Semantic Interpretation of Unrealized Syntactic Material in LTAG

Olga Babko-Malaya
University of Pennsylvania
malayao@ldc.upenn.edu

Abstract

This paper presents a LTAG-based analysis of gapping and VP ellipsis, which proposes that resolution of the elided material is part of a general disambiguation procedure, which is also responsible for resolution of underspecified representations of scope.

1 Introduction

The problem of ellipsis resolution is to recover the interpretation of the elided material. For example, in (1), the elided VP is interpreted as being identical to the verb in the preceding sentence. Likewise, in the gapping structures, as shown in (2), the interpretation of a gap is being identified with the interpretation of the preceding verb.

(1) Mary likes Bill. Jane does too.
(2) Mary ate beans and others -- rice.

Whereas some approaches assume syntactic identity between the antecedent and the elided material (e.g. Fiengo and May 1994), others suggest that VP ellipses are proforms, semantically identified with their antecedents (see Dallymple et al 1991, Shieber et al 1996, Hardt 1993, 1999).

This paper follows semantic approaches to ellipsis resolution. It adopts the LTAG semantics of Kallmeyer and Romero 2004 and proposes that resolution of ellipses and gaps is part of a general disambiguation procedure, which is also responsible for resolution of underspecified representations of scope.

2 LTAG Semantics with Semantic Unification

In LTAG framework (Joshi and Schabes 1997), the basic units are (elementary) trees, which can be combined into bigger trees by substitution or adjunction. LTAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. Given that derivation steps in LTAG correspond to predicate-argument applications, it is usually assumed that LTAG semantics is based on the derivation tree, rather than the derived tree (Kallmeyer and Joshi 2003).

Semantic composition which we adopt is based on LTAG semantics with semantic unification (Kallmeyer and Romero 2004). In the derivation tree, elementary trees are replaced by their semantic representations and corresponding feature structures. Semantic representations are as defined in Kallmeyer and Joshi 2003, except that they do not have argument variables. These representations consist of a set of formulas (typed λ-expressions with labels) and a set of scope constraints.

Each semantic representation is linked to a feature structure. Feature structures, as illustrated by different examples below, include a feature i whose values are individual variables and features p and MaxS, whose values are propositional labels. Semantic composition consists of feature unification. After having performed all unifications, the union of all semantic representations is built.

Consider, for example, the semantic representations and feature structures associated with the elementary trees of the sentence shown in (3).

(3) Mary dates Bill

```
S
    NP       VP
      [i: v1] date       NP
                 [i: v2]

NP  [i: x] Mary(x)

NP  [i: y] Bill(y)
```
Derivation tree: 

```
       date
       1       2
      mary            bill
  l4: student(x)  l  4 ≤ R2
  l3: some(y, R 3, N 3)
  l5: course(y)  l 5 ≤ R3
l1: like(x, y)   l 1 ≤ N2   l 1 ≤ N3
```

Semantic composition proceeds on the derivation tree and consists of feature unification:

(4) \( l_1: \text{date(v}_1, \text{v}2 \) )

```
  1 [i: v1]
  2 [i: v2]
```

Performing two unifications, \( v_1=x, v_2=y \), we arrive at the final interpretation of this sentence: \( l_1: \text{date(x, y), bill(y, mary(x). This representation is interpreted conjunctively, with free variables being existentially bound.} \)

Quantificational NPs are analyzed as multi-component TAGs, where the scope part of the quantifier introduces the proposition containing the quantifier, and the predicate-argument part introduces the restrictive clause (see Kallmeyer and Joshi 2003).

(5) \( S^* \)

```
  S
  \[ NP [i: x, p: P_1] \]
  \[ every student \]
  \[ l_1: \text{student(x)} \]
  \[ l_1 \leq R_2, P_1 \leq N_2 \]

  \[ NP [i: y, p: P_3] \]
  \[ some(y, R_3, N_3) \]
  \[ l_1: \text{course(y)} \]
  \[ l_1 \leq R_3 \]
  \[ l_1: \text{like(v}_1, \text{v}2 \) ]

  \[ i: \text{like(v}_1, \text{v}2 \) ]
  \[ p_1: i: \text{v}1 \]
```

The final representation of this sentence is underspecified for scope, given that there are no constraints which restrict the relative scope of every and some. In order to obtain one of the readings, a disambiguation mapping is needed:

