DEALING WITH CONJUNCTIONS IN A MACHINE TRANSLATION ENVIRONMENT

Xiuming HUANG
Institute of Linguistics
Chinese Academy of Social Sciences
Beijing, China*

ABSTRACT

The paper presents an algorithm, written in PROLOG, for processing English sentences which contain either Gapping, Right Node Raising (RNR) or Reduced Conjunction (RC). The DCG (Definite Clause Grammar) formalism (Pereira & Warren 80) is adopted. The algorithm is highly efficient and capable of processing a full range of coordinate constructions containing any number of coordinate conjunctions ('and', 'or', and 'but'). The algorithm is part of an English-Chinese machine translation system which is in the course of construction.

0 INTRODUCTION

Theoretical linguists have made a considerable investigation into coordinate constructions (Ross 67a, Hankamer 73, Schachter 77, Sag 77, Gazdar 81 and Sobin 82, to name a few), giving descriptions of the phenomena from various perspectives. Some of the descriptions are stimulating or convincing. Computational linguists, on the other hand, have achieved less than their theoretical counterparts.

(Woods 73)'s SYSCONJ, to my knowledge, is the first and the most often referenced facility designed specifically for coordinate construction processing. It can get the correct analysis for RC sentences like

(1) John drove his car through and completely demolished a plate glass window

but only after trying and failing an indefinite number of times, due to its highly non-deterministic nature.

(Church 79) claims "some impressive initial progress" processing conjunctions with his NL parser YAP. Using a Marcus-type attention shift mechanism, YAP can parse many conjunction constructions including some cases of Gapping. It doesn't offer a complete solution to conjunction processing though: the Gapping sentences YAP deals with are only those with two NP remnants in a Gapped conjunct.

(McCord 80) proposes a "more straightforward and more controllable" way of parsing sentences like (1) within a Slot Grammar framework. He treats "drove his car through and completely demolished" as a conjoined VP, which doesn't seem quite valid.

(Boguraev 83) suggests that when "and" is encountered, a new ATN arc be dynamically constructed which seeks to recognise a right hand constituent categorially similar to the left hand one just completed or being currently processed. The problem is that the left-hand conjunct may not be the current or most recent constituent but the constituent of which that former one is a part.

(Berwick 83) parses successfully Gapped sentences like

(2) Max gave Sally a nickel yesterday, and a dime today

using an extended Marcus-type deterministic parser. It is not clear, though, how his parser would treat RC sentences like (1) where the first conjunct is not a complete clause.

The present work attacks the coordinate construction problem along the lines of DCG. Its coverage is wider than the existing systems: both Gapping, RNR and RC, as well as ordinary cases of coordinate sentences, are taken into consideration. The work is a major development of (Huang 83)'s CASSEX package, which in turn was based on (Boguraev 79)'s work, a system for resolving linguistic ambiguities which combined ATN grammars (Woods 73) and Preference Semantics (Wilks 75).

In the first section of the paper, problems raised for Natural Language Processing by Gapping, RNR and RC are investigated. Section 2 gives a grouping of sentences containing coordinate conjunctions. Finally, the algorithm is described in Section 3.

I GAPPING, RIGHT NODE RAISING AND REDUCED CONJUNCTION

1.1 Gapping

Gapping is the case where the verb or the verb together with some other elements in the non-leftmost conjuncts is deleted from a sentence:

(3) Bob saw Bill and Sue [saw] Mary.

-----

* Mailing address: Cognitive Studies Centre,
University of Essex,
Colchester CO4 3SQ, England.

(4) Max wants to try to begin to write a novel, and Alex [wants to try to begin to write] a play.

Linguists have described rules for generating Gapping, though none of them has made any effort to formulate a rule for detecting Gapping. (Ross 67b) is the first who suggested a rule for Gapping. The formalisation of the rule is due to (Hankamer 73):

Gapping
NP X A Z and NP X B Z --> NP X A Z and NP B
where A and B are nonidentical major constituents*.

(Sag 76) pointed out that there were cases where the left peripheral in the right conjunct might be a non-NP, as in

(5) At our house, we play poker, and at Betsy's house, bridge.

It should be noted that the two NPs in the Gapping rule must not be the same, otherwise (7) would be derived from (6):

(6) Bob saw Bill and Bob saw Mary.
(7) Bob saw Bill and Bob Mary.

whereas people actually say

(8) Bob saw Bill and Mary.

When processing (8), we treat it as a simplex containing a compound object ("Bill and Mary") functioning as a unit ("unit interpretation"), although as a rule we treat sentence containing conjunction as derived from a "complex", a sentence consisting of more than one clause, in this case "Bob saw Bill and Bob saw Mary" ("sentence coordination interpretation"). The reason for analysing (8) as a simplex is first, for the purpose of translation, unit interpretation is adequate (the ambiguity, if any, will be "transferred" to the target language); secondly, it is easier to process.

