NTUMT Strategy

for Prepositional-Phrase Attachment

Ka-Wai Chui, Yia-ping Lin,

徐嘉慧 林雅萍

Shuanfan Huang & I-Peng Lin

黃宣範 林一鵬

National Taiwan University

國立台灣大學

Proceedings of ROCLING II (1989)

中華民國第二屆計算語言學研討會論文集 163-186 頁

Abstract

This paper aims to propose a new parsing strategy to tackle the notorious prepositional-phrase attachment problem (PP attachment problem) in our NTUMT system.

First of all, correct PP attachment is determined in our PP-Attachment Table (PAT), which requires both syntactic and semantic analyses of verbs, nouns and prepositions in lexicon. PAT is indeed the component where all the idiosyncratic attachment conditions are specified for prepositions.

As to our parsing strategy, it can be considered as an interaction between two drivers -- Intelligent ATN (IATN) and Phrase Structure ATN (PATN). IATN scans the input sentence leftwards and activates PATN to construct the first bar-level structures. PATN is then responsible for building up structures whenever IATN gives the command. Correct PP attachment is governed by the seven states in the IATN Grammar, giving priority to verbs. Since our system just outputs one parsing tree, for a PP which is ambiguous in attaching both to the preceding verb and the preceding noun, the PP will be assigned to be verb modifier.

NTUMT Strategy for Prepositional Phrase Attachment

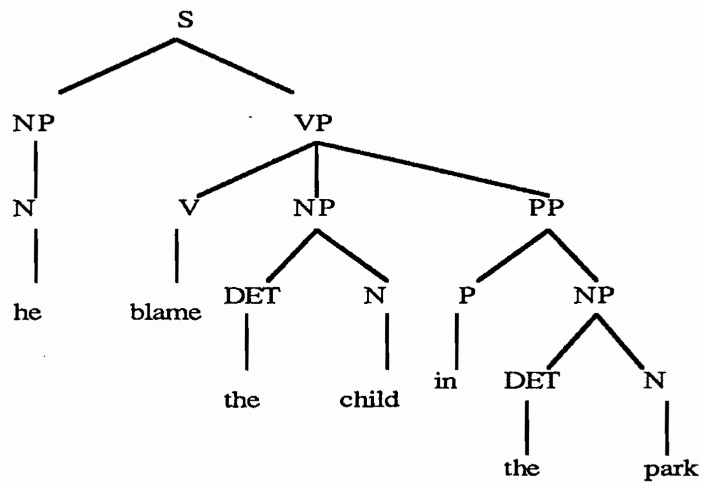
1. The Problem

English prepositional phrases (PP), the postverbal ones in particular, have always been a major problem in parsing. The problem has to do with correctly attaching them to other sentence constituents. This problem of prepositional-phrase attachment (PP attachment) can be exemplified in the following sentence:

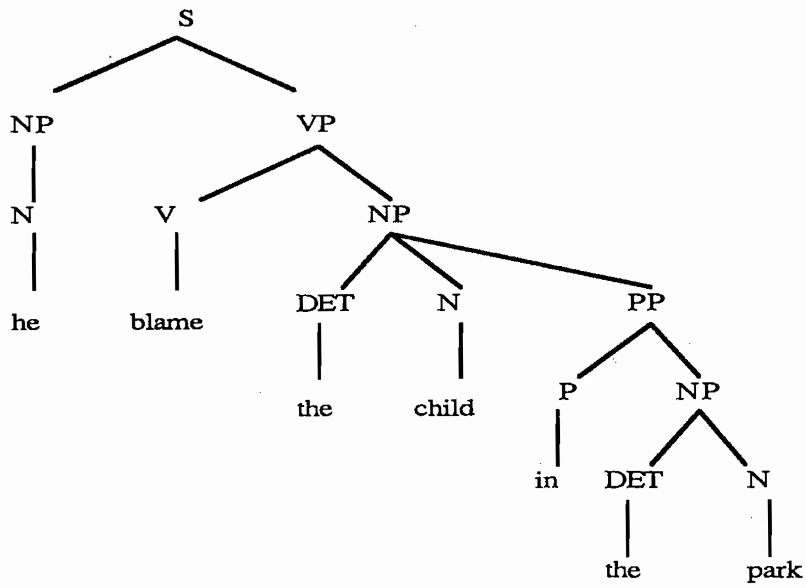
(1) He blamed the child in the park

Sentence (1) is ambiguous in that the PP *in the park* can either be a verb modifier, meaning that the whole event happens in the park; or a noun modifier, showing that the child he blamed is in the park, not elsewhere. Two different tree diagrams, (1a) and (1b), then result:

(1a)



(1b)

