Data-Oriented Parsing and the Penn Chinese Treebank

Mary Hearne & Andy Way

School of Computing,
Dublin City University,
Dublin
Email:{mhearne,away}@computing.dcu.ie

Abstract

We present an investigation into parsing the Penn Chinese Treebank using a Data-Oriented Parsing (DOP) approach. DOP comprises an experience-based approach to natural language parsing. Most published research in the DOP framework uses PS-trees as its representation schema. Drawbacks of the DOP approach centre around issues of efficiency. We incorporate recent advances in DOP parsing techniques into a novel DOP parser which generates a compact representation of all subtrees which can be derived from any full parse tree.

We compare our work to previous work on parsing the Penn Chinese Treebank, and provide both a quantitative and qualitative evaluation. While our results in terms of Precision and Recall are slightly below those published in related research, our approach requires no manual encoding of head rules, nor is a development phase *per se* necessary. We also note that certain constructions which were problematic in this previous work can be handled correctly by our DOP parser. Finally, we observe that the 'DOP Hypothesis' is confirmed for parsing the Penn Chinese Treebank.

1 Introduction

We investigate the parsing of the Penn Chinese Treebank (CTB) (Xue, 2004) using a Data-Oriented Parsing (DOP: Bod, 1992; Bod, 1998; Bod et al., 2003) approach. DOP comprises an experience-based approach to natural language parsing. Most published research in the DOP framework uses PS-trees as its representation schema. These trees are broken down into subtrees, which are combined together to parse new sentences. Most criticism of the DOP-based approach centres on questions of efficiency: in general, the number of fragments projected far exceeds the number of grammar rules projected, so standard chart-parsing techniques cannot directly be applied in a DOP parser.

More recently, however, advances have been made which have led to considerable optimisations of DOP models (Sima'an, 1999). Using similar techniques, we have developed a novel DOP parser which optimises for top-down computation of the most probable *parse* rather than bottom-up computation of the most probable *derivation*. Our previous work has used the English component of the Xerox HomeCentre corpus, a collection of 980 sentences which were drawn from printer manuals and annotated using the Lexical-Functional Grammar framework. These trees can be fragmented using the DOP decomposition operations to give in excess of 534 billion fragments. In section 2, we report on a novel, dynamic method that we have developed which generates a compact representation of all fragments which can be derived from a particular tree. This allows us to store and access only the original treebank trees, rather than explicitly creating the entire fragment base. Using this method, we can efficiently retrieve only those fragments directly useful in analysing the given input string. In section 2, we also describe the two-phase analysis and Monte-Carlo disambiguation components in our parser.

The Chinese Treebank comprises 325 articles of Xinhua newswire text in the areas of economics, politics and culture. There are 4185 sentences in total, and approximately 100,000 words (about 1/10 the size of the Penn-II Treebank). Despite the fact that our parser was constructed for English and for a different treebank involving texts from different domains, we did not have to make any adaptations at all in order to parse the CTB. This is due to the fact that it is entirely language independent, requiring only that training data be in the form of context-free phrase-structure trees, thus ensuring the flexibility of the DOP approach. The related research that we describe in section 5 requires the hand coding of a set of head rules for Chinese or the development of a dependency parser, in addition to which a specific 'development phase' is required on top of the normal training stage. In addition, and importantly, our work on parsing the Chinese Treebank shows that the 'DOP Hypothesis', which states that parse accuracy increases as larger fragments are included in the fragment base, is confirmed.

In section 4, we provide the results obtained from running our parser in a number of experiments carried out on the CTB which we describe in section 3. While our results are not directly comparable with the previous research on parsing the CTB, given that different splits into training and test data are used, we perform slightly worse in terms of Precision and Recall compared to the related work. Nonetheless, given that previous work on parsing the CTB employs a rich arsenal of extra resources, purely in quantitative terms, we consider our results to be extremely promising. In section 5, we provide a qualitative comparison of our results with this previous work, and show that certain constructions which were problematic in this work can be handled correctly by our DOP parser. Finally, we conclude and provide some avenues for further research.

