FILLING GAPS IN PARSED ENGLISH SENTENCES

COY C. DAY, JR.

SEPTEMBER 1989   AI89-113
Filling Gaps in Parsed English Sentences

Coy C. Day, Jr.
The University of Texas at Austin
September 29, 1989

Abstract

Gaps occur in natural language when sentence constituents are omitted as a result of movement rules, subjectless clause constructions, and compound phrases. In order to properly represent the information contained in a sentence, such gaps must be filled before the final, logical representation of the sentence is created. This paper examines the use of purely syntactic methods for eliminating gaps which exist after parsing. Syntactic graphs, rather than parse trees, are used to represent the surface structure of the sentences. It is shown that certain gaps can be filled using only syntactic information. Rules are presented for handling these gaps. There are other gaps, however, which cannot be filled syntactically. Hopefully, knowledge about the world or the context of the sentence can be utilized to fill these gaps.

*Support for this research was provided by the U.S. Army Research Office under contract DAAG-84-K-0060
1 Introduction

A natural language processing system can be considered to have at least these two phases: parsing and semantic processing. The output of the parser is a set of syntactic relations among surface constituents of the input sentence. These relations are then used to create an internal representation of the sentence. Somewhere in this process, anaphors that are present in the surface structure must be resolved before the representation is useful to other processes.

This paper considers gaps as a form of anaphora and develops a method for replacing them with their intended referents. As an example, consider sentence (1).

(1) He is the man I'm waiting for.

This sentence contains a relative clause that is missing the object of its preposition as a result of wh-movement. Before the final representation is created, man must be attached to for to give the clause its intended meaning. Clearly, this gap can be seen as a null anaphor which must be resolved by including its antecedent in the logical representation.

In Section 4, we will examine a variety of instances in which gaps occur and attempt to find syntactic rules which resolve each gap, or at least aid in its resolution. Before that, however, we must discuss gaps in more detail and also present the representation used in this paper for the surface structure of a sentence. Section 2 is a limited survey of the linguistics literature on gaps. Section 3 describes the syntactic graph, the compact representation of all possible parses of a sentence which this paper utilizes. Such a graph greatly simplifies the removal of gaps (known as degapping) and the notation of degapping rules. Our conclusions are presented in Section 5.

2 Gaps

The linguistics literature considers each type of gap in its own setting. For example, gaps that result from relativization are handled by wh-movement [1]. Thus, the structure of the NP in sentence (1) would be represented as

(2) [NP man [s I am waiting for —]].

The claim is that the wh-pronoun (covert in this example) is moved from its underlying position (—) to its surface position, producing a gap. In this paper, a relative
clause gap is considered to be filled not by the relative pronoun but by the pronoun's antecedent. In this way, a level of indirection is avoided. For example, the gap in (1) is filled by man rather than the implied whom, since whom would then have to be marked as referring to man.

Gaps which result from clauses with empty subjects (e.g. infinitive phrases) are described differently [3]. The gap in (3) results from its transformation to a passive sentence.

(3) He is believed to be right.

The structure of (3) is

(4) He is believed [— to be right].

As a result of passivization, the subject of the infinitive phrase has been moved to the front of the sentence, leaving a gap.

Another common structure is illustrated by (5).

(5) John persuaded Mary to resign.

In this case, Mary is considered to be the object of the main clause and not the subject of the subordinate clause. For this reason, a different representation is required for the structure of (5):

(6) John persuaded Mary [PRO to resign].

Since Mary is the object of persuaded, it is not included in the subordinate clause. The symbol PRO is used instead, indicating that there is an empty pronoun which must have an antecedent. This use of PRO resembles the characterization of gaps in this paper as null anaphors which must be resolved.

3 Syntactic Graphs

The syntactic graph is a recently devised method for representing the surface structure of a sentence [4]. Although parse trees, the structures normally used to describe surface structure, correspond to only one parse of a sentence, the syntactic graph represents in a compact form all possible syntactic relations in a single structure. This compact representation contains all the necessary syntactic information that is found in the collection of all parse trees of the sentence. Also, syntactically ambiguous points
are marked so that semantic processes can focus on those points when attempting to disambiguate the sentence.