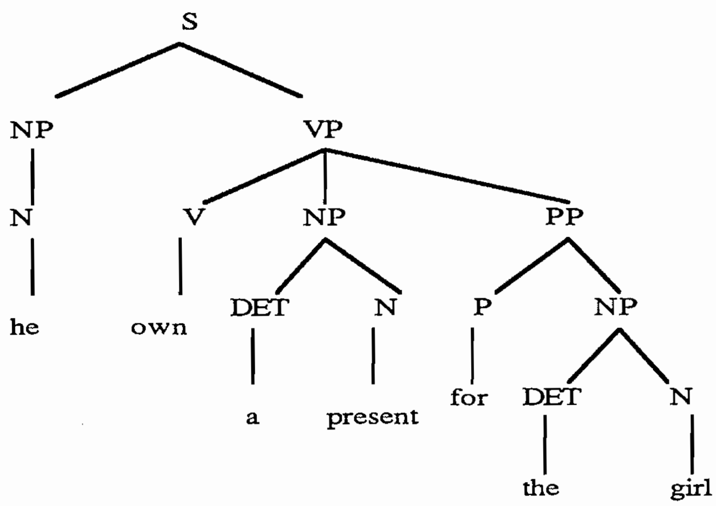


The syntactic ambiguity in sentence (1) does not constitute any semantic anomaly. Disambiguation of this type relies on contextual information. However, some type of attachment is semantically unacceptable. Consider the following sentence:

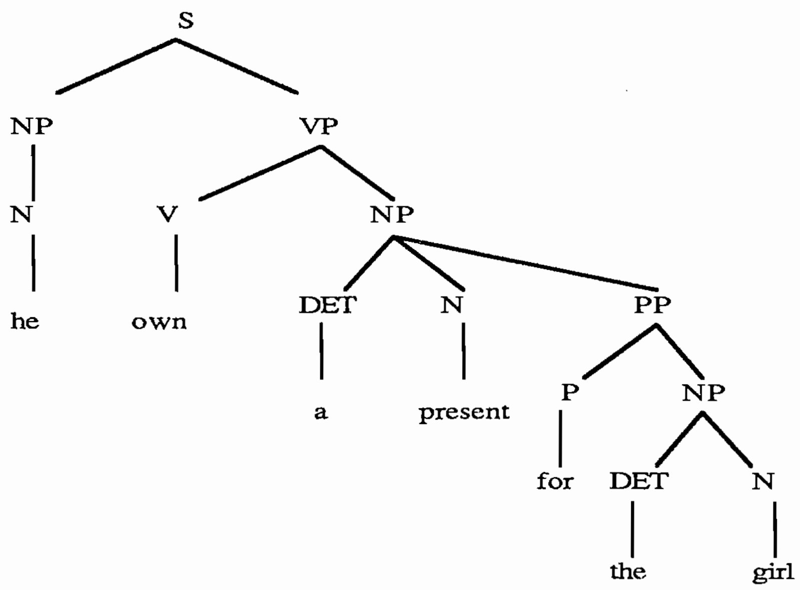
(2) He owned a present <u>for the girl</u>

Sentence (2) shares with sentence (1) in yielding two parsing trees according to our context-free phrase structure rules:

(2a)



(2b)



For the PP *for the girl* to be verb modifier as in (2a) is semantically anomalous. The correct attachment is attaching the PP to the noun *present*. Disambiguation of this type does require both syntactic and semantic analyses of verbs, nouns and prepositions.

In short, PP attachment is not a trivial problem. Of the 929 sentences found in volume 5 and 6 of the English textbooks used by the 3rd-year junior high school students, 36.4% of which include at least one PP. Detailed statistics are displayed in the following table:

	volume 5	volume 6	Total
sentences with at least one PP	152	186	338
sentences without PP	341	250	591
Total	493	436	929

Any English parser should be capable of handling the ambiguous PP attachment as in (1) and (2), but rejecting those semantically unacceptable ones like (2a).

2. Literary Review

In the past, different approaches have been proposed to tackle this PP attachment problem, including Frazier and Fodor's Right Association and Minimal Attachment (1979), Fodor's Lexical Preference (1981), Hirst's Principle of Parsimony (1984), Wilks' Preference Semantics, as well as Schubert's solution by taking syntax, semantics and pragmatics into account. Since these previous treatments are far from satisfactory, Xiuming Huang (1987) presented his most recent resolution for PP attachment in his XTRA system.

