灣|
     Head:Nac:危機)))|time:Dd:仍|Head:VJ3:存有|range:NP(quantifier:Neqa:若干|Head:Nac:歧見))
31) PP(Head:P21:在|DUMMY:GP(DUMMY:NP(quantifier:Nep:此|Head:Ndabf:亂世)|Head:Ng:中))
32) VP(Head:VK1:希望|goal:NP(property:NP•的(head:Ncc:世間|Head:DE:的)|Head:Nab:妻子))
33) S(reason:Cbaa:因|theme:NP(quantifier:Nep:此|Head:Ncb:區)|time:Ndabf:早期|Head:VJ3:受到|range:NP(theme:NP(DUMMY1:NP(
     property:Nab:山|Head:Nab:溪)|Head:Caa:和|DUMMY2:NP(property:Nv1:噴出|Head:Naa:泉水))|nominal:Str:的|Head:Nv4:温潤))
34) VP(Head:VL4:致使|goal:NP(DUMMY1:NP(property:Nac:我國|property:Nad:精密性|property:Nac:科技|property:Nv1:醫療|
     Head:Naeb:設備)|Head:Caa:和|DUMMY2:Naeb:藥品)|theme:VP(time:Dd:仍|deontics:Dbab:需|quantity:Neqa:大量|Head:VC31:進口))
35) PP(Head:P03:為了|DUMMY:VH11:一鳴驚人)
36) S(theme:NP(property:Naeb:融資|Head:Naeb:餘額)|Head:VP(Head:VP(Head:VH16:增加|quantifier:NP(quantity:Daa:近|
     Head:DM:九億元))|Head:VP(Head:VG2:為|range:DM:２５２億元)))
37) VP(time:PP(Head:P13:趁|DUMMY:S(theme:NP(property:Nhab:對方|Head:Nab:投手)|negation:Dc:不|Head:VH11:穩))|time:Dd:頻頻|
     Head:VD2:搶|theme:Nac:分)
38) VP(manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:一|Head:Caa:比|DUMMY2:Neu:一)))|Head:VH11:戰平)
39) VP(Head:VJ2:禁得起|goal:VP(time:NP(quantifier:DM:每個|Head:N(Head:Nac:星期|Head:Ndabf:假日))|quantity:Dab:都|
     Head:VC1:上|goal:Ncb:高爾夫球場)|particle:Ta:的)
40) S(agent:Nad:大盤|Head:VC31:燃出|theme:NP(property:VP•之(head:VP(manner:Dh:更|Head:VH11:盛大)|Head:DE:之)|
     Head:Naa:火焰))
41) VP(goal:PP(Head:P07:將|DUMMY:Nab:書頁)|quantity:Neqa:全部|Head:VC2:打開)
42) NP(Head:Ndabc:九月份)
43) NP(predication:VP•的(head:VP(Head:VC31:拿|aspect:Di:過|theme:NP(possessor:NP(property:Nba:士林電機|Head:Nab:董事長)|
     Head:Nab:名片))|Head:DE:的)|Head:Nab:人)
44) VP(deontics:Dbab:要|Head:VC2:慎選|goal:Nab:同伴)
45) S(topic:NP(quantifier:DM:兩個|Head:Nab:兒子)|agent:P02:被|Head:VC2:派任|goal:NP(predication:VP•的(head:VP(
     evaluation:Dbb:並|negation:Dc:不|Head:VH15:合適)|Head:DE:的)|Head:Nac:位置))
46) VP(Head:VC2:撞|aspect:Di:了|goal:NP(quantifier:DM:個|quantifier:Neqa:滿|Head:Nab:頭)|complement:VH11:金星亂冒)
47) S(agent:NP(quantifier:DM:四十位|property:Nad:民主黨籍|Head:Nab:眾議員)|time:Ndabd:今天|Head:VE2:說)
48) PP(Head:P31[+part]:對|DUMMY:NP(property:Nba:長榮|Head:Nad:航空)|Head:P31[+part]:而言)
49) VP(time:DM:前八局|time:Dd:曾|frequency:NP(quantifier:Nes:有|Head:DM:三次)|Head:VC2:攻到|goal:Nab:三壘)
50) NP(DUMMY:Nca:中研院|Head:Cab:等)
```

Figure 18: Appendix A: The 50 first trees of the reference corpus.