2 Data-Oriented Parsing

2.1 Theoretical Background

Data-oriented models of language (e.g. Bod, 1992; Bod 1998) are based on the assumption that humans perceive and produce language by availing of previous language experiences rather than abstract grammar rules. These models exploit large treebanks comprising linguistic representations of previously occurring utterances. Analyses of new input sentences are produced by combining fragments from the treebank; the most probable analysis is determined using the relative frequencies of these fragments.

The tree fragments used in Tree-DOP are called subtrees. Two decomposition operators are used in order to produce subtrees from sentence representations:

1. the *root operator* which takes any node in a tree to be the root of a subtree and deletes all nodes except this new root and all nodes dominated by it;

2. the *frontier operator* which selects a (possibly empty) set of nodes in the newly created subtree, excluding the root, and deletes all subtrees dominated by these nodes.

As an example, the complete set of DOP fragments which can be derived from the representation of *John swims* is shown in Figure 1.

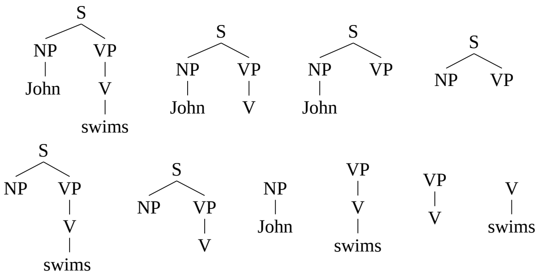


Figure 1: The complete Tree-DOP multiset of fragments for the sentence *John swims*.

Representations for new input are formed by combining other fragments using the composition operator, namely leftmost substitution, which ensures that each derivation in DOP is unique. The composition of trees t_1 and t_2 ($t_1 \circ t_2$) is only possible if the leftmost frontier node of t_1 and the root node of t_2 are of the same category. The resulting tree is a copy of t_1 where t_2 has been substituted at its leftmost nonterminal frontier node, as demonstrated in Figure 2.

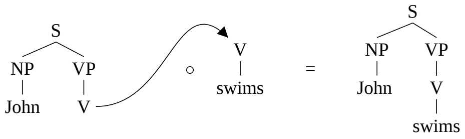


Figure 2: The DOP composition operation

The probability of a derivation is the joint probability of choosing each of the subtrees involved in that derivation. Letting $|e|$ be the number of times subtree *e* occurs in the corpus and r(*e*) be the root node category of *e*, the probability assigned to *e* is

$$P(e) = \frac{|e|}{\sum_{u:r(u)=r(e)} |u|} \tag{1}$$

The probability of a derivation is the product of the probabilities of choosing each of the subtrees involved in that derivation. Thus, the probability of a derivation $t_1 \circ ... \circ t_n$ is given by:

$$P(t_1 \circ ... \circ t_n) = \prod_i P(t_i) \tag{2}$$

A parse tree can potentially be generated by many different derivations, each of which has its own probability of being generated. Therefore, the probability of a parse tree *T* is the sum of the probabilities of its distinct derivations:

$$P(T) = \sum_{D\ derives\ T} P(D) \tag{3}$$

2.2 Implementation

The DOP approach requires the projection of a tree-substitution grammar (i.e. a set of fragments) from a given treebank rather than a context-free grammar as used in rule-based parsing. However, in general, the number of fragments projected far exceeds the number of grammar rules projected. This means that it is not feasible, in terms of time and memory, to directly apply standard chart-parsing techniques in the development of a DOP system.

2.2.1 Fragmentation

The 980 trees contained in the English section of the HomeCentre corpus can be generalised to give in excess of 534 billion fragments. Even generating only those fragments of depth 6 or less results in over 4.5 million fragments. Clearly, generating, storing and searching this number of fragments, as well as gathering frequencies of occurrence for each subtree, is a non-trivial task.

As outlined in Section 2.1, tree fragments are extracted by firstly applying the root operation to each original treebank tree, yielding intermediate fragments, and then applying the frontier operation to each of these intermediate fragments in turn to generate the complete set of fragments. As an alternative, we have developed a dynamic method to generate a compact representation of all fragments that can be derived from a particular tree.