Although Huang relies on the integration of syntactic analysis with semantic and contextual interpretation by means of the case preferences of verbs, nouns and prepositions, the *pp-attachment* strategy adopted in his XTRA system suffers from a number of defects. The most serious of which is the inadequacy of the 'seven clauses', which have to be applied sequentially until one succeeds. His clause 1 states that '% check the noun phrase immediately preceding the pp for any case preferences. If its preferences are satisfied then attach the pp to the (Object) np, producing *Rebuilt-Object* (p.115).' Huang obviously takes noun case preference as priority, starting with the noun phrase immediately preceding the PP and working leftwards. This ensures correct PP attachment in those sentences like the following:

(3) **He lost the ticket to Paris**

According to Huang, the semantic formula for one sense of the noun *ticket* has a direction case (p.114):

sem(ticket1, ... , preps([prep(to), prep-obj(*pla), case(direction)]))

Since *ticket* includes a case preference of direction that matches the preplate for *to*, the PP *to Paris* performs as noun modifier exclusively. Nevertheless, there are counterexamples showing that such priority for nouns may trigger off wrong attachment. Consider the following sentence:

(4) He sent the ticket to Paris

Clause 1 is invalid in (4) because the PP *to Paris* being assigned to the preceding noun, just like (3), violates the semantic rule. Besides, it is the verb *sent* which subcategorizes obligatorily this direction case. PP attachment in (4) depends on the verb, rather than on the noun. Though the PP is by no means subcatgorized by the verb *lost* in (3), it is believed that correct attachment can still rely on the propertites of the verb and the preposition itself in that the direction indicated by *to Paris* has to co-occur with those locomotion verbs. Since *lost* indicates a state instead, it fails to take a direction case. The PP will thus be assigned to the preceding noun automatically. In short, not only does the discussion above indicates that the sequential application of Huang's seven clauses may constitute wrong attachment, but using noun case preference as priority further neglects the importance of verb subcategorization, as well as the semantic properties of verbs.

In the following, the parsing strategy adopted in our NTUMT system will be introduced. The strategy ensures correct PP attachment for the 338 sentences in the two English textbooks, which even include two and three postverbal prepositional phrases like the followings:

(5) He drove his car to the market in town

(6) He saw the money on the desk in the room next to mine

Since our system just outputs one parsing tree, for a PP which is ambiguous in attaching both to the preceding verb and the preceding noun, just like sentence (1), the PP will be assigned to become verb modifier as verbs always maintain priority in our system.

3. NTUMT Strategy for PP Attachment

In our NTUMT system, the syntactic and semantic analyses of verbs, nouns and prepositions in lexicon are an integral part of the system. The PP Attachment Table (PAT) is devised to base on these types of information in determining correct PP attachment. Besides, the essence of our parsing strategy gives priority to verb, and emphasizing the surface order of input sentences in that the passive counterpart of (4) allows the PP to be noun modifier:

(7) The ticket to Paris was sent by him

As a matter of fact, (7) does not cause any attachment problem.

In the following, the syntactic and semantic analyses of verbs, nouns and prepositions in lexicon, the PP Attachment Table, together with the parsing strategy, will be introduced respectively.

3.1 Lexicon

3.1.1 verbs

The syntactic and semantic analyses of verbs are crucial to PP attachment. Not only do they provide the subcategorization information which shows whether the PP in question is an argument or a modifier, such as *to Paris* of *sent* in (4); but they also specify the co-occurrence restrictions with PP by virtue of their own semantic feature, such as *lost* in relation to the PP in (3).

Take *push* for instance, it is such a ditransitive verb that it requires two arguments-- a noun phrase and a prepositional phrase. According to our classification, the syntax of *push* belongs to T9:

(push push (V (SUBCAT T9))

The semantic interpretation of verb comprises case and feature assignment. Hence, *push* assigns *patient* to the following concrete noun, the case of *goal* to the prepositional phrase. Together with the semantic property of the verb, being assigned the feature+*action*, all these information of *push* are represented in lexicon as follows:

(push push (V (SUBCAT T9) (F +action)) (T9 ((SUBJ agent +animate) (OBJ1 patient +concrete) (PP goal NIL))))

3.1.2 nouns

As argued in section 2, using case preference of noun to solve the PP attachment problem does not guarantee correct attachment in every situation, mainly because PP is usually nominal modifier, rather than argument. Take example (3) for instance, the noun *ticket* co-occuring with a direction case *to Paris* may also co-occur with a benefactive role as in (8):

(8) He lost the ticket for Mary

Therefore, it is unreasonable to specify just the direction case while the benefactive case is out of consideration. However, specifying exhaustively the possible optional PP a particular noun may take is also uneconomical, especially the information from verb and preposition are so explicit and bountiful. In lexicon, optional PP will not be specified for nouns.

Of course, we are not denying the co-occurrence restrictions of certain PP to particular nouns. On the contrary, there are certain nouns which do subcategorize PP, e.g. the time noun (represented as *Nt* in our classification) in (9):

(9) It is time for winter vacation

The subcategorized information will be specified for those nouns. Besides, every noun has its own inherent nominal features as exemplified below:

(time time (NOUN (SUBCAT Nt) (NUM SG)) (Nt (F +time) (P event)))

3.1.3 prepositions

Prepositions are the main character in PP attachment. Their syntactic and semantic analyses are the input information to PAT.