**Disambiguations:**

1. \( R_2 \rightarrow l_1, R_3 \rightarrow l_1, N_2 \rightarrow l_1, N_3 \rightarrow l_1; \)
   \( \text{some(y, course(y), every(x, student(x), like(x, y)))} \)
2. \( R_2 \rightarrow l_1, R_3 \rightarrow l_1, N_2 \rightarrow l_1, N_3 \rightarrow l_1; \)
   \( \text{every(x, student(x), some(y, course(y), like(x, y)))} \)

Disambiguations are functions from propositional variables to propositional labels that respect the scope constraints, such that after having applied this mapping, the transitive closure of the resulting scope is a partial order.

3 The Problem of Ellipsis Resolution in LTAG semantics

Given LTAG semantics, there are two possible approaches to resolution of the elided material: reconstruction can be done as part of the unification process or as part of the disambiguation procedure. If reconstruction was done as unification, the semantic representation of the elided material would be disambiguated in the final representation. On the other hand, it is well known that resolution of ellipses and gaps can be ambiguous. For example, the sentence in (6), discussed in Siegel 1987 and Johnson 2003 among others, has 2 interpretations:

(6) **Ward can’t eat caviar and his guests -- dried beans**

Can’t (eat (ward, caviar)) & eat (his guests, dried beans))
Can’t (eat(ward, caviar)) & can’t (eat(his guests, dried beans))

As this example shows, the gap in (6) can be reconstructed by selecting either the verb or the negated modal as its antecedent. The two interpretations represent different scope readings between the conjunction and negation, which should be analyzed as underspecified in LTAG semantics. Resolution of gaps, therefore, cannot be done as part of unification, since it depends on the disambiguated interpretation. The question is whether it is possible to define an underspecified representation of these two readings, and what kind of resolution mechanism can be used to disambiguate these interpretations?

---

1 Other cases of ambiguous interpretations of the elided material are discussed in section 7.
4 LTAG Semantics of Gapping

In LTAG semantics, semantic representations are introduced by lexicalized trees. In order to account for the analysis of gapping and VP ellipsis, this paper proposes that semantics should be defined on both lexicalized and non-lexicalized trees. Specifically, we propose that

Interpretation of a gap (or elided VP) is the semantic interpretation of a non-lexicalized S tree.

The semantic representations of lexicalized S trees under this new approach are derived compositionally, given the meaning of a non-lexicalized S tree and the meaning of a verb.

\[ (7) \quad S \]

\[
NP[v_1] \quad VP \quad V \quad [Ag: v_3, Pat: v_4, MaxS: C_1] \]

\[
NP[v_2] \quad [Ag: v, Pat: u, MaxS: C] \]

\[
\lambda u \lambda v. C \quad v_1(v) \quad \lambda u \lambda v. C \quad v_2(v) \]

Non-lexicalized trees introduce a propositional label and a propositional variable, illustrated by \( l_1 \) and \( C \) above. If a tree is a transitive S-tree, there are two lambda bound variables, which correspond to the Agent and Patient features of the verb. Performing feature unifications (\( v_1=v, v_2=u, C=C_1=C \)) and scope constraint disambiguations (\( C \leq l_0 \)), the proposition \( l_1 \) will be reduced to:

\[
\lambda u \lambda v. date(v, u)(v_1)=date(v_1, v_2)
\]

Given this proposal, we suggest that the semantics of gaps, VPE and other types of ellipsis are reduced to can’t modal, and propositions \( l_2 \) are unified with the proposition \( l_2 \) yields the final representation, where \( l_2 \) and \( l_3 \) are underspecified. There is only one disambiguation of the variable \( C \) in this sentence: \( C \rightarrow l_0 \), which gives us the desired interpretation of the sentence:

\[ (8) \quad Ward \ can’t \ eat \ caviar \ and \ his \ guests \ -- \ dried \ beans. \]

Resolution of the gap in this sentence is enforced by the feature structure of ‘and’, which unifies MaxS as well as Agent and Patient features. This analysis therefore accounts for the fact that gapping “is intimately entangled with the syntax of coordination (as opposed to VP ellipsis)” (Johnson 2003). On the other hand, as the next example illustrates, it is crucial that propositional variables introduced by non-lexicalized trees are not unified during semantic composition, but rather are identified with their antecedents as part of the disambiguation procedure.