Compact representations are built by firstly applying the root operation, creating an intermediate tree for each node in the original tree. Then, rather than explicitly applying the frontier operation, we associate each fragment that can be generated by applying the frontier operation to intermediate trees with a unique number. In the example in figure 3, the tree on the left representing the noun phrase *the man* yields a total of six fragments. In this instance, we associate these fragments with the numbers 1 – 6. Application of the root operation results in the creation of the three intermediate trees to the right with root nodes NP, D and N. Nodes in intermediate trees are annotated with fragment numbers such that the presence of a particular number at any given node in the tree indicates that this node is also present in the relevant fragment. The annotation of the intermediate trees with root nodes D and N in figure 3 is trivial because application of the frontier operation will result in the extraction of only one fragment from each. The annotation of the intermediate tree with root NP is more complex as four fragments can be extracted from it via frontier. If a fragment number is absent at a non-frontier node but present at its parent node then this indicates that, in that particular fragment, the node is a substitution site. All possible fragments of a given tree can be generated by reading off one fragment at a time via the presence or absence of its unique fragment number at each node. The annotation of the tree with root node NP in figure 3 indicates that fragments 1 – 4 have root node NP, that node D is a substitution site in fragment 2 and N is a substitution site in fragment 3, and that both D and N are substitution sites in fragment 1. Fragment 4 corresponds exactly to the original tree. Frequencies are calculated by recursively comparing all annotated trees and identifying duplicates.

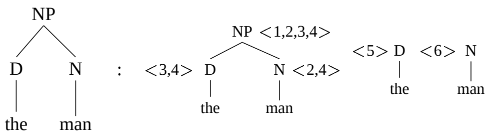


Figure 3: Compact fragment representation for the tree representing the NP *the man*.

This method allows us to store and access only the original treebank trees, thus alleviating the need to explicitly create the fragment base – a task which, given a corpus of reasonable size and complexity, quickly becomes unfeasible. Instead, we can efficiently retrieve only those fragments directly useful in analysing the given input string.

2.2.2 Analysis

A chart built during the analysis phase is a compact representation of all possible derivations leading to valid parses of the input string, which can be constructed either bottom-up or top-down. In order to build an STSG chart using conventional chart-parsing techniques, each fragment must be expressed as a rewrite rule of the form *root* $\longrightarrow$ *frontier*$_1$ ... *frontier*$_n$ and a direct reference to the original tree structure must be retained. However, these approaches are not designed to handle the sheer numbers of fragments involved in parsing within the DOP framework. We have developed a two-phase analysis component based on an optimisation proposed by (Sima'an, 1999). However, we have optimised for top-down computation of the most probable parse rather than bottom-up computation of the most probable derivation.

The set of parses that can be generated for any given sentence using a tree-substitution grammar is a subset of those that can be generated by means of the context-free grammar underlying that tree-substitution grammar. Thus, the first phase of analysis involves using the context-free grammar underlying the treebank to compute an approximation of the parse space for the input using the CKY algorithm as illustrated in figure 4. Given that the grammar underlying the English section of the HomeCentre corpus comprises just 2606 rules, this clearly constitutes a dramatic reduction of the initial search space. During the second phase, il-

2	NP ⟶ $D_{[0][1]}$ $N_{[1][1]}$, <1,2,3,4>	
1	D ⟶ the, <3,4,5>	N ⟶ man, <2,4,6>
	0 ”the”	1 ”man”

Figure 4: CFG parse space for *the man*

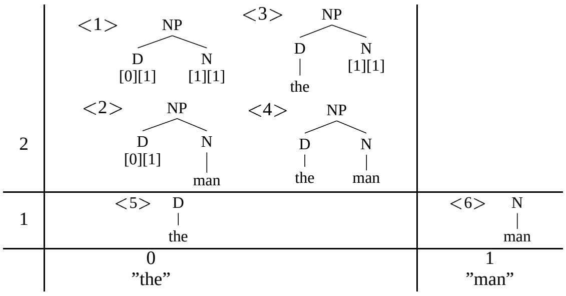


Figure 5: STSG parse space for *the man*

lustrated in figure 5, the tree-substitution grammar is applied to this reduced parse space to generate the exact DOP parse space for the input string.