Firstly, not all of the prepositions subcategorize noun phrase exclusively. *by*, for example, subcategorizes either a noun phrase (PREP1 in our classification) or a gerundive phrase (PREP6 in our classification).

(10) He went to school by bus

(11) He earned money by writing stories

These syntactic information can help solve the attachment problem to a certain extent as *by* being *PREP6* always modifies verb. They are represented in our lexicon as:

(by by (P (SUBCAT PREP1 PREP6)))

Secondly, for each subcategorization, semantic analysis will be provided in form of case and feature. In fact, it is possible for a preposition to carry different semantic cases. For instance, *by* being *PREP1* may perform the roles of location, instrument, time, and agent, which are exemplified below:

(12) location: He stood by the window

(13) instrument: He went to school by bus

(14) time: He will finish his homework by tomorrow

(15) agent: He was hurt by the dog

Deciding what semantic role(s) a preposition takes depends on many factors. The first is the noun that follows. Thus, the PP in (14) suggests *time* mainly because *tomorrow* is a time noun. The second factor is the semantics of verb in that for the PP *by the window* in (12) to be a location, the verb has to indicate a state, just like *stood*. The last factor is sentence type The PP in (15) is agentive mainly due to the passive construction in which it appears. The semantic cases, together with the various types of conditions, are represented in lexicon as follows:

(by by (P (SUBCAT PREP1 PREP6)) (PREP1 (location +state) (instrument +vehicle) (time +time) (agent +passive)) (PREP6 (event)))

Of course, there are cases that rely on none of the factors stated above for identification. The case of goal in (16) is always subcategorized by verb, and no further condition is specified.

(16) goal: He put the money in his pocket

3.2 PP Attachment Table (PAT)

As mentioned before, the input information to the PP Attachment Table (PAT) comes from the syntactic and semantic analyses of prepositions. The function of PAT is specifying the attachment conditions idiosyncratic to each preposition. This section aims to explain the function and details of PAT.

Firstly, some of the semantic cases, no matter they are subcategorized or not, always modify verbs, rather than the preceding nouns. They are the cases of goal, instrument, end, commitative etc. Other prepositions, such as *like*, *of*, are usually noun modifiers. Since their presence ensure correct attachment, we devise two markers to show these special propertities: *VPP* to those cases which have to be attached to verbs exclusively; *NPP* to those prepositions that only modify nouns. In PAT, they are represented as:

(on (goal (VPP NIL)) ...)

(like (NIL (NPP NIL)) ...)

For the rest that may be attached to verbs and nouns, such as location, both markers are then assigned to them simultaneously. It is this type of PP which constitute the attachment problem. Further conditions are needed to clarify their status. The conditions come chiefly from verbs. Consider the following sentences:

(17) He read a letter <u>from Jack</u>

(18) He bought a car <u>from Jack</u>

The PP *from Jack* , which plays the role of source, is both optional in (17) and (18). Yet, semantics rules out the attachment to the verb *read* in (17), but not to *bought* in (18). It is because *read* is such a non-locomotion verb that it fails to co-occur with a source case. This kind of knowledge is then specified in PAT for *from* so that for it to be a noun modifier, the verb should carry the feature *-locomotion* These restrictions are represented in PAT as:

(from (source (NPP -locomotion) (VPP NIL)))

The general format for every preposition in PAT is:

(prep_word (case (attachment condition)*)*)*
where *attachment ::= NPP | VPP*
condition::= semantics_of_the_ preceding_verb|
semantics_of_the_ preceding_noun|
semantics_of_the_noun_after_ prep/
sentence type

In conclusion, PAT specifies the attachment conditions for every preposition, including whether a particular case is *VPP* or *NPP*. For those that can be both, further conditions are then provided.

3.3. Parsing Strategy

According to the statistics, 36.4% of English sentences found in the two textbooks include at least one PP, our parser thus takes the problem of PP attachment into serious consideration. Therefore, our discussion of parsing strategy is also subject to PP attachment only. The whole framework can be clearly shown in Figure 1:

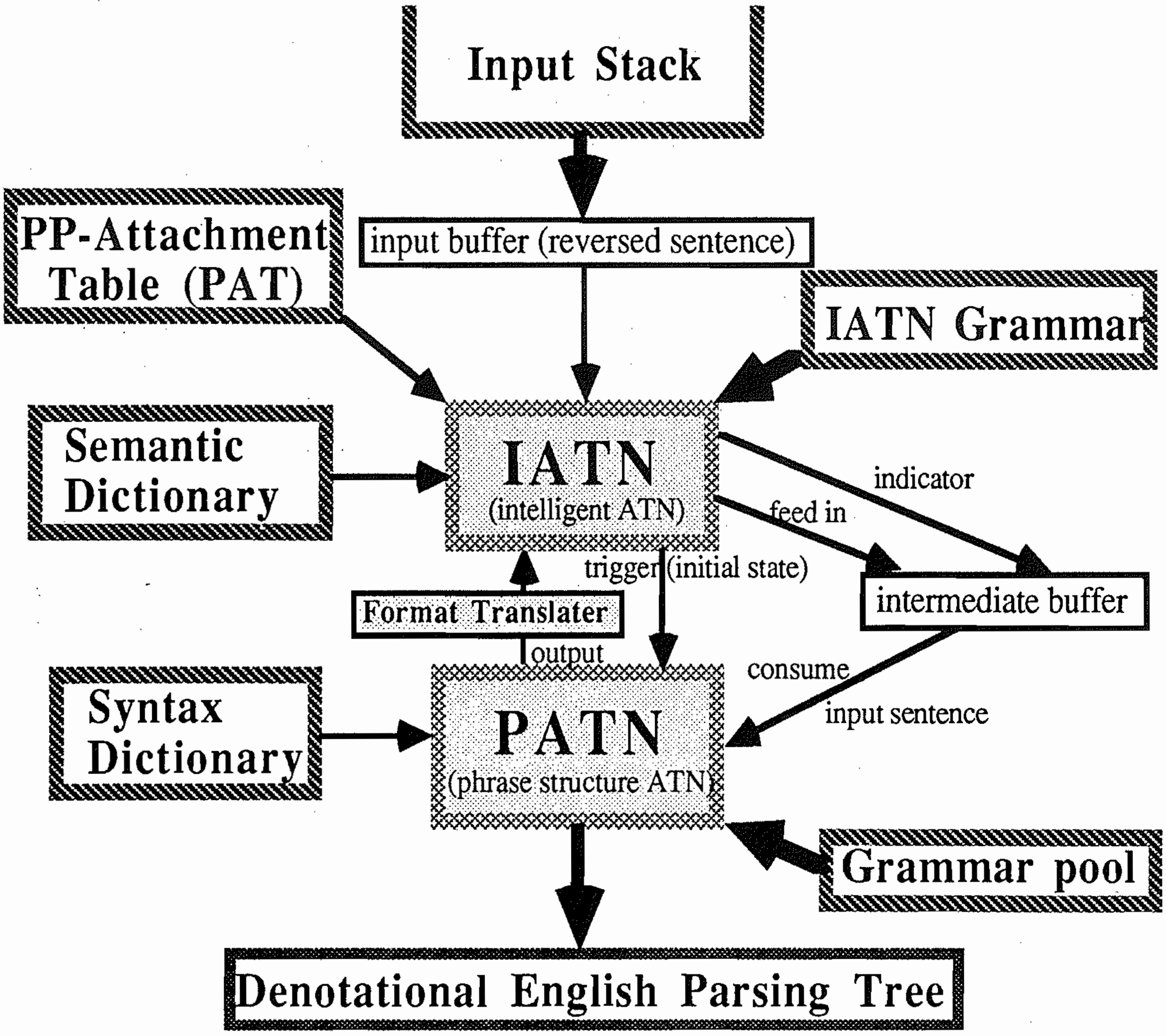


Figure1: the framework of NTUMT system

Simply speaking, our parsing strategy can be thought of as an interaction between two processors -- Intelligent ATN (IATN) and Phrase Structure ATN (PATN). The Input Stack stores all the possible category combinations of the words in the input sentence. It feeds IATN one combination each time. IATN then starts scanning the input combination leftwards, and activating PATN to construct the first bar-level structures (according to the X-bar grammar).