Syntactic graphs consist of dominator-modifier relationships [5] among the constituents of X grammars [2], providing a functional description of a sentence. The nodes of a syntactic graph are the words of the sentence,\(^1\) and labeled arcs (triples which consist of a label and two nodes) describe relationships among them. Since a syntactic graph is represented by a set of triples, processes have direct access to the information in the graph and are not forced to navigate the entire structure, as is required for parse trees. In this way, the graph makes it easy to detect gaps by the presence of “target” arcs, triples that signal the presence of a particular gap and contain the information needed to fill it.

The syntactic graph of a sentence is created using the chart formed by an all-paths parser. Interpretations not allowed by the grammar are prevented using an exclusion matrix. The purpose of this matrix is to store the co-occurrence constraints among the arcs.\(^2\) Grammar rules for the parse are in augmented phrase structure form, but are written in such a way that the deviation from context-free grammar is kept to a minimum.

The graph of sentence (1), repeated here as (7), illustrates the syntactic graph notation.

(7) He is the man I’m waiting for.

The graph includes the following triples:

1. (snip, be, he)
2. (subcomp, be, man)
3. (det, man, the)
4. (relclause, man, wait)
5. (aux, wait, be)

\(^1\)This is, of course, a simplification. Actually, each node stores the root form of the word along with other lexical information and the word’s position in the sentence. In addition, if a word functions as more than one part of speech in multiple interpretations of the same sentence, it will be represented by a different node for each possible part of speech.

\(^2\)These constraints are necessary due to information that is lost in converting from the parse forest to this more compact form. In a parse forest, one can determine whether two constituents can co-occur by checking all parse trees one by one. A syntactic graph, since it contains of all possible constituents, makes it difficult to determine whether triples can co-occur. This is accomplished by the exclusion matrix [4].
6. (snp, wait, I)
7. (sntp, wait, for)

Each triple consists of the arc name followed by the dominator node and then the modifier node, the node that is dominated. Each arc label indicates the relationship between the two nodes. The snp label marks the relationship between the head verb of a sentence or clause and the NP that serves as its surface subject. The first (dominator) node is the head verb of the clause and the second (modifier) node is the head noun of the NP. In the graph of (7), the first triple indicates that the surface subject of the sentence is he. The sentence is represented by its head verb, be.

The other triples are interpreted similarly. The subcomp label is used to represent the subject complement of a sentence. Thus, the dominator node is the head verb of the sentence and the modifier node is the head word of the phrase that acts as subject complement. In the second triple, be is used as before to represent the sentence and man is recorded as the subject complement. The det label is used to attach a determiner to the phrase it modifies. Thus, the triple (det, man, the) records the as a determiner modifying man. In the fourth triple, a relclause label marks the relative clause as modifying man. The head verb of the clause, wait, is used to represent it. Next, the aux label attaches the auxiliary be to wait. The sixth triple, (snp, wait, I), indicates that the surface subject of the relative clause is I.

In the final triple, the sntp arc name is a special label included in the notation to allow for a sentence or clause which ends in a preposition. This arc is special because it acts as a flag that signals the presence of a particular kind of gap. In essence, the task of detecting the gap has been carried out already by the parser and now the gap is filled easily.3

4 Degapping Rules

The gaps considered in this paper satisfy the following description:

A gap is a deleted constituent that either (1) is subordinate to a verb other than the main verb of the sentence or (2) is part of a phrase that is conjoined with the phrase containing its referent.

3There are many arc names that did not appear in this example. Others will be described as they are introduced.
Gaps of the first type are often found in relative clauses and infinitive and participial phrases. These structures introduce gaps because they do not require all the components of an independent clause. The second class of gaps results from the presence of conjunctions, as in compound NPs and compound VPs. A modifier of a compound NP, for example, might be incorrectly parsed as modifying only one of the conjoined phrases.