In order to reduce from the CFG parse space to the STSG parse space, a correspondence must be drawn between the context-free grammar rules used during the first phase and the tree fragments we wish to insert into the chart during the second phase. The fragmentation process described in Section 2.2.1 provides this correspondence because it allows the identification of all fragments in which each context-free grammar rule occurs. When extracting CFG rules from the treebank tree in figure 3, we also find all occurrences of each rule in the set of annotated trees and extract the annotations on the node corresponding to the left hand side of the relevant rule. Thus, all rules in figure 4 are annotated with explicit references to the fragments in which they occur in the tree-substitution grammar. Rather than returning to the tree-substitution grammar, this information allows us to rebuild the set of fragments appropriate to the current parse space as shown in figure 5, thus resulting in a highly optimised second analysis phase.

2.2.3 Disambiguation

Disambiguation is the final stage in the parsing process and involves selecting the most probable parse or derivation from the parse chart. Within the DOP framework this constitutes an NP-complete problem (Sima'an, 1999) as many different derivations can result in the same parse and, therefore, the most probable derivation (MPD) does not necessarily equal the most probable parse (MPP).

Monte-Carlo sampling involves searching over a reduced random sample of the search space which can be generated in polynomial time and was first proposed as a method for maximisation of the MPP in the DOP framework in (Bod, 1992). Our implementation incorporates the refinements detailed in (Chappelier & Rajman, 2003).

3 Experiments

We have performed experiments on a subset of the Penn Chinese Treebank (Version 2.0) (Xue, 2004). We calculated the dimensions of each tree in terms of its depth, width and number of nodes and selected only those trees which were of average size or smaller, resulting in a dataset containing 1473 treebank trees. We then divided this dataset into three random training/test splits. The sole constraint imposed on each split was that all words in the testset also be present in the training data. Each testset contained 150 sentences and each training set contained 1323 treebank trees.

In addition to performing standard tree normalisations – the removal of empty nodes, trees dominating no non-empty nodes and A over A unaries – we also removed X over A unaries such that all unary-branching trees are of the form PRE-TERMINAL ⟶ terminal. We did not remove functional information from the syntactic tags. During disambiguation, the maximum number of samples taken was 5,000.

In DOP, the fragment space is generally pruned by excluding fragments greater than a certain depth in order to render the search for the most probable parse tractable. For each split, we performed three sets of experiments, limiting the fragment space to fragments of depth 1, depth 2 or less and depth 3 or less. Furthermore, these experiments were performed on both tagged and untagged input.

3.1 Parsing tagged input with DOP

When parsing tagged input, two options present themselves. The first involves taking as input only tag sequences and parsing them as though they were terminals, while the second involves taking as input <tag,word> pairs. Unlike PCFG parsing, these two approaches are not equivalent for DOP because DOP grammars contain lexicalised fragments. Under the first approach, all lexicalised fragments are immediately excluded from the parse space. The second approach, on the other hand, only excludes those lexicalised fragments whose pre-terminals do not correspond to the input tags and is, therefore, inherently more powerful. This approach can be viewed as an input-driven pruning mechanism and is the methodology we have chosen to adopt.

In certain instances, adhering to the specified tag sequence will result in no parse being produced. This generally indicates a word of unknown category, i.e. a word which was seen in the training data, but never with the tag with which we now see it in the input string. Here, we have chosen to treat such words as "un-tagged" words and simply include in the parse space all relevant lexicalised fragments, regardless of the pre-terminals they specify for these words. Where we have successfully constructed a parse space covering all input words but still cannot produce a full parse, we revert to an "un-tagged" parse.

Depth	Recall	Precision	F-score
1	62.68	63.22	62.94
2	69.96	68.09	69.01
3	72.93	69.73	71.29

Table 1: Results achieved on untagged input.

Depth	Recall	Precision	F-score
1	70.69	69.55	70.11
2	77.35	74.28	75.78
3	77.92	74.46	76.15

Table 2: Results achieved on tagged input.

4 Results

4.1 Quality

Table 1 shows standard recall, precision and f-score results on untagged input strings at depths 1, 2 and 3 averaged over all splits. Increasing the size of the fragment base to include fragments of depth 2 results in a 7.28% increase in recall and a 4.87% increase in precision. Increasing from depth 2 to depth 3 results in further increases in accuracy of 2.97% for recall and 1.64% for precision. The average increase in f-score from depth 1 to depth 3 is 8.35%.