Another fact worth noticing is that in the above Gapping rule, B in the second conjunct could be anything, but not empty. E.g., the (a)s in the following sentences are Gapping examples, but the (b)s are not:

(9) (a) Max spoke fluently, and Albert haltingly.
*(b) Max spoke fluently, and Albert.
(10) (a) Max wrote a novel, and Alex a play.
*(b) Max wrote a novel, and Alex.
(11) (a) Bob saw Bill, and Sue Mary.
(b) Bob saw Bill, and Sue.

Before trying to draw a rule for detecting Gapping, we will observe the difference between (12) and (13) on one hand, and (14) on the other:

(12) Bob met Sue and Mary in London.
(13) I knew the man with the telescope and the woman with the umbrella.
(14) Bob met Sue in Paris and Mary in London.

As we stated above, (12) is not a case of Gapping; instead, we take "Sue and Mary" as a coordinate NP. Nor is (13) a case of Gapping. (14), however, cannot be treated as phrasal coordination because the PP in the left conjunct ("in Paris") is directly dominated by the main verb so that "Mary" is prevented from being conjoined to "Sue".

Now, the Gapping Detecting Rule:

The structure 'NP1 V A X and NP2 B' where the left conjunct is a complete clause, A and B are major constituents, and X is either NIL or a constituent not dominated by A, is a case of Gapping if

```
Gapping if (OR (AND (X = NIL) (B = NP))
               (AND (V = 3-valency verb)*
                    (OR (B = NP) (B = to NP)))
               (AND (X /= NP) (X /= NIL)))**
```

1.2 Right Node Raising (RNR)

RNR is the case where the object in the non-rightmost conjunct is missing.

(15) John struck and kicked the boy.
(16) Bob looked at and Bill took the jar.

RNR raises less serious problems than Gapping does. All we need to do is to parse the right conjunct first, then copy the object over to the left conjunct so that a representation for the left clause can be constructed. Then we combine the two to get a representation for the sentence.

Sentences like the following may raise difficulty for parsing:

(17) I ate and you drank everything they brought. (cf. Church 79)

(17) can be analysed either as a complex of two full clauses, or RNR, according to whether we treat "ate" as transitive or intransitive.

1.3 Reduced Conjunction

Reduced Conjunction is the case where the conjoined surface strings are not well-formed constituents as in

(18) John drove his car through and completely demolished a plate glass window.

where the conjoined surface strings "drove his car through" and "completely demolished" are not well-formed constituents. The problem will not be as

----

* According to the dependency grammar we adopt, we define a major constituent of a given sentence S as a constituent immediately dominated by the main verb of S.

* 3-valency verbs are those which can appear in the structure 'NP V NP NP', such as 'give', 'name', 'select', 'call', etc.
** Here "/=" means "is not".

serious as might have seemed, given our understanding of Gapping and RNR. After we process the left conjunct, we know that an object is still needed (assuming that "through" is a preposition). Then we parse the right conjunct, copying over the subject from the left; finally, we copy the object from the right conjunct to the left to complete the left clause.

II GROUPING OF SENTENCES CONTAINING CONJUNCTIONS

We can sort sentences containing conjunctions into three major groups on the basis of the nature of the left-most conjunct: Group A contains sentences whose left-most conjuncts are recognized by the analyser as complete clauses; Group B, the left-most conjuncts are not complete clauses, but contain verbs; and Group C, all the other cases. The following is a detailed grouping with example sentences:

A1. (Gapping) Clause-internal ellipsis:
(19) I played football and John tennis.
(20) Bob met Sue in Paris and John Mary in London.
(21) Max spoke fluently and Albert haltingly.

A2. (Gapping) Left-peripheral ellipsis with two NP remnants:
(22) Max gave a nickel to Sally and a dime to Harvey.
(23) Max gave Sally a nickel and Harvey a dime.
(24) Jack calls Joe Mike and Sam Harry.

A3. (Gapping) Left-peripheral ellipsis with one NP remnant and some non-NP remnant(s):
(25) Bob met Sue in Paris and Mary in London.
(26) John played football yesterday and tennis today.

A4. (Gapping) Right-peripheral ellipsis concomitant with clause-internal ellipsis:
(27) Jack begged Elsie to get married and Wilfred Phoebe.
(28) John persuaded Dr. Thomas to examine Mary, and Bill Dr. Jones.
(29) Betsy talked to Bill on Sunday, and Alan to Sandy.

A5. The right conjunct is a complete clause:
(30) I played football and John watched the television.