Although there is a choice of when to remove gaps, this paper discusses removing them immediately after parsing. One reason for doing so is to make available all the information possible to later processes. With these gaps filled early, the role of later semantic processes is deciding which of the possible parses to eliminate, rather than creating additional meanings after parsing has been completed. Also, it is cheaper to degap without searching semantic information. Purely syntactic transformations are quick and precise, especially when syntactic graphs are used.

In the following pages, we examine syntactic rules for degapping and their ability to handle various occurrences of gaps. By considering the arcs present in syntactic graphs which contain gaps, rules for filling such gaps are developed. Each rule checks for the presence of target triples that indicate a particular gap. If required, a transformation is then applied to the triples that represent the sentence, asserting new triples to fill the gap and deleting existing triples which degapping has made unnecessary.

\subsection{Relative Clauses}

Relative clauses are the simplest structures to degap, since the parse of the sentence is all the information needed to fill the gap. Once a relative clause has been parsed, the referent must be the noun modified by the clause. The only remaining task is to determine how the gap fits into the syntactic structure. This is discovered by examining the syntactic graph of the clause.

Consider our earlier example, in which the gap is the object of a preposition.

\begin{quote}
(8) He is the man I'm waiting for.
\end{quote}

Rule 1 detects this gap and removes it by asserting the noun modified by the relative clause as the object of the preposition.

\textbf{Rule 1} If the syntactic graph includes triples that match (relclause, X, Y) and (sntp, Y, Z), then assert (upp, Y, Z) and (ppn, Z, X) and delete (relclause, X, Y) and (sntp, Y, Z) — and (relpro, Y, -) if present.
The notation here is meant to resemble Prolog; the underscore matches anything. The \texttt{relpro} triple records the relative pronoun used by attaching it to the relative clause. It is removed, if present, because it is not needed in the semantic representation. Its deletion will also prevent it from being detected by other degapping rules, causing undesired behavior.

The \texttt{vpp} label indicates an adverbial PP. The dominator node is the verb modified and the modifier node is the preposition. The \texttt{ppn} marks a PP and includes the preposition followed by the object of the preposition.

In our example, \texttt{(relclause, man, wait)} and \texttt{(sntp, wait, for)} are the target triples, because they indicate a relative clause ending in a preposition. Using the nodes in the target triples (e.g. \texttt{X}, \texttt{Y}, and \texttt{Z}), the rule then fills the gap by including in the graph the proper relationships among these nodes and the existing nodes. In this case, \texttt{(vpp, wait, for)} and \texttt{(ppn, for, man)} are added to eliminate the gap. The new graph resulting from the application of Rule 1 is this set of triples: \texttt{(snp, be, he)}, \texttt{(subcomp, be, man)}, \texttt{(det, man, the)}, \texttt{(snp, wait, I)}, \texttt{(aux, wait, be)}, \texttt{(vpp, wait, for)}, \texttt{(ppn, for, man)}.

The effect of this rule can be demonstrated in terms of a parse tree by translating from the syntactic graph representation. The parse of the original sentence is shown in Figure 1. The parse of the degapped sentence is actually two trees (Figure 2), since removing \texttt{(relclause, man, wait)} eliminated the dependency of the clause on \texttt{man}.

A relative clause missing the object of a preposition may be preceded rather than followed by the preposition, as in (9).

(9) That is the area from which it came.

The syntactic graph of (9) contains \texttt{(snp, be, that)}, \texttt{(subcomp, be, area)}, \texttt{(det, area, the)}, \texttt{(relclause, area, come)}, \texttt{(relprep, come, from)}, \texttt{(relpro, come, which)}, \texttt{(snp, come, it)}. Here, the \texttt{relprep} triple marks the preposition which introduces the clause. The rule for degapping such clauses is a modification of Rule 1.

\textbf{Rule 2} \textit{If the syntactic graph includes \texttt{(relclause, Z, X)}, \texttt{(relprep, X, Y)}, and \texttt{(relpro, X, W)}, then assert \texttt{(vpp, X, Y)} and \texttt{(ppn, Y, Z)} and delete \texttt{(relprep, X, Y)}, \texttt{(relclause, Z, X)}, and \texttt{(relpro, X, W)}.}

The resulting degapped graph for (9) is \texttt{(snp, be, that)}, \texttt{(subcomp, be, area)}, \texttt{(det, area, the)}, \texttt{(snp, come, it)}, \texttt{(vpp, come, from)}, \texttt{(ppn, from, area)}. As before, this graph represents two parse trees.
Figure 1: Parse tree before degapping

Figure 2: Parse trees after degapping
In another type of relative clause, the gap is the object of the head verb of the clause. Consider the following sentence:

(10) He returned the markers that he borrowed.