Table 2 shows recall, precision and f-score results on tagged input strings at depths 1, 2 and 3, again averaged over all splits. Increasing the size of the fragment base to include fragments of depth 2 results in a 6.66% increase in recall and a 4.73% increase in precision. Increasing from depth 2 to depth 3 results in small increases in recall and precision of 0.57% and 0.18% respectively. The average increase in f-score from depth 1 to depth 3 is 6.04%.

The *DOP Hypothesis* states that parse accuracy increases as larger fragments are included in the fragment base. This hypothesis has been shown for the first time to hold for the parsing of English on several different treebanks (Bod, 1998; Bod & Kaplan, 2003; Bod, 2003). It has recently been shown to hold for Data-Oriented Translation from English to French when the DOT system is trained on the HomeCentre Corpus (Hearne & Way, 2003). The results presented here confirm that this hypothesis also holds for the parsing of Chinese text when the parser is trained on the Chinese Penn Treebank. However, the increase in accuracy from depth 2 to depth 3 on tagged input is minimal.

Depth	secs/sentence	frags/sentence
1	94.39	373.38
2	117.65	1407.77
3	121.99	1493.89

Table 3: Efficiency on untagged input.

Depth	secs/sentence	frags/sentence
1	57.60	263.29
2	76.93	976.82
3	88.16	1182.07

Table 4: Efficiency on tagged input.

4.2 Efficiency

The time taken to parse raw input strings varies from 94.39 secs/sentence at depth 1 to 121.99 secs/sentence at depth 3, as shown in Table 3. Obviously, parsing is faster over tagged strings due to the corresponding reduction in ambiguity. Table 4 shows that parse times on tagged input vary between 68.29 secs/sentence at depth 1 and 88.16 secs/sentence at depth 3. These tables also clearly illustrate that average parse times generally correspond to the average number of fragments present in the parse space for each sentence at each depth. It is reasonable to expect that, as the number of training fragments available increases and the number of fragments relevant to the parse space increases, the time taken to produce a parse also increases. However, parsing can be separated into two distinct phases: the construction of the parse space and the selection of the most probable parse. While this expectation holds true for the first phase, it is not necessarily the case for disambiguation. Despite increases in the average numbers of training fragments and relevant fragments, parse times decrease for splits s1 and s2 by 2.68 secs/sentence and 4.23 secs/sentence respectively from depth 2 to depth 3. As sentence length and the number of samples taken remains constant at each depth, variation in disambiguation time is due to variation in the lengths of the derivations sampled. Longer derivations arise where many smaller fragments are sampled, and these derivations require more time. As fragment depth increases, larger fragments are available for selection, resulting in shorter derivations and, therefore, decreased disambiguation time.

No comparison of parse times is possible given that the previous work on parsing the CTB did not provide any such details. While our parse times may be deemed rather slow, faster times for data-oriented parsing have been achieved by extracting a probabilis-

	Precision	Recall	F-Score
Bikel & Chiang 2000	77.2	76.2	76.7
Levy & Manning 2003	78.4	79.2	78.8
Chiang & Bikel 2002	81.8	78.8	79.9

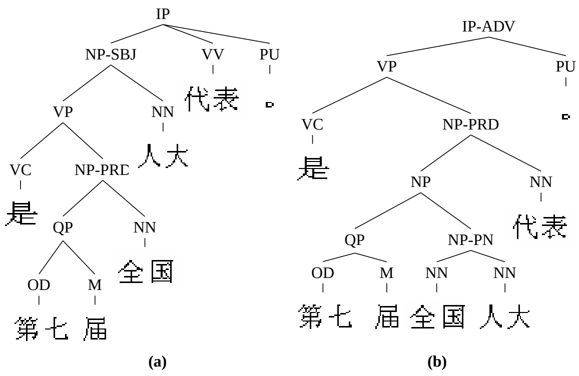
Table 5: Previous Results on Parsing the Chinese Treebank for sentences less than or equal to 40 words.

tic context-free grammar (PCFG) which generates the same strings and trees with the same probabilities as the corresponding DOP grammar (Goodman, 2003). While this is worthy of investigation in further research, our primary aim is not parsing *per se*, but rather machine translation (MT). Our intention is to build large-scale DOP and LFG-DOP (Bod & Kaplan, 1998) systems (cf. Poutsma, 2000; Hearne and Way, 2003; Way, 2003). Such models require aligned PS-trees (and, for LFG-DOP models, LFG f(unctional)-structures corresponding to these trees, hence our use of the HomeCentre corpus), and to date, no efficient PCFG reduction has been developed which can be applied to a bilingual treebank and which will generate the same source/target strings and trees with the same probabilities as the corresponding bilingual DOP-based grammar. Accordingly, if we were to adapt our parser to incorporate Goodman's ideas, there is no guarantee that such savings would carry over to MT. We have, therefore, decided to maintain the flexibility of a DOP-parser which is immediately utilisable in the area of machine translation.