```
1) S(time:Dd:漲升|Head:VG2:屬|range:VH11:强勢)
2) S(agent:NP(property:NP•的(head:NP(contrast:Cbca:故|Head:Ncda:今日)|Head:DE:的)|Head:Nad:成交量)|Head:VC2:應|goal:NP(
     predication:VP•的(head:VH11:温和|Head:DE:的)|Head:Nv1:放大))
3) S(theme:VP(Head:VC2:跟|goal:Nba:亞巴納爾)|Head:VH11:一樣)
4) S(contrast:Cbca:而|Head:VC2:官某|goal:VP(agent:NP(property:NP(DUMMY:Cab:等|Head:Cab:又)|Head:Nv1:前往)|Head:VA4:取款))
5) VP(Head:VC33:接力|theme:NP(property:Nba:賴正全|reason:Cab:等|quantifier:Neu:五|Head:Nab:人)|manner:A:三等|theme:Nad:一級)
6) VP(Head:VH11:嚴重|theme:NP(DUMMY1:NP(property:Nv1:影響|Head:Nad:市容)|Head:Caa:與|DUMMY2:NP(theme:Nad:居家|Head:Nv4:安寧)))
7) S(evaluation:Dbb:難怪|theme:Nhaa:我們|quantifier:DM:幾位|Head:VC1:走|location:PP(Head:P21:在|DUMMY:NP(property:Nab:路|
     Head:Ncda:上)))
8) S(theme:NP(quantifier:DM:這種|property:NP•的(head:NP(property:Nad:政治|Head:Nac:層面)|Head:DE:的)|Head:Nv4:去勢)|
     Head:VK2:現象)
9) NP(property:VP(evaluation:Dbb:到底(Head:VH11:誠實)|property:Nv4:誠實|Head:Nab:是)|Head:Tc:什麼)
10) S(quantity:Daa:幾乎|time:Dd:已|Head:VC1:到|aspect:Di:了|goal:NP(predication:VP•的(head:VH11:分秒必爭|Head:DE:的)|
      Head:Nad:地步))
11) S(theme:NP(property:Nca:南非|Head:NP(predication:VP•的(head:VP(manner:A:特有|Head:VH11:新鮮)|Head:DE:的)|Head:Nab:龍蝦))|
      theme:Nad:原味|theme:VJ1:浸於|Head:VA11:鹵|goal:Ncda:間)
12) S(quantity:Dab:完全|Head:VJ2:漠視|goal:NP(quantifier:Nep:其|Head:Nv4:存在))
13) S(Head:VL4:使|goal:Nv1:防守|theme:VP(negation:Dc:不|Head:VH14:出現|theme:Nac:漏洞))
14) S(agent:Nba:韓國隊|quantity:Daa:共|Head:VC31:擊出|theme:NP(quantifier:DM:十四支|Head:Nac:安打))
15) NP(property:VP•的(head:VP(manner:VH11:明顯|Head:VH11:無用)|Head:DE:的)|Head:Nad:結構)
16) S(agent:Nhaa:我們|Head:VH11:乾脆|goal:VP(Head:VH11:奢侈|duration:DM:一下))
17) S(Head:VK2:包括|goal:NP(property:Nad:高中|Head:Nab:籃球))
18) S(quantity:Dab:尤|theme:Nv1:應對|goal:PP(Head:P11:以|DUMMY:Nac:變化球)|Head:VI2:為主)
19) conjunction(Head:Cbca:因而)
20) S(goal:Nhaa:他|manner:VP•的(Head:NP(property:VC31:花|particle:Ta:了)|Head:DE:一年)|agent:PP(Head:P49:的|DUMMY:NP(
      property:Nad:時間|Head:Nad:苦練))|agent:Nad:拳擊|Head:VC2:和)
21) S(agent:Nac:終場|manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:十|Head:Caa:比|DUMMY2:Neu:六)))|Head:VC2:擊敗|
      goal:Nba:台電隊)
22) S(Head:VC2:第一道|goal:NP(apposition:Nab:菜|Head:Nba:終於)|complement:VP(Head:VA11:來|particle:Ta:了))
23) S(experiencer:Nhaa:她|Head:VF1:下決心|goal:S(manner:Dh:死命|Head:VA4:減肥))
24) S(agent:NP(property:Nca:蘇聯|Head:Nab:官員)|evaluation:Dbb:則|Head:VE2:表示)
25) S(theme:NP(apposition:Nhaa:你|Head:Nddc:今天)|Head:V_2:有|range:NP(quantifier:Nes:沒有|Head:Nab:空))
26) NP(property:GP•的(head:GP(DUMMY:Nab:長明燈|Head:Ng:裡)|Head:DE:的)|Head:Nba:莊光七)
27) S(contrast:Cbca:否則|epistemics:Dbaa:會|Head:VH11(property:VH11:赤貧|Head:DM:一年))
28) S(agent:Nba:吉田修司|epistemics:Dbaa:是|time:PP(Head:P21:在|DUMMY:VP(time:Ndaba:去年|Head:VC1:升上))|