The initial syntactic graph in this case contains these triples: (snp, return, he), (vnp, return, marker), (det, marker, the), (relclause, marker, borrow), (relpro, borrow, that), (snp, borrow, he). This graph is degapped by the following rule.

**Rule 3** If the graph includes (relclause, X, Y) and (snp, Y, _) and not (relprep, Y, _) or (sntp, Y, _), then assert (vnp, Y, X) and delete (relclause, X, Y) — and (relpro, Y, _) if present.

The triples (relprep, Y, _) and (sntp, Y, _) are excluded in order to prevent this rule from applying to the previous gaps. This is required because those gaps could also include (relclause, X, Y) and (snp, Y, _), causing an incorrect transformation to be carried out on the graph. The resulting graph for (10) is (snp, return, he), (vnp, return, marker), (det, marker, the), (snp, borrow, he), (vnp, borrow, marker).

The gap may also replace the subject of the relative clause. Consider (11), which has a syntactic graph representation of (snp, like, she), (vnp, like, child), (relclause, child, behave), (relpro, behave, who), (vmod, behave, nicely).

(11) She likes children who behave nicely.

This gap is filled by Rule 4.

**Rule 4** If the graph includes (relclause, X, Y) and (relpro, Y, Z) and not (snp, Y, _), (relprep, Y, _), or (sntp, Y, _), then assert (snp, Y, X) and delete (relclause, X, Y) and (relpro, Y, Z).

The resulting graph for (11) is (snp, like, she), (vnp, like, child), (snp, behave, child), (vmod, behave, nicely).

The relative participle, illustrated by (12), is similar to the relative clause.

(12) The woman wearing a sweater plays in the band.

The syntactic graph of this sentence includes the following triples: (snp, play, woman), (det, woman, the), (nvp, woman, wear), (vnp, wear, sweater), (det, sweater, a), (vpp, play, in), (ppn, in, band), (det, band, the). The nvp triple is the target triple for such gaps; it denotes a verb phrase that modifies a noun. In this case, *wearing a sweater* is a VP modifying *woman*. 

9
All that is required to degap this sentence is that the triple \((nvp, \text{woman}, \text{wear})\) be replaced by \((snp, \text{wear}, \text{woman})\). This new triple signals the presence of another sentence and reflects that \text{wear} dominates \text{woman}. The transformation to be used is given by Rule 5, which degaps all relative participles.

**Rule 5** If the graph includes \((nvp, X, Y)\), then delete \((nvp, X, Y)\) and replace it with \((snp, Y, X)\).

Other relative clauses are created using the possessive \textit{whose}.

(13) She is the woman whose book I borrowed.

These clauses pose no new difficulties and are easily degapped using variations of the first four rules above, using the NP following \textit{whose} to fill the gap.

The rules above apply to all relative clauses, whether restrictive or nonrestrictive. Nonrestrictive clauses should be marked in the graph so they can be treated properly during later processing. This may be done by adding the triple (nonrestrictive, \(X\), yes), where \(X\) is the head verb of the clause. Although the relative clause gaps are easily and syntactically removed, there are not many other gaps for which this is true. Certain infinitive gaps, however, are nearly syntactic in nature.

### 4.2 Infinitive Phrases

Infinitive phrases create gaps because they lack subjects. For example, in sentence (14) \textit{Joan} is the missing subject of \textit{go}. In (15), \textit{Jim} is the subject.

(14) Joan wants to go to the opera.

(15) Joan wants Jim to go to the opera.