\[ (7) \quad Mary \ ate \ beans \ and \ others \ -- \ rice. \]

\[
S \quad [p: l_2, Ag: v_1, Pat: u_1, MaxS:C_1] \]

\[
\lambda u \lambda v. C \quad \lambda u \lambda v. C \quad \lambda u \lambda v. C \quad \lambda u \lambda v. C \]

\[
\lambda u \lambda v. eat(v, u) \quad \lambda u \lambda v. eat(v, u) \quad \lambda u \lambda v. eat(v, u) \quad \lambda u \lambda v. eat(v, u) \]

Final Representation:

\[
1_2: l_2 \land 1_3 \quad l_2: \lambda u \lambda v. C \quad l_1: \lambda u \lambda v. C \quad l_0: \lambda u \lambda v. date(v, u) \quad l_0 \leq C \quad l_0 \leq C \quad l_0 \leq C \]

mary(x), beans(y), others(z), rice(w)
5 LTAG Analysis of VP Ellipsis

The analysis of gapping presented above can be easily extended to the analysis of VP ellipsis. VPE differs from gapping in that it is not restricted to coordinated structures. Whereas in the examples above resolution of gaps was enforced by the feature structure of ‘and’, in the case of VPE, a similar unification, forced by pragmatic constraints, results in recovering the elided material.

As the example in (9) illustrates, our analysis of VPE assumes the following modification of the semantics of non-lexicalized trees: propositions introduced by non-lexicalized trees have one lambda-bound variable, so that each argument is introduced by a separate proposition. For example, the interpretation of a transitive tree below has two propositions \( l_1 \) and \( l_2 \), and two propositional variables \( C_1 \) and \( C_2 \). The proposition \( l_1 \) corresponds to the meaning of a VP, which is missing in the standard TAG-based analyses. This decomposition of the meaning of a non-lexicalized tree, therefore, can be independently motivated by the existence of modifiers which predicate of VPs. We further assume that the MaxS feature of the S tree corresponds to the variable introduced by the agent (or the highest-ranked argument).

(9) Mary likes Bill. Jane does too.

Resolution of gaps under this analysis is done as part of the scope resolution procedure on underspecified representations. A crucial feature of this analysis is that the propositions \( l_1 \) and \( l_2 \) are ‘underspecified’ in the final representation and the variable \( C \) is computed during the disambiguation, i.e. when all scope ambiguities are being resolved. In this respect this analysis differs from previous approaches, where the final representation did not include any variables, except for the arguments of quantifiers or other scopal elements.\(^2\)

\(^2\) However, see Babko-Malaya 2004, where a similar analysis is proposed to account for the semantics of coordinated structures with quantified NPs.
Now consider the second sentence: Jane does too:

\[
\begin{array}{c}
S \quad \text{[Ag: v, MaxS: C_1]} \\
NP[v_3] \quad \text{VP} \\
| \quad l_1: \lambda v_3, C_3 (v_3) \\
NP \quad V \\
\end{array}
\]

Final Representation:

This sentence introduces an intransitive tree and one propositional variable \(C_3\). This variable is not constrained within the sentence, and parallel to other pro-forms, it gets its interpretation from the previous discourse. Specifically, the interpretation of the second sentence is derived by unification of the S features of the second and the first S-trees in (9): \(C_3=C_1\), \(v_3=v\). Given that \(C_1\) is mapped to \(l_1\) above, it corresponds to the proposition being reconstructed: \(C_1(=C_1) \rightarrow l_2\)

\(l_1: \lambda v. \text{like}(v, u) (r) = \text{like}(r, u)\)

### 6 Scope Parallelism

Many previous approaches impose parallelism constraints on the interpretation of the elided material (e.g. Fox 2000, Asher et al 2001 among others). Under the present analysis, scope parallelism comes for free. Consider, for example, the following sentence discussed in Dalrymple et al 1991, among others, where ambiguity is resolved in the same way in both the antecedent and at the ellipsis site: John gave every student a test, and Bill did too. The final interpretation of the first sentence is given in (10) and has 2 possible disambiguations.