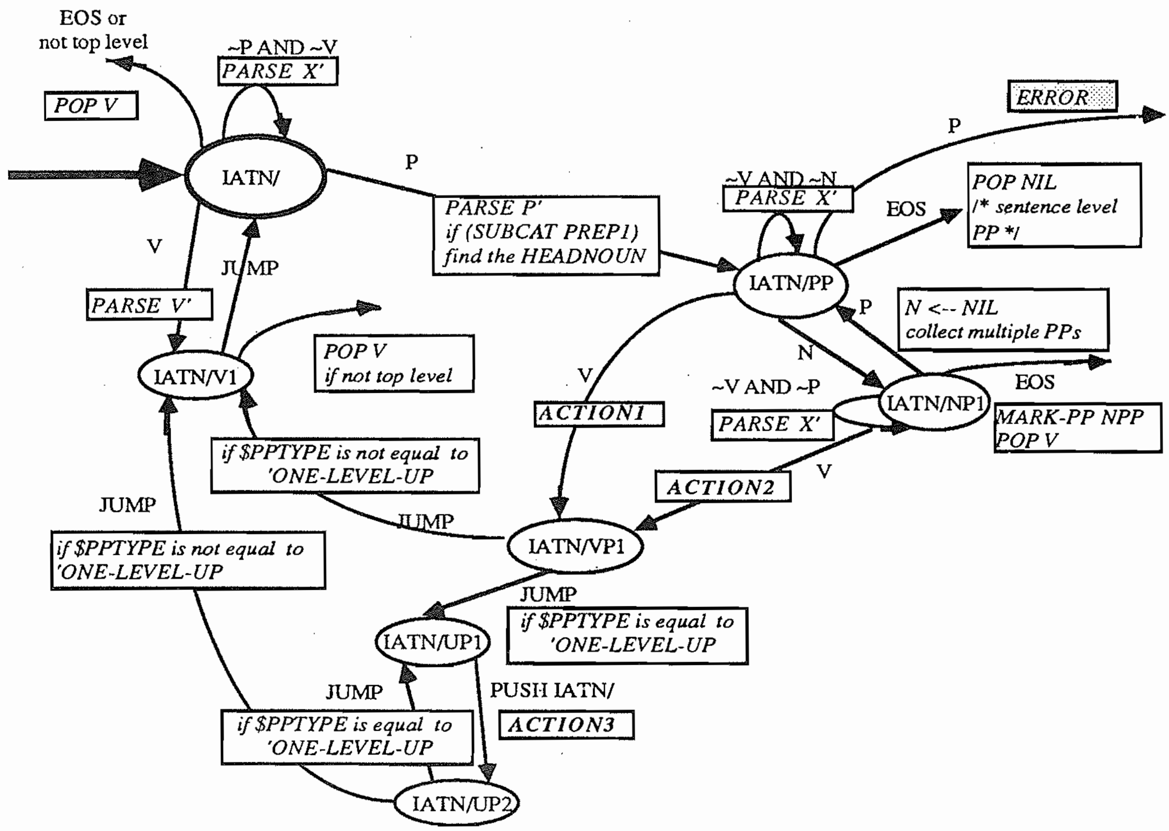
The IATN grammar, which comprises seven states, is capable of solving the problem of PP attachment. They will be discussed individually in the following:

1. The initial state is IATN/, which instructs PATN to build up structures to first bar level. Moreover, whenever a preposition is encountered, it goes to the second state -- IATN/PP.
2. In the state IATN/PP, it tries to find a preceding noun or verb. If none is found, the PP should be the sentence modifier. However, if it meets a noun, the semantic information of the noun will be stored in the register N and then enters into another state IATN/NP1. If it is a verb instead, the function of PP-ATTACHMENT will be called. It searches for the attachment conditions from PAT. The result value, which is either *NPP* or *VPP*, will be added to the PP1/ list in the Intermediate Buffer. Afterwards, it enters IATN/VP1.
3. IATN/NP1 so far includes the information of a prepositional phrase and the noun coming from IATN/PP. When it moves on and finds a preposition, another PP will be grouped together again. The nominal information originally stored in register N will be pushed into the register HEADNOUN, while the prepositional information will be pushed into the register P. In short, the loop -- IATN/PP --> IATN/NP1 --> IATN/PP -- groups as many preposition phrases as possible in the sentence. However, if the

current word is a verb instead, PP-ATTACHMENT will be called. Then, it enters the state of IATN/VP1.

4. In IATN/VP1, in case the verb does not satisfy the attachment conditions, the grammar will go to IATN/UP1 to find another verb of higher-level for attachment. Only after the attachment conditions have been met does it jump to IATN/V1.
5. Under the condition that the PP fails to attach to the verb or the noun of the most proximate clause, IATN/UP1 is then responsible for finding another verb in the higher clause.
6. In IATN/UP2, the information of the verb will be checked against the attachment conditions. If they are still not satisfied, the grammar will go back to IATN/UP1. Thus, IATN/UP1 and IATN/UP2 form a loop until the attachment condition have been satisfied. The grammar then enters the last state-- IATN/V1.
7. The last state is IATN/V1. It either returns the semantic information of the 'qualified' verb to the previous IATN for PP attachment, or goes back to the initial state to process the rest of the words in the sentence.

The complete State Transition Diagram for IATN grammar is shown in Figure 2 below:



<u>*NOTES*</u>

ACTION1 (PP-ATTACHMENT V, NIL, P, HEADNOUN)
(PARSE V")

ACTION2 (PP-ATTACHMENT V, N, P, HEADNOUN)
(PARSE V")

ACTION3 (COND ((* (PP-ATTACHMENT * NIL P H EADNOUN))
(T (MARK-PP VPP)))

Figure 2: State Transition Diagram for IATN grammar

Finally, several examples are provided in the Appendix to show how IATN and PATN solve the PP attachment problem.