The syntactic graph of (14) contains \((snp, \text{want}, \text{Joan}), (\text{vinf}, \text{want}, \text{go}), (\text{inf-to}, \text{go}, \text{to}), (\text{vpp}, \text{go}, \text{to}), (\text{ppn}, \text{to}, \text{opera}), (\text{det}, \text{opera}, \text{the})\). The \textit{vinf} triple indicates that \textit{go} is an infinitive verb that follows \textit{want}. \((\text{inf-to}, \text{go}, \text{to})\) includes in the graph the \textit{to} preceding the infinitive \textit{go}. As this sentence illustrates, when a verb complement infinitive phrase is not preceded by the direct object of the sentence, the subject of the sentence serves as the subject of the infinitive. Rule 6 applies this information.

**Rule 6** If the syntactic graph includes \((\text{vinf}, X, Z)\) and \((snp, X, Y)\) and not \((\text{vnp}, X, -)\), then assert \((snp, Z, Y)\).
Thus, the graph of sentence (14) after degapping includes the new triple (snp, go, Joan).

Before degapping, the graph of (15) includes (snp, want, Joan), (vnp, want, Jim), (vinf, want, go), (inf.to, go, to), (vpp, go, to), (ppn, to, opera), (det, opera, the). For this and other sentences which include an object as well as a verb complement infinitive phrase, the object serves as subject of the infinitive. Thus, Rule 7 applies.

**Rule 7** *If the graph includes (vinf, X, Z) and (vnp, X, Y), then assert (snp, Z, Y).*

According to this rule, (snp, go, Jim) is added to the graph of (15).

As mentioned before, these gaps cannot be removed as consistently as those in relative clauses. For example, Rule 6 usually fails on sentences with certain head verbs, such as be or said.

(16) The goal tonight is to eat dinner before the restaurants close.

(17) The doctor said to take these three times a day.

Such gaps cannot be filled given only the parse of the sentence. In (16), for example, (snp, eat, goal) is incorrectly added to the graph, indicating that the subject of eat is goal. In (17), (snp, take, doctor) is asserted, as if the doctor is taking the medicine. To prevent errors such as these, constraints must be added to Rule 6.

Rule 7 occasionally fails because the subject of the sentence may fill the gap even though a direct object is present.

(18) He promised Mary to go to the store.

(19) He killed three guards to escape.

In both (18) and (19), He is the subject of the infinitive.

One might consider solving this problem by partitioning the verbs in the lexicon into two groups. The first group would contain verbs whose subject serves as subject of the infinitive phrase. The second would consist of those verbs whose object fills the gap. However, (20) and (21) reveal that such a scheme could not always work. In both sentences, the main verb is the same. Yet, the sentences require different referents for the gap.

(20) The group lobbied Congress to get the lawmakers’ attention.

(21) The group lobbied Congress to pass the new tax bill.
Proper interpretation of these sentences requires more than surface structure; there is no escaping semantic analysis. The complete resolution of the gap must be deferred. However, syntax can indicate the possible fillers for these gaps. In (20) and (21), the gap must be filled by either the subject or object of the main verb. Therefore, both can be presented to the semantic processes as possibilities, reducing later computation.

So far, this discussion has included only those infinitive phrases which function as verb complements. Other infinitive phrases are more difficult or even impossible to degap syntactically, as in (22) through (26).

(22) He needs a place to sleep.

(23) He gave Tom a place to sleep.

(24) Dan’s only requirement is a place to sleep.

(25) To go on a hike is tiring.

(26) To go on a hike is tiring for John.

Sentences (22), (23), and (24), which include adjectival infinitive phrases, demonstrate the difficulty of such phrases. These sentences can be degapped only after the semantic representation is created. (25) and (26) contain infinitive phrases as subjects. Although certain structures (e.g. for John in 26) may indicate how the gap is to be filled, such infinitives often have implied subjects. In (25), the subject gap is filled by everyone.

Infinitive gaps are much like participle gaps, since both constructs usually appear without a subject and have similar function. Just as some infinitive phrases are easier to degap, so are certain participles. In particular, we will consider those which act as adverbs.

### 4.3 Adverbial Participles

Participles which act as adverbs generally take the subject of the verb they modify as their own subject, as in (27). In this case, *thinking the party was over* is an adverbial phrase modifying *decided*. Clearly, *John* is the subject.