A6. The right conjunct is a verb phrase to be treated as a clause with the subject deleted:
(31) The man kicked the child and threw the ball.

A7. Sentences where the "unit interpretation" should be taken:
(32) Bob met Sue and Mary in London.
(33) I knew the girl bitten by the dog and the cat.

B1. Right Node Raising:
(34) The man kicked and threw the ball.
(35) The man kicked and the woman threw the ball.

B2. Reduced Conjunction:
(36) John drove his car through and completely demolished a plate glass window.

C. Unit interpretations:
(37) The man with the telescope and the woman with the umbrella kicked the ball.
(38) Slowly and stealthily, he crept towards his victim.

III THE ALGORITHM

The following algorithm, implemented in PROLOG Version 3.3 (shown here in much abridged form), produces correct syntactico-semantic representations for all the sentences given in Section 2. We show here some of the essential clauses* of the algorithm: 'sentence', 'rest_sentencel' and 'sentence_ conjunction'. The top-most clause 'sentence' parses sentences consisting of one or more conjuncts. In the body of 'sentence', we have as sub-goals the disjunction of 'noun_phrase' and 'noun_phrasel', for getting the sentence subject; the disjunction of '[W], is_verb' and 'verbl', plus 'rest_verb', for treating the verb of the sentence; the disjunction of 'rest_sentence' and 'rest_ sentencel' for handling the object, prepositional phrases, etc; and finally 'sentence_conjunction', for handling coordinate conjunctions.

The Gapping, RNR and RC sentences in Section II contain deletions from either left or right conjuncts or both. Deleted subjects in right conjuncts are handled by 'noun_phrasel' in our program; deleted verbs in right conjuncts by 'verbl'. The most difficult deletions to handle (for previous systems) are those from the left conjuncts, ie. the deleted objects of RNR (Group B1) and the deleted preposition objects of RC (Group B2), because when the left conjuncts are being parsed, the deleted parts are not available. This is dealt with neatly in PROLOG DCG by using logical variables which stand for the deleted parts, are "holes" in the structures built, and get filled later by unification as the parsing proceeds.

```
sentence(Stn, P_Subj, P_Subj_Head_Noun, P_Verb,
    P_V_Type, P_Contentverb, P_Tense,
    P_Obj, P_Obj_Head_Noun) -->
      % P_ means "possible": P_ arguments only
      % have values if 'sentence' is called by
      % 'sentence_conjunction' to parsea second
      % (right) conjunct. Those values will be
      % carried over from the left conjunct.
   (noun_phrase(Subj, Head_Noun);
   noun_phrasel(P_Subj, P_Subj_Head_Noun, Subj,
    Head_Noun)),
      % 'noun_phrasel' copies over the subject
      % from the left conjunct.
  adverbial_phrase(Adv),
  ([W],
      % W is the next lexical item.
   is_verb(W,Verb,Tense);
      % Is W a verb?
  verbl(P_Verb, Verb, P_Contentverb, Contentverb,
     P_Tense, Tense, P_V_Type, V_Type)),
      % 'verbl' copies over the verb from the
      % leftconjunct.
```

----

* A 'clause' in our DCG comprises a head (a single goal) and a body (a sequence of zero or more goals).

```
  rest_verb(Verb,Tense,Verbl,Tensel),
      % 'rest_verb' checks whether Verb is an
      % auxiliary.
  (rest_sentence(dcl,Subj,Head_Noun,Verbl, V_Type,
     Contentverb,Tensel,Obj, Obj_Head_Noun, P_Obj,
     P_Obj_Head_Noun, Indobj, S);
      % 'rest_sentence' handles all cases but RC.
  rest_sentencel(dcl,Subj,Head_Noun,Verbl, V_Type,
     Contentverb,Tensel, Obj, Obj_Head_Noun,
     P_Obj, P_Obj_Head_Noun, Indobj, S)),
      % 'rest_sentencel' handles RC.
  sentence_conjunction(S, Stn, Subj, Head_Noun,
     Verbl, V_Type, Contentverb, Tensel, Obj,
     Obj_Head_Noun).

rest_sentencel(Type, Subj,Head_Noun,Verbl, V_Type,
    Contentverb, Tense, Prep_Obj,Prep_Obj_Head_
    Noun,       P_Obj, P_Obj_Head_Noun, Indobj,
    s(type(Type),   tense(Tense), v(Verb_sense,
    agent(Subj),    object(Obj), post_verb_
    mods(prep(Prep),   prep_obj(Prep_Obj)))) -->
      % Here Prep_Obj is a logical variable which
      %will be  instantiated later when  the
      % right conjunct has been parsed.
  {verb_type(Verb, V_Type)},
  complement(V_Type, Verb, Contentverb, Subj,
     Head_Noun, Obj, Obj_Head_Noun, P_Obj,
     P_Obj_Head_Noun, v(Verb_sense, agent(Subj),
     object(Obj),     post_verb_mods(prep(W),
     prep_obj(Prep_Obj))),
      % The  sentence object is processed and  the
      % verb structure built  here.
  [W],
  {preposition(W)}.

sentence_conjunction(S,s(conj(W), S, Sconj), Subj,
     Head_Noun, Verbl, V_Type, Verb2, Tense, Obj,
     Obj_Head_Noun)  -->
  ([','], [W]; [W]),
  {conj(W)},
      % Checks whether W is a conjunction.
  sentence(Sconj, Subj, Head_Noun, Verbl,  V_Type,
  Verb2, Tense, Obj, Obj_Head_Noun).
      % 'sentence' is called recursively to parse
      % right conjuncts.

sentence_conjunction(S, S, _, _, _, _, _, _, _, _)
     --> [].     % Boundary condition.
```