4. Conclusion

By the help of the detailed syntactic and semantic analyses of verbs, nouns and prepositions in lexicon, as well as the attachment conditions in PAT, our parsing strategy can make correct PP attachment for the 338 sentences found in the two English textbooks. For those PPs that are semantically ambiguous, like sentence (1), disambiguation relies on contextual information, which is beyond the ability of our NTUMT system. Further research in this area will be needed.

In fine, since our system just provides one result, the ambiguous prepositional phrase will be assigned to the verb.

Appendix

Example 1: **(19) He loved the shirt <u>in the closet</u>**

Format V-NP-PP

IATN/ state	*****	**next_state**	**intermediate buffer after PATN parsing**
IATN/	closet	IATN/	((7 7 (NP1/ (ROOT . <closet>) ...)))
IATN/	the	IATN/	((6 6 (DETP1/ (ROOT . <the>) ...)) (7 7 ...))
IATN/	in	IATN/PP	((5 7 (PP1/ (ROOT . <in the closet>) (SUBCAT PREP1))))
IATN/PP	shirt	IATN/NP1	((4 4 (N (SUBCAT ...))) (5 7 ...))
IATN/NP1	the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>))) (4 4 ...) (5 7 ...))
IATN/NP1	loved	IATN/VP1	((2 2 (V ...)) (3 3 ...) (4 4 ...) (5 7 ... (ATTACH . NPP)))
			/* after IATN call function PP-AGREEMENT */
		==>	((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/VP1	he	IATN/V1	((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/V1	he	IATN/	((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/	he	IATN/	((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 7 ...))
IATN/	EOS		POP

Example 2 : **(20) He met the girl <u>in front of the restaurant</u> <u>at noon</u>**

Format V-NP-PP-PP

IATN/ state	*****	**next_state**	**intermediate buffer after PATN parsing**
IATN/	noon	IATN/	((9 9 (NP1/ (ROOT . <noon>) ...)))
IATN/	at	IATN/PP	((8 9 (PP1/ (ROOT . <at noon>) (CASE time)...)))
IATN/PP	restaurant	IATN/NP1	((7 7 (N (SUBCAT ...))) (8 9 ...)) /* N <-- restaurant; HEADNOUN <-- noon; P <-- at*/
IATN/NP1	the	IATN/NP1	((6 6 (DETP1/ (ROOT . <the>))) (7 7 (N (SUBCAT ...))) (8 9 ...))
IATN/NP1	in_front_of	IATN/PP	((5 5 (P (CASE loc)...)) (6 6 (DETP1/ (ROOT . <the>))) (7 7 (N (SUBCAT ...))) (8 9 ...)) /* N <-- NIL; HEADNOUN <-- (restaurant . noon); P <-- (in_front_of . at) */
IATN/PP	girl	IATN/NP1	((4 4 (N ...)) (5 5 (P ...)) (6 6 (DETP1/ (ROOT . <the>))) (7 7 (N (SUBCAT ...))) (8 9 ...)) /* N <-- girl; HEADNOUN <-- (restaurant . noon); P <-- (in_front_of . at) */
IATN/NP1	the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>) ...)) (4 4 ...) (5 5 ...) (6 6 ...) (7 7 ...) (8 9 ...))
IATN/NP1	met	IATN/VP1	((2 2 (V (ROOT . <meet>) (F))) (3 3 ...)) (4 4 ...) (5 5 (P (CASE loc) (ATTACH . VPP)...)) (6 6 ...) (7 7 ...) (8 9 (PP1/ (ATTACH . VPP)...))) /* after IATN call function PP-AGREEMENT */
		==>	((2 7 (VP1/ (ROOT . <met the girl in_front_of the

			restaurant at noon>))))
IATN/VP1	he	IATN/V1	((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/V1	he	IATN/	((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/	he	IATN/	((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 7 ...))
IATN/	EOS		POP

Example 3: **(21) He saw the money on the desk in the room next to mine**

Format V-NP-PP-PP-PP

IATN/ state	*****	**next_state**	**intermediate buffer after PATN parsing**
IATN/	mine	IATN/	((12 12 (NP1/ (ROOT . <mine>) ...)))
IATN/	next_to	IATN/PP	((11 12 (PP1/ (ROOT . <next_to mine>) (CASE loc)...)))
IATN/PP	room	IATN/NP1	((10 10 (N (SUBCAT ...))) (11 12 ...)) /* N <-- room; HEADNOUN <-- mine; P <-- next_to*/
IATN/NP1	the	IATN/NP1	((9 9 (DETP1/ (ROOT . <the>))) (10 10...) (11 12 ...))
IATN/NP1	in	IATN/PP	((8 8 (P (CASE loc)...)) (9 9...) (10 10 ...) (11 12 ...)) /* N <-- NIL; HEADNOUN <-- (room . mine); P <-- (in . next_to) */
IATN/PP	desk	IATN/NP1	((7 7 (N ...)) (8 8 (P (CASE loc)...)) (9 9...) (10 10 ...) (11 12 ...)) /* N <-- desk; HEADNOUN <-- (room . mine); P <-- (in . next_to) */
IATN/NP1	the	IATN/NP1	((6 6 (DETP1/ (ROOT . <the>))) (7 7 ...) (8 8 ...) (9 9 ...) (10 10 ...) (11 12 ...))
IATN/NP1	on	IATN/PP	((5 5 (P (CASE loc) ...)) (6 6 ...) (7 7 ...) (8 8 (P (CASE loc)...)) (9 9...) (10 10 ...) (11 12 ...)) /* N <-- NIL; HEADNOUN <-- (desk room . mine); P <-- (on in . next_to) */