5 Contrast with Related Research

Previous work on parsing the CTB includes (Bikel & Chiang, 2000), (Chiang & Bikel, 2002) and (Levy & Manning, 2003). Bikel & Chiang (2000) use two models for their experiments, one based on the BBN model of (Miller et al., 1998), and the other on Tree-Insertion Grammar (TIG) (Schabes & Waters, 1995), adapted from (Chiang, 2000). Chiang & Bikel (2002) uses the same TIG-parser, but use Inside-Outside reestimation to improve the set of head rules for Chinese given in (Bikel & Chiang, 2000). Levy & Manning (2003) use the factored parsing model of (Klein & Manning, 2002), which involves combining a parse derived from a non-lexicalised, maximum likelihood estimated PCFG with a parse obtained independently from a dependency model.

Levy & Manning (2003) discuss why they chose not to use the same training and test data as (Bikel & Chiang, 2000). The latter used articles 1–270 for training, 301–325 for system development, and 271–300 for testing. Levy & Manning (2003) point out that "this development set was uncharacteristic of the corpus as a whole and not ideal for development". Accordingly, they use articles 1–25 for development and 26–270 for training under development. Given these discrepancies, the two approaches are not directly comparable. Despite the differences in training and development data, they nevertheless performed experiments on the same testset. The respective results from these three approaches on this test data are given in Table 5.



Gloss: *is deputy to the 7th NPC*[1]

Figure 6: NN mis-tagged as VV at depth 1 (top) is correctly tagged at depths 2 and 3.

As stated in section 3, we use different training and test sets again compared to these previously published papers. We need to ensure that certain trees were excluded from these datasets so that the number of tree fragments was not overly onerous. Given this, our results are not directly comparable with those given in Table 5. Our f-scores are 8.61% lower than those of (Chiang & Bikel, 2002) on un-tagged input strings and 3.75% lower on tagged input. More experiments are required to determine whether further increasing fragment depth and the amount of training data used will result in improved performance.

Levy & Manning (2003) provide an in-depth analysis of various error types according to a number of criteria: multilevel VP adjunction, NP-NP modification, Coordination, and tagging errors. In the next three sections, we provide a comparison with Levy & Manning (*op cit.*) on the latter three types of error. This comparison is based on manual analysis of the parses produced for 100 test sentences, all of which were contained in the same training/test split and were therefore parsed over the same training data at each depth.

5.1 Tagging

Levy & Manning (2003) observe that the main error in tagging was the tendency to mistag verbs (VV) as common nouns (NN) and vice versa. They note that while all languages provide a means whereby verbs can be converted into nouns, this is particularly a problem in Chinese, given its sparse morphology. While this is also true of English, morphological variants of ambiguous N-V words can be inserted to resolve the ambiguity. The only way in which such ambiguity can be resolved in Chinese is to see whether adverbial or prenominal modifiers can co-occur with the said word

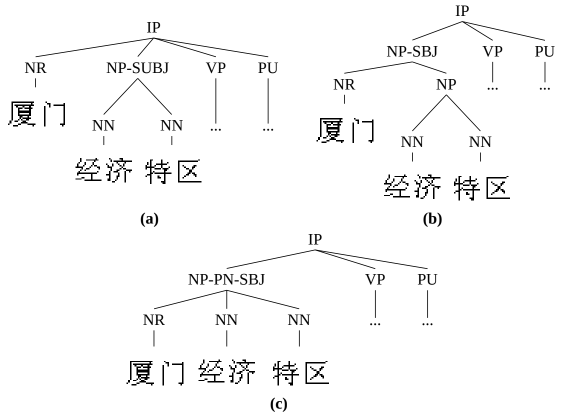
In order to try to evaluate the impact of N-V ambiguity in Chinese, Levy & Manning (*op cit.*) trained their parser with the VV and NN tags merged. Unsurprisingly, the F-scores decrease: by 5.4% for their vanilla PCFG parser, and by 1.7% for the refined model.