(27) John decided to leave, thinking the party was over.
Adverbial participles preceded by a preposition, however, may take as their subject either the subject or object of the verb they modify.

(28) John punished the children for breaking dishes.

(29) John punished the children by reducing their allowances.

Both (28) and (29) contain gaps due to the participles’ missing subjects. In (28) the gap is filled by the object of the main verb, while in (29) it is filled by the subject. Since the structure of these two sentences is identical, these gaps are clearly more difficult than those of relative clauses. Also, since the main verb is the same in both sentences, it cannot indicate how to fill the gap. The only syntactic solution to this problem is to let the preposition preceding the participle indicate the subject.

The effect of the preposition can seen in (30) and (31), in which the prepositions in (28) and (29) have been exchanged.

(30) John punished the children by breaking dishes.

(31) John punished the children for reducing their allowances.

Changing the prepositions has altered the meaning. The use of the preposition by in (30) has made John the subject of break. Similarly, the use of for in (31) now has the children reducing their own allowances. The strongest statement that can be made concerning the effect of the preposition on the choice of subject is this: only for, from, and about may precede the participle when the object of the main verb fills the gap. The following two rules are required to add the appropriate triples.

**Rule 8** If the syntactic graph includes \((snp, X, Y), (vpp, X, Z), \) and \((ppv, Z, V), \) then assert \((snp, V, Y).\)

**Rule 9** If the graph includes \((vnp, X, Y), (vpp, X, Z), \) and \((ppv, Z, V), \) and \(Z\) is either for, from or about, then assert \((snp, V, Y).\)

To see the effect of these rules, consider sentence (28). Before degapping, its graph is \((snp, punish, John), (vnp, punish, child), (det, child, the), (vpp, punish, for), (ppv, for, break), (vnp, break, dish).\) The \(ppv\) triple indicates that the object of the preposition is a verb form. In this case, \(break\) is the object of \(for\). The result of applying Rule 8 is that the triple \((snp, break, John)\) is added. Since Rule 9 applies,
(snp, break, child) is possible as well. The decision between the two can be made only after further processing.

A similar ambiguity arises in other participle structures. For example, when the participle follows a conjunction such as when or while, either the subject (e.g. (32)) or object (e.g. (33)) of the sentence may fill the gap.

(32) John sneezes when discussing cats.

(33) John helps children when crossing the street.

Other participles are more difficult to degap. Most gerunds, for example, are impossible to degap syntactically.

4.4 Gerunds

Except for trivial cases, as in (34), there are no syntactic rules for degapping gerunds.

(34) Jim's riding his bike causes panic in the streets.

Although rules could be written for certain narrow uses of gerunds, there would have to be many such rules. It is more desirable to postpone degapping until later. (35) through (39) illustrate some of the problems associated with degapping gerunds.

(35) Riding a bike is fun.

(36) Riding his bike was fun.

(37) The thought of riding a bike scared Jim.

(38) Tom thought riding his bike was fun.

(39) Riding a bike was fun for Tom.

These sentences illustrate a few of the places that the subject of a gerund may be found. In sentence (35), the subject of riding is understood to be everyone. The interpretation of (36), however, depends on context. For example, the following sequence implies that the gap is to be filled by John.

(40) John rode his bike all day. Riding his bike was fun.

However, the sequence below yields another meaning.
(41) Mary rode John's bike all day. Riding his bike was fun.

Now, the gap is filled by Mary.

Sentences (37) through (39) show that even when the subject is present in the sentence it may be located in any number of places. These are all reasons why syntactic processing is insufficient for degapping gerunds.

4.5 Compound VPs

Compound NP and compound VP gaps are different from the others examined here because they result from the presence of conjunctions rather than subordinate clauses. An advantage of this is that the referent is always present and is simple to find: it modifies the phrase conjoined with the phrase in which the gap occurs. A disadvantage, however, is that it is never clear, syntactically, whether there is actually a gap. Consider, for example, (42) and (43).

(42) He sang and played his guitar.

(43) He washed and rinsed the dishes.