IATN/PP	money	IATN/NP1	((4 4 (N ...)) (5 5 ...) (6 6 ...) (7 7 ...) (8 8 (P ...)) (9 9...) (10 10 ...) (11 12 ...)) /* N <-- money; HEADNOUN <-- (desk room . mine); P <-- (on in . next_to) */
IATN/NP1	the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>))) (4 4 ...) (5 5 ...) (6 6 ...) (7 7 ...) (8 8 ...) (9 9 ...) (10 10 ...) (11 12 ...))
IATN/NP1	saw	IATN/VP1	((2 2 (V (ROOT . <saw>) (F +perceptual))) (3 3 ...)) (4 4 ...) (5 5 (P (CASE .NPP)...)) (6 6 ...) (7 7 ...) (8 8 (CASE . NPP)...)) (9 9 ...) (10 10 ...) (11 12 (PP1/ (ATTACH . NPP)...))) /* after IATN call function PP-AGREEMENT */
		==>	((2 12 (VP1/ (ROOT . <saw the money on the desk in the room next_to mine>))))
IATN/VP1	he	IATN/V1	((2 12 (VP1/ (ROOT . <saw the money on the desk in the room next_to mine>))))
IATN/V1	he	IATN/	((2 12 (VP1/ (ROOT . < saw the money on the desk in the room next_to mine>))))
IATN/	he	IATN/	((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 12 ...))
IATN/	EOS		POP

References

Bates, M. (1978) "The Theory and Practice of Augmented Transition Network Grammars" Natural Language Communication with Computers (Lecture notes in computer science 63). Berlin: Springer-Verlag, 1978, pp. 191-259

Fass, D. & Wilks, Y. (1983) "Preference Semantics, III-Formedness, and Metaphor" American Journal of Computational Linguistics, vol 9, NO. 3-4, Jul-Dec 1983, pp. 178-187

Fodor, J.D. & Frazier, L. (1980) "Is the human sentence parsing mechanism an ATN?" Cognition, vol 8, 1980, pp. 417-459

Frazier, L. & Fodor, J.D. (1978) "The sausage machine: a new two-stage parsing model" Cognition, 6 (4), Dec 1978, pp. 291-325

Hirst, G. (1983) "A Foundation for Semantic Interpretation" Proc. of ACL-83, Cambrdige, Massachusetts, June 1983, pp. 64-73

Hirst, G. (1984) "A Semantic Process for Syntactic Disambiguation" Proc. of AAAI-84, Austin, pp. 148-152

Huang, X. (1987) "XTRA: The Design and Implementation of a Fully Automatic Machine Translation" PhD. Thesis

Lin, Yia-ping. (1989) "Embedding X' Theory in ATN System to Solve PP Attachment Problem" M.S. Thesis, National Taiwan University.

Schubert, L.K. "Are There Preference Trade-off in Attachment Decisions?" Proc. of AAAI-86, pp. 601-605

Shieber, S.M. (1983) "Sentence Disambiguation by a Shift-Reduced Parsing Technique" Proc. of IJCAI-83, Kahlsruhe, W.Germany, pp. 699-703

Wanner, E. (1980) "The ATN and the Sausage Machine: Which one is Balony?" Cognition, vol. 8, 1980, pp. 209-225

Wilks, Y.A., Huang, H.H. & Fass, D.C. (1985) "Syntax, Preference and Right Attachment" Proc. of IJCAI 85, los Angeles, California, pp. 779-784

Woods, W.A. (1969) "Augmented Transition Networks for Natural Language Analysis" Harvard Computational Laboratory Report No. CS-1, Harvard University, Cambrdige, MA., 1969

Woods, W.A. (1970) "Transition Networks Grammar for Natural Language Analysis" Communications of the ACM. 13 (1970), pp. 591-606

Woods, W.A. (1973) "An Experimental Parsing System for Transition Network Grammars" Natural Language Processing, Randall Rustin, ed., New York: Algorithmics Press, 1973