      Head:VC31:巨人隊|goal:Nab:一軍)
29) NP(DUMMY:NP(quantifier:DM:這位|predication:VP•的(head:VH11:勇敢|Head:DE:的)|Head:Nab:司機)|Head:Tc:老兄)
30) S(theme:Nhab:雙方|topic:PP(Head:P31:對|DUMMY:VP(manner:Dh:如何|Head:VC2:解決|goal:NP(property:Nca:波斯灣|
      Head:Nac:危機)))|time:Dd:仍|Head:VJ3:存有|range:NP(quantifier:Neqa:若干|Head:Nac:歧見))
31) PP(Head:P21:在|DUMMY:GP(DUMMY:NP(property:Nep:此|Head:Ndabf:亂世)|Head:Ng:中))
32) NP(predication:VK1:希望|predication:VP•的(theme:Ncc:世間|Head:DE:的)|Head:Nab:妻子)
33) S(agent:NP(reason:Cbaa:因|DUMMY:Nep:此)|Head:VE2:區|goal:NP(DUMMY1:NP(property:Ndabf:早期|Head:Naeb:受到)|Head:Caa:山|
      DUMMY2:NP(property:NP•的(head:NP(DUMMY1:Nab:溪|Head:Caa:和|DUMMY2:NP(property:Nv1:噴出|Head:Naa:泉水))|Head:DE:的)|
      Head:Nv1:温潤)))
34) S(Head:VL4:致使|goal:NP(property:NP(property:Nac:我國|property:Nad:精密性|Head:Nac:科技)|property:Nv1:醫療|
      Head:Naeb:設備)|theme:VP(theme:NP(reason:Caa:和|Head:Naeb:藥品)|time:Dd:仍|Head:VK2:需|goal:Neqa:大量|goal:Nv1:進口))
35) PP(Head:P03:為了|DUMMY:VA12:一鳴驚人)
36) S(theme:NP(property:Naeb:融資|Head:Naeb:餘額)|manner:H16:增加|quantity:Daa:近|theme:DM:九億元|Head:VG2:為|
      range:DM:２５２億元)
37) S(time:PP(Head:P13:趁|DUMMY:S(theme:NP(property:Nhab:對方|Head:Nab:投手)|negation:Dc:不|Head:VH11:穩))|time:Dd:頻頻|
      Head:VD2:搶|theme:Nac:分)
38) VP(manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:一|Head:Caa:比|DUMMY2:Neu:一)))|Head:VH11:戰平)
39) VP•的(head:VP(benefactor:PP(Head:P03:禁得起|DUMMY:NP(quantifier:DM:每個|Head:Nac:星期))|Head:VC31:假日|theme:NP(
           quantity:Dab:都|property:Ncda:上|Head:Ncb:高爾夫球場))|Head:DE:的)
40) S(Head:VL1:大盤|goal:VP(Head:VC31:燃出|goal:VP(Head:VG2:更|range:NP(predication:VP•的(head:VH11:盛大|Head:DE:之)|
      Head:Naa:火焰))))
41) S(theme:PP(Head:P07:將|DUMMY:Nab:書頁)|Head:VG1:全部|range:Nv4:打開)
42) NP(Head:Ndabb:九月份)
43) NP(predication:VP•的(head:VP(Head:VC31:拿|location:GP(DUMMY:NP(possessor:NP(degree:Dfa:過|Head:Nba:士林電機)|Head:Nab:董事長)|
       Head:Ng:名片))|Head:DE:的)|Head:Nab:人)
44) S(Head:VE2:要|goal:VP(Head:VC2:慎選|goal:Nad:同伴))
45) S(theme:VP(DUMMY:NP(quantifier:DM:兩個|Head:Nab:兒子)|Head:Ng:被)|theme:VC2:派任|Head:VJ1:並|range:NP(property:VP•的(
      head:VP(negation:Dc:不|Head:VH15:合適)|Head:DE:的)|Head:Nac:位置))
46) S(Head:VC2:撞|aspect:Di:了|goal:NP(quantifier:DM:個|quantifier:Neqa:滿|property:Nab:頭|Head:Nv1:金星亂冒))
47) PP(Head:P43:四十位|DUMMY:S(agent:NP(apposition:NP(property:Nad:民主黨籍|Head:Nab:眾議員)|Head:Ndabd:今天)|Head:VE2:說))
48) S(Head:VC2:對|goal:NP(property:Nba:長榮|property:Nad:航空|Head:Ndabd:而言))
49) S(theme:NP(quantifier:DM:前八局|Head:Nbc:曾)|Head:V_2:有|range:NP(quantifier:DM:三次|predication:VC2:攻到|Head:Nab:三壘))
50) NP(DUMMY:Naa:中研院|Head:Cab:等)
```

Figure 19: Appendix B: Coverage with 19.803 training sentences.