Since (42) and (43) are identical structurally, they will be degapped in the same way: the object of the second verb is attached to the compound VP as well. In (42), this causes his guitar to be incorrectly asserted as the object of the compound VP. However, in (43) the dishes is appropriately added as the object of the compound VP. In both cases, ambiguity is added to the parse which must be resolved later.

Rule 10 degaps sentences like (43), in which an object follows a compound VP. The graph of (43) illustrates the representation of compound phrases in syntactic graphs. It includes these triples: (snp, and, he), (conjvp, and, wash), (conjvp, and, rinse), (vnp, rinse, dish), (det, dish, the). The triple (snp, and, he) indicates that he is the subject of a compound VP in which and is the conjunction that joins the components of the VP. These components are given by the next two triples, (conjvp, and, wash) and (conjvp, and, rinse), creating a hierarchy among the triples. These conjvp triples simply record the conjunction and as dominating the VPs whose heads are wash and rinse. Triples which modify the compound VP as a whole are attached to the and node. Those which modify only one of the conjoined phrases are attached to the head verb of that phrase.

The desired transformation, therefore, is to add to the graph a new arc that attaches the object of the final conjoined VP to the conjunction joining the VPs.
In (46), it is ambiguous whether cold modifies only rivers or rivers and streams, but this ambiguity is not represented in its parse: (snp, swim, Jim), (vpp, swim, in), (ppn, in, and), (conjnp, and, river), (mod, river, cold), (conjnp, and, stream). The solution, given by Rule 13, is to include the second interpretation as well, so that later processes can choose between the two. Thus, Rule 13 adds (mod, and, cold) to the graph.

**Rule 13** If the graph includes (conjnp, X, Y), (conjnp, X, Z), and (mod, Y, W), and Y precedes Z, then assert (mod, X, W).

Rule 14 degaps compound NPs followed by PPs, as in (47).

**Rule 14** If the graph includes (conjnp, X, Y), (conjnp, X, Z), and (npp, Z, W), and Y precedes Z, then assert (npp, X, W).

Rules 13 and 14 may add some triples that are clearly incorrect, increasing the work done by the semantic processes. This is because the rules do not check for additional modifiers as did the rules for compound VPs presented above. (48) through (50) illustrate the problem.

(48) American rivers and Canadian streams are beautiful.

(49) Green rivers and streams in America are beautiful.

(50) America's green rivers and clear streams are beautiful.

(48) does not contain a gap and yet Rule 13 applies and so will assert that American modifies rivers and Canadian streams as well as rivers. Unfortunately, we cannot check for additional modifiers as in Rules 10, 11 and 12 since such modifiers are allowed in compound NPs even when a gap exists. This is apparent in (49) and (50). Green may modify streams in (49) even though streams is modified by the PP in America. Also, the PP may modify rivers although rivers is modified by Green. In this way, extra triples are added that must be removed later.

5 Conclusions

In this paper, we have examined certain gaps that occur in English sentences. Our goal was to find syntactic transformations that would fill the gap in the syntactic
graph representation of the sentence. Such transformations modify the graph by deleting existing triples and asserting new ones. A gap is detected by the presence of certain target triples and is filled using the information stored in those triples.

From this examination of various gaps, it is now clear that there are some — namely, those in relative clauses — which can be removed using only syntactic information. There are other gaps which syntactic information cannot resolve with certainty, but which may be aided by syntactic information since the possible referents for the gap can be determined syntactically. Examples are the gaps in infinitive phrases, adverbial participles, and compound phrases.

There are still other gaps to which no syntactic transformations apply. Gerunds, for example, are used in so many ways that any resolution of the gaps they contain must rely on semantics, information from elsewhere in the text, or general world knowledge.

For those gaps that can be removed syntactically, the syntactic graph notation simplifies the process by presenting the structure of a sentence in such a way that it is easy to detect gaps and retrieve the information needed to fill them. The target triples are easily detected, and only simple transformations are required.

For the gaps that cannot be resolved syntactically, other methods have been developed. These methods cannot remove all remaining gaps, however; more research is required in this area.

References