When parsing raw input strings, our tagging accuracy increases from 92.48% at depth 1 to 93.92% at depth 2, with no further improvement at depth 3. We also observed that the main source of error concerned ambiguity as to whether certain words should be tagged as nouns or verbs – these errors accounted for 38.85% of all incorrect tag assignments. The addition of fragments of depth 2 to the parse space reduced this type of error to a certain extent – as illustrated in Figure 6 – but no further improvements were seen as fragments of depth 3 were introduced.

5.2 NP-NP modification

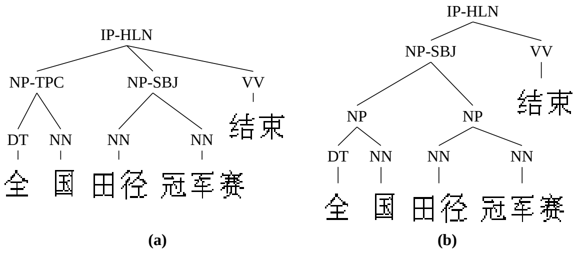
Levy & Manning (*op cit.*) note that this type of error was the most common in their experiments. Compound noun interpretation is notoriously difficult in English as well, of course, but Levy & Manning observe that such structures typically receive a flat interpretation in the Penn-II Treebank. While such ambiguity is difficult to resolve in Chinese, the fact that the different semantic interpretations will have different dependency parses enables certain cases to be interpreted correctly, but only “when word frequencies are large enough to be reliable”. Nevertheless, even where the dependency parse was unable to help, they noted that “the internal distributions (i) of NP modifiers and (ii) left-modified NPs both differ from the internal distribution of NPs in general”. Accordingly, they mark each type as (i) or (ii) in the PCFG parser which reduces the amount of bias against NP-NP modification in nominal compounds.

Again, we found that the addition of larger fragments to the parse space led to greater accuracy in the interpretation of compound nouns. Figure 7 illustrates how, as the available context increases, the required shallow NR-NN-NN modification is correctly identified, while Figure 8 shows the alternative situation, where a deeper parse is required. However, even at depth 3 NP-NP modification errors are still common.



Gloss: *Xiamen Special Economic Zone...*[1]

Figure 7: Shallow NR-NN-NN modification is correctly identified stepwise as depth increases from depth 1 (top) to depth 3 (bottom).



Gloss: *The National Track and Field Championship has finished*[1]

Figure 8: NP-NP modification, incorrectly analysed at depth 1 (top) is correctly analysed at depths 2 & 3.

5.3 Coordination

With respect to coordination, Levy & Manning (*op cit.*) found two main error types: misattachment of the right conjunct where this is either verbal or nominal. There are two main problems for VP coordination: firstly, due to *pro*-drop, any VP coordination is ambiguous with a higher IP coordination (assuming there to be a rule $IP \longrightarrow VP$ somewhere in the grammar); and secondly, VPs in the CTB are multi-level, which makes it difficult to establish the scoping of adjuncts. Levy & Manning (*op cit.*) find that the first of these problems can be lessened somewhat by marking adverbs which possess an IP grandparent, while the second problem is alleviated to a certain extent by marking VPs as adjunction or complementation structures. They also note that only like VPs are coordinated in their training phase. As for NP coordination, the major scoping problem was in false high scopings, which are reduced by the marking of NP conjuncts. They found no cases of false low attachments at all for NPs.

We encountered similar difficulties in analysing NP coordination, and achieved little improvement despite the additional contextual information available when larger fragments were added to the parse space. Con-

[1] Translations provided by a native speaker of Chinese with fluent English.

trary to the observations of Levy & Manning (*op cit.*), we found no errors in VP coordination. However, VP coordination was not particularly common in the set of manually analysed parses and further investigation is needed in order gain a clearer picture as to how it is analysed under the DOP approach.

6 Conclusions and Future Work

This work has provided an account of how our Data-Oriented Parser fared in parsing the Chinese Treebank. Despite the fact that our parser was initially constructed for English on a different treebank involving texts from different domains, we did not have to make any adaptations at all in order to parse the CTB. Unlike related research, no further mechanisms such as a manual encoding of head rules for Chinese or a dependency parser, were required.