\[
\begin{array}{c}
l_0: \text{give}(v, u, w) \\
l_1: \lambda v. C_1 (x) \\
l_2: \lambda u. C_2 (y) \quad l_2 \leq C_1 \\
l_1: \lambda w. C_3 (z) \quad l_1 \leq C_2 \\
l_1: \text{every}(y, R_7, N_7) \quad l_1: \text{some}(z, R_3, N_3) \\
l_1: \text{student}(y) \quad l_1: \text{test}(z) \quad \text{john(x)} \\
l_0 \leq C_1, l_0 \leq N_4, l_0 \leq N_7, l_0 \leq R_1, l_0 \leq R_4 \\
\end{array}
\]

The surface reading (every >> some) is derived by the following mapping: \(C_3 \rightarrow l_0, C_2 \rightarrow l_3, R_7 \rightarrow l_5, N_7 \rightarrow l_3, C_1 \rightarrow l_2, R_3 \rightarrow l_3, N_3 \rightarrow l_3, C_1 \rightarrow l_2, R_5 \rightarrow l_3, N_5 \rightarrow l_3,\)

The interpretation of the second sentence is derived by unifying the S-features of the S-trees (as shown in the previous section). As the result, the variables \(C_3\) and \(v_3\) are unified with the variables \(C_1\) and \(v\). Given that \(C_1\) is being mapped to the proposition \(l_1\) above, \(C_1\) is being reconstructed as the proposition \(\text{every}(y, \text{student}(y), \text{some}(x, \text{test}(x), \text{give}(v, y, z)))\) and \(l_1\) corresponds to the desired reading of this sentence:

\[
\begin{array}{c}
l_1: \lambda v. \text{like}(v, u) (r) = \text{like}(r, u) \\
l_1: \text{every}(y, \text{student}(y), \text{some}(x, \text{test}(x), \text{give}(v, y, z))) (r) = \text{every}(y, \text{student}(y), \text{some}(x, \text{test}(x), \text{give}(v, y, z))) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
\end{array}
\]

The inverse reading (where \text{some}>>\text{every}) can be obtained by the following mapping \(C_3 \rightarrow l_0, C_2 \rightarrow l_3, R_7 \rightarrow l_5, N_7 \rightarrow l_3, C_1 \rightarrow l_2, R_3 \rightarrow l_3, N_3 \rightarrow l_3, l_1: \text{give}(v, y, z)\)

\[
\begin{array}{c}
l_1: \text{every}(y, \text{student}(y), \text{give}(v, y, z)) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
\end{array}
\]

Now, when the second sentence is interpreted, \(C_3\) is unified with \(C_1\), which is being mapped to \(l_1: C_1(=C_1) \rightarrow l_2\). The proposition \(l_1\), then, is reduced to: \(\lambda v. \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) (r) = \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(r, y, z)))\)

As this example illustrates, scope parallelism follows from the present analysis, given that \(C_3\) is unified with a disambiguated interpretation of a VP. It can also be shown that the wide scope puzzle (Sag 1980), shown in (12) is not unexpected under this approach, however, the analysis of this phenomenon is beyond the scope of this paper.\(^3\)

\[
\begin{array}{c}
l_1: \lambda v. \text{like}(v, u) (r) = \text{like}(r, u) \\
l_1: \text{every}(y, \text{student}(y), \text{some}(x, \text{test}(x), \text{give}(v, y, z))) (r) = \text{every}(y, \text{student}(y), \text{some}(x, \text{test}(x), \text{give}(v, y, z))) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
l_1: \text{some}(x, \text{test}(x), \text{every}(y, \text{student}(y), \text{give}(v, y, z))) \\
\end{array}
\]

\[^3\text{As Hirschbuhler 1982, Fox 2000 among others noted, there are constructions where subjects of VPE can have narrow scope relative to nonsubjects. For example, the sentence A Canadian flag was hanging in front of every building. An American flag was too has a reading in which each building has both an American and a Canadian flag standing in front of it. The existence of such readings does not present a problem for the present analysis, if we adopt an analysis of quantificational NPs proposed in Babko-Malaya 2004.}\]
7 Antecedent Contained Deletion (ACD)

Further evidence for the proposed analysis comes from sentences with ACD, discussed in Sag 1980, Egg and Erk 2001, Asher et al 2001, Jacobson (to appear), and illustrated in (13):

(13) John wants Mary to read every book Bill does.