For sentence (36) ("John drove his car through and completely demolished a plate glass window"), for instance, when parsing the left conjunct, 'rest_sentencel' will be called eventually. The following verb structure will be built: v(drovel,agent(np(pronoun(John))), object(np(det (his), pre_mod([]), n(carl), post_mods([]))), post_verb_mods(prep_mods(prep(through), prep_obj(Prep_Obj)), where the logical variable Prep_Obj will be unified later with the argument standing for the object in the right conjunct (ie, "a plate glass window"). When 'sentence' is called via the subgoal 'sentence_conjunction' to process the right conjunct, the deleted subject "John" will be copied over via 'noun_phrasel'. Finally a structure is built which is a combination of two complete clauses. During the processing little effort is wasted. The backward deleted constituents ("a plate glass window" here) are recovered by using logical variables; the forward deleted ones ("John" here) by passing over values (via unification) from the conjunct already processed. Moreover, the 'try-and-fail' procedure is carried out in a controlled and intelligent way. Thus a high efficiency lacking in many other systems is achieved (space prevents us from providing a detailed discussion of this issue here).

ACKNOWLEDGEMENTS

I would like to thank Y. Wilks, D. Arnold, D. Fass and C. Grover for their comments and instructive discussions. Any errors are mine.

BIBLIOGRAPHY

Berwick, R. C. (1983) "A deterministic parser with broad coverage." Bundy, A. (ed), Proceedings of IJCAI 83, William Kaufman, Inc.

Boguraev, B. K. (1979) Automatic Resolution of Linguistic Ambiguities. Technical Report No. 11, University of Cambridge Computer Laboratory, Cambridge.

Boguraev, B. K. (1983) "Recognising conjunctions withing the ATN framework." Sparck-Jones, K. and Wilks, Y. (eds), Automatic Natural Language Parsing, Ellis Horwood.

Church, K. W. (1980) On Memory Limitations in Natural Language Processing. MIT. Reproduced by Indiana Univ. Ling. Club, Bloomingtong, 1982.

Gazdar, G. (1981) "Unbounded dependencies and coordinate structure," Linguistic Enquiry, 12: 155 - 184.

Hankamer, J. (1973) "Unacceptable ambiguity," Linguistic Inquiry, 4: 17-68.

Huang, X-M. (1983) "Dealing with conjunctions in a machine translation environment," Proceedings of the Association for Computational Linguistics European Chapter Meeting, Pisa.

McCord, M. C. (1980) "Slot grammars," American Journal of Computational Linguistics, 6:1,31-43.

Pereira, F. & Warren, D. (1980) "Definite clause grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks," Artificial Intelligence, 13: 231 - 278.

Ross, J. R. (1967a) Constraints on Variables in Syntax. Doctoral Dissertation, MIT,Cambridge, Massachusetts. Reproduced by Indiana Univ. Ling. Club, Bloomington, 1968.

Ross, J. R. (1967b) "Gapping and the order of constituents," Indiana Univ. Ling. Club, Bloomington. Also in Bierwisch, M. and K. Heidolph, (eds), Recent Developments in Linguistics, Mouton, The Hague, 1971.

Sag, I. A. (1976) Deletion and Logical Form. Ph.D. thesis, MIT, Cambridge, Mass.

Schachter, P. (1977) "Constraints on coordination," Language, 53: 86 - 103.

Sobin, N. (1982) "On gapping and discontinuous constituent structure," Linguistics,20:727-745.

Wilks, Y. A. (1975) "Preference Semantics," Keenan (ed), Formal Semantics of Natural Language, Cambridge Univ. Press, London.

Woods, W. A. (1973) "A experimental parsing system for Transition Network Grammar," Rustin, R. (ed), Natural Language Processing, Algorithmic Press, N. Y.