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

(i) John drove his car through and
completely demolished a plate glass window
but only after trying and failing an indefinite
tonumber 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.

(Huang 83) 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 categorically 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.
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):

\[
\text{Gap pl ng NP X A Z and NP X B Z } \rightarrow \text{ NP X A Z and NP B}
\]

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

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):

Bob saw Bill and Bob saw Mary.

(7) Bob saw Bill and Bob Mary.

whereas people actually say

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:

(a) Max spoke fluently, and Albert haltingly.
(b) Max spoke fluently, and Albert.

(a) Max wrote a novel, and Alex a play.
(b) Max wrote a novel, and Alex.

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 (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.

John struck and kicked the boy.
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:

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

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

* 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".

* 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.
seriously 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 haltingly.

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 syntactic-semantic representations for all the sentences given in Section 2. We show here some of the essential clauses* of the algorithm: 'sentence', 'rest_sentence' and 'sentence_conjunct'. 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_phrase', for getting the sentence subject; the disjunction of '[W]', is_verb' and 'verb1', plus 'rest_verb', for treating the verb of the sentence; the disjunction of 'rest_sentence' and 'rest_sentence' for handling the object, prepositional phrases, etc; and finally 'sentence_conjunct', for handling coordinate conjuncts.

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_phrase' in our program; deleted verbs in right conjuncts by 'verb1'. The most difficult deletions to handle (for previous systems) are those from the left conjuncts, i.e. the deleted objects of RNR (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.

The following algorithm, implemented in PROLOG Version 3.3 (shown here in much abridged form), produces correct syntactic-semantic representations for all the sentences given in Section 2. We show here some of the essential clauses* of the algorithm: 'sentence', 'rest_sentence' and 'sentence_conjunct'. 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_phrase', for getting the sentence subject; the disjunction of '[W]', is_verb' and 'verb1', plus 'rest_verb', for treating the verb of the sentence; the disjunction of 'rest_sentence' and 'rest_sentence' for handling the object, prepositional phrases, etc; and finally 'sentence_conjunct', for handling coordinate conjuncts.

The following algorithm, implemented in PROLOG Version 3.3 (shown here in much abridged form), produces correct syntactic-semantic representations for all the sentences given in Section 2. We show here some of the essential clauses* of the algorithm: 'sentence', 'rest_sentence' and 'sentence_conjunct'. 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_phrase', for getting the sentence subject; the disjunction of '[W]', is_verb' and 'verb1', plus 'rest_verb', for treating the verb of the sentence; the disjunction of 'rest_sentence' and 'rest_sentence' for handling the object, prepositional phrases, etc; and finally 'sentence_conjunct', for handling coordinate conjuncts.

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_phrase' in our program; deleted verbs in right conjuncts by 'verb1'. The most difficult deletions to handle (for previous systems) are those from the left conjuncts, i.e. the deleted objects of RNR (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(Srn, P_Sbj, P_Sbj_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_conjunct' to parse second % (right) conjunct. Those values will be % carried over from the left conjunct.
  % 'noun_phrase' copies over the subject % from the left conjunct.
  % 'noun_phrase' 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?
  verb1(P_Verb, Verb, P_ContentVerb, ContentVerb, P_Tense, Tense, P_V_Type, V_Type),
  % 'verb1' copies over the verb from the % left conjunct.

* A 'clause' in our DCG comprises a head (a single goal) and a body (a sequence of zero or more goals).
For sentence (36) ("John drove his car through and completely demolished a plate glass window"), for instance, when parsing the left conjunct, 'rest_sentence' will be called eventually. The following verb structure will be built:

\[ \text{return verb}(\text{Verb}, \text{Tense}, \text{Verb1}, \text{Tense1}) \]

\[ \% 'rest verb' checks whether Verb is an auxiliary. \]

\[ \text{return sentence}(\text{dc1}, \text{Subj}, \text{Head Noun}, \text{Verb1}, \text{V Type}, \text{Contentverb}, \text{Tense}, \text{Obj}, \text{Obj Head Noun}, \text{P Obj}, \text{P Obj Head Noun}, \text{Indobj}, \text{S}); \]

\[ \% 'rest_sentence' handles all cases but RC. \]

\[ \text{return sentence}(\text{dc1}, \text{Subj}, \text{Head Noun}, \text{Verb1}, \text{V Type}, \text{Contentverb}, \text{Tense}, \text{Obj}, \text{Obj Head Noun}, \text{P Obj}, \text{P Obj Head Noun}, \text{Indobj}, \text{S}); \]

\[ \% 'rest_sentence' handles RC. \]

\[ \text{sentence conjunction}(\text{S}, \text{Stn}, \text{Subj}, \text{Head Noun}, \text{Verb1}, \text{V Type}, \text{Contentverb}, \text{Tense}, \text{Obj}, \text{Obj Head Noun}); \]

\[ \% 'sentence' conjunction(S, S, _, _, _, _, _, _, _, _); \]

\[ \% Checks whether W is a conjunction. \]

\[ \text{sentence conjunction}(\text{S}, \text{S conj}, \text{Subj}, \text{Head Noun}, \text{Verb1}, \text{V Type}, \text{Contentverb}, \text{Tense}, \text{Obj}, \text{Obj Head Noun}) \rightarrow (\text{[', ']}, \text{[WT; W]}, \text{[conj(W)]}); \]

\[ \% 'sentence' is called recursively to parse right conjuncts. \]

\[ \% 'sentence' conjunction(S, S, _, _, _, _, _, _, _, _); \]

\[ \% Boundary condition. \]

For sentence (36) ("John drove his car through and completely demolished a plate glass window") 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).