The elided material in this sentence is understood as either "Bill reads" or "Bill wants Mary to read". Given that 'want' and 'every' can take different scope, four possible readings are expected. However, puzzling in this case is the unavailability of one of these readings: "John wants that for every book that Bill wants Mary to read, she reads it. Let us consider the final interpretation of this sentence:

\[
\begin{align*}
I_4 & : \text{want(v}_0, N_4) \quad I_0 : \lambda v_0, C_2(\text{want} \ r) \\
I_1 & : \text{read(v, u)} \quad I_2 : \lambda v, C_1(\text{read} \ x) \\
I_3 & : \text{\& book(y) \& I}_3 \\
I_4 & : \text{every(y, R}_5, N_5) \quad I_1 : \lambda v_1, C(\text{z}) \\
\text{mary(x), john(r), bill(z)} \\
\text{I}_1 \leq C_1, I_3 \leq C_2, I_4 \leq N_5, I_4 \leq R_5, I_1 \leq N_4
\end{align*}
\]

The non-lexicalized S tree introduces a proposition I1 and variables C and v3. These variables can be unified with either S features of the 'read'-tree (i.e. C1 and v), or S features of the 'want'-tree (i.e. C2 and v0). In the first case, the small ellipsis interpretation is derived, and both scope readings are available: C = C1, v3 = v C/C1 -> I1, C2 -> I1, I2: read(x, y) every >> want:

\[
\begin{align*}
N_5 \rightarrow I_0, C_2 \rightarrow I_1, N_4 \rightarrow I_1, R_5 \rightarrow I_0 \\
& \text{I_3: every(y, book(y) \& read(z, y) \& want(r, read(x, y))} \\
& \text{I}_1 \rightarrow I_1, I_2 \rightarrow I_2, I_0: \text{want}(r, \text{every(y, book(y) \& read(z, y) \& read(x, y)))}
\end{align*}
\]

If C and v3 are unified with S features of the 'want'-tree, then the large ellipsis interpretation is derived: C = C2, v3 = v0 C/C2 -> I1, N4 -> I2, C4 -> I1, C1 -> I1, I3: want(r, read(x, y)), I3: want(z, read(x, y))

The reading where every >> want is derived by the following constraints: N4 -> I0, C1 -> I1, R3 -> I5 I2: every(y, book(y) \& want(z, read(x, y)), want(r, read(x, y)) I0

The fourth possible reading, where want >> every, however, is predicted to be unavailable under the present assumptions. This reading, want(r, every(y, book(y) \& want(z, read(x, y)), read(x, y)), cannot be derived, since it requires the proposition I3 to be 'inserted' within the proposition I0.

References

Asher N., D. Hardt and J.Busquets. 2001. Discourse Parallelism, Scope, and Ellipsis", Journal of Semantics, 18(1), pp. 1-25.

Babko-Malaya O. 2004 “LTAG Semantics of NP-Coordination" in Proc. of TAG+7, Vancouver

Dalrymple M., S.Shibeer and F.Pereira 1991. Ellipsis and Higher-Order Unification. In Linguistics and Philosophy 14.

Egg, M. and K.Erk 2001. A compositional approach to VP ellipsis 8th International Conference on HPSG, Trondheim, Norway

Fiengo R. and R. May 1994. Indices and Identity. MIT Press, Cambridge

Fox, D. 2000 Economy and Semantic Interpretation, MIT Press

Hardt, D. 1993. VP Ellipsis: Form, Meaning and Processing. PhD Diss. Univ. of PA

Hardt, D. 1999. Dynamic Interpretation of VP Ellipsis. Linguistics and Philosophy. 22.2. Heim, I. 2001

Jacobson, P. (to appear) Direct Compositionality and Variable-Free Semantics: The Case of Antecedent Contained Deletion", in K. Johnson (ed.), Topics in Ellipsis, Oxford University Press.

Johnson, K. 2003. In search of the Middle Field. Ms. Univ. of Mass.

Joshi A. and Y. Schabes 1997. Tree-Adjoining Grammars, in G. Rozenberg and A. Salomaa (eds.) Handbook of Formal Languages, Springer, Berlin

Kallmeyer, L. and A.K Joshi 2003. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Research on Language and Computation 1(1-2), 358.

Kallmeyer, L. and M. Romero. 2004. LTAG Semantics with Semantic Unification. In Proceedings of TAG+7, Vancouver, Canada

Sag, Ivan 1980 Deletion and Logical Form. Garland Press: New York

Shieber, S., F.Pereira, and M.Dalrymple 1996. Interactions of Scope and Ellipsis. In Linguistics and Philosophy 19, pp.527-552.