While our results are not directly comparable with this related research, our figures in terms of Precision and Recall are slightly lower. Nonetheless, given the fact that the related work requires a number of other resources, we consider our results to be extremely promising. In addition, in a qualitative evaluation, we observed that our approach was better able to handle certain constructions which posed problems in previous work.

This is the first attempt to apply data-oriented methods to the CTB, and importantly our work confirms the 'DOP Hypothesis', which states that parse accuracy increases as larger fragments are included in the fragment base.

As for further work, while our main interests are in the area of DOP-based models of translation, there remain insights from Goodman's (2003) research which show that parse times can be decreased considerably for DOP. In addition, we would like to apply our parser to sections of the Penn-II Treebank to compare our results on Chinese for English.

Acknowledgements

Our thanks are due to Khalil Sima'an for providing useful feedback on some of the issues discussed in this paper and to three anonymous reviewers for their helpful comments. This work was supported by the *China / Ireland Science and Technology Collaboration Research Fund*.

References

Daniel Bikel & David Chiang. 2000. Two Statistical Parsing Models applied to the Chinese Treebank. In *Proc. 2nd Chinese Language Processing Workshop*, pp.1–6.

Rens Bod. 1992. A Computational Model of Language Performance. In *Proceedings COLING-92*, Nantes, France, pp. 855 – 859.

Rens Bod. 1998. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, Stanford, CA.

Rens Bod. 2003. An Efficient Implementation of a New DOP Model. In *Proceedings of EACL'03*, Budapest, Hungary, pp. 19 – 26.

Rens Bod and Ron Kaplan. 1998. A Probabilistic Corpus-Driven Model for Lexical Functional Analysis. In *COLING-ACL '98*, Montreal, Canada, pp.145–151.

Rens Bod and Ron Kaplan. 2003. A DOP Model for Lexical-Functional Grammar. In Bod *et al.*, eds. (2003), pp. 211 – 232.

Rens Bod, Remko Scha and Khalil Sima'an. eds. 2003. *Data-Oriented Parsing*. CSLI, Stanford CA.

Jean-Cédric Chappelier and Martin Rajman. 2003. Parsing DOP with Monte-Carlo Techniques. In Bod *et al.*, eds. (2003), pp. 83 – 106.

David Chiang. 2000. Statistical Parsing with an automatically-extracted Tree Adjoining Grammar. In *Proc. 38th ACL*, Hong Kong, China, pp. 456 – 463.

David Chiang and Daniel Bikel. 2002. Recovering Latent Information in Treebanks. In *Proc. COLING-2002*, pp.183–189.

Joshua Goodman. 2003. Efficient parsing of DOP with PCFG-reductions. In Bod *et al.*, eds. (2003), pp. 125 – 146.

Mary Hearne and Andy Way. 2003. Seeing the Wood for the Trees: Data-Oriented Translation. In *Proceedings of MT Summit IX*, New Orleans, USA, pp.165–172.

Dan Klein and Chris Manning. 2002. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Proc. of 15th Conference on the Advances in Neural Information Processing Systems (NIPS-2002)*, Vancouver, Canada.

Roger Levy and Chris Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proc. ACL-03*, Sapporo, Japan, pp. 439 – 446.

Scott Miller, Heidi Fox, Lance Ramshaw and Ralph Weischedel. 1998. SIFT – Statistically-derived Information from Text. In *Proc. Seventh Message Understanding Conference (MUC-7)*, Washington, DC.

Arjen Poutsma. 2000. Data-Oriented Translation. In *18th COLING*, Saarbrücken, Germany, pp.635–641.

Yves Schabes and Richard Waters. 1995. Tree Insertion Grammar–A Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced. *Computational Linguistics* **21**:479–513.

Khalil Sima'an. 1999. *Learning Efficient Disambiguation*, PhD Thesis, University of Utrecht, The Netherlands.

Andy Way. 2003. Machine Translation using LFG-DOP. In Bod *et al.*, eds. (2003), pp. 359 – 384.

Nianwen Xue, Fu-Dong Chiou and Martha Palmer. 2004. 'The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus', In *Natural Language Engineering*, 10(4):1-30.