# An Empirical Approach to VP Ellipsis

**Daniel Hardt**
Department of Computing Science
Villanova University
Villanova, PA 19085-1699
*hardt@vill.edu*

**Abstract**

A system for automatically identifying and resolving VP ellipsis is described. Automatic identification of VP ellipsis is performed by matching tree patterns in a parsed corpus. The antecedent is determined through the use of a syntactic filter on possible antecedents, together with preference factors such as recency, structural prominence, priming, and parallelism. In a test performed on examples from the Penn Treebank, the system selected the correct antecedent in 83.5% of VP ellipsis occurrences.

## Introduction

Ellipsis is a pervasive phenomenon in natural language, and it has been a major topic of study in theoretical linguistics and computational linguistics in the past several decades (Ross 1967; Sag 1976; Williams 1977; Hankamer & Sag 1976; Webber 1978; Lappin 1984; Sag & Hankamer 1984; Chao 1987; Ristad 1990; Kitagawa 1990; Harper 1990; Lappin 1992; Hardt 1993; Kehler 1993; Fiengo & May 1994). However, there has been little empirically-oriented work on ellipsis. The availability of parsed corpora such as the Penn Treebank (Marcus, Santorini, & Marcinkiewicz 1993) makes it possible to empirically investigate elliptical phenomena in a way not possible before.

In this paper, an empirical approach to VP ellipsis (VPE) is described, consisting of a method for automatically identifying and resolving VPE occurrences. Automatic identification of VPE occurrences is performed by matching tree patterns in a parsed corpus; this is implemented using the tree pattern-matching utility *tgrep*.[1] The antecedent is determined through the use of a syntactic filter on possible antecedents, together with preference factors such as recency, structural position, priming, and parallelism. The VPE antecedent location system (VPEAL) is implemented in Common Lisp, and operates on Penn Treebank parse trees, as selected by *tgrep*. The system was developed using approximately 120 examples selected from a cross-section of the Penn Treebank. Then, a blind test was performed upon 103 new examples from the Treebank. The system selected the correct antecedent in 86 of these examples, for a success rate of 83.5%.

An immediate application of the system will be to annotate the Penn Treebank for VPE resolution. With minor modifications, the system can be used on other syntactically annotated corpora, or in conjunction with a parser.

In what follows, the major subparts of the system are considered in turn: first, the automatic identification of VPE is described. Next, we discuss the Syntactic Filter on potential antecedents, after which the following Preference Factors are described: Recency, Syntactic Position, Priming, and Parallelism. Finally, the test results are presented.

## Identification

VPE occurrences are automatically identified using the tree pattern-matching utility *tgrep*. VPE is defined as a Sentence (S) with an Auxiliary (AUX) but no VP. This is captured by the following tgrep pattern:

```
tgrep '/^S/<AUX \!<VP'
```

The following example is matched by this pattern:

```
(TOP (S (NP (PRP You))
   (AUX (MD can))
      (VP (VB do)
         (NP (DT this))))   (. .))
(TOP (S (CC But) (, ,)
      (SBAR (IN if)
            (S (NP (PRP you))
               (AUX (VBP do))))
```

In the Penn Treebank, auxiliary verbs are sometimes labeled as VPs. Thus, the following additional pattern is needed:

```
tgrep 'S<(VP<-(/^V/<-/^[Hh]ave$/|/^[Hh]as$/
|/^[Hh]ad$/|/^[Dd]o$/|/^[Dd]oing$/|/^[Dd]one$/
|/^[Dd]id$/|/^[Dd]oes$/|/^[Ww]ill$/|/^[Ww]ould$/|
/^[Cc]an$/|/^[Cc]ould$/|/^[Ss]hall$/
|/^[Ss]hould$/) \!$ VP)'
```

[1] *tgrep* is written by Rich Pito, University of Pennsylvania.

This pattern describes an S that immediately dominate a VP that in turn dominates a Verb that is actually an auxiliary verb: that is, a form of *have, do, will, can,* or *should.* Furthermore, it is specified that the VP does not dominate any other VP. This captures cases of VPE such as the following example:

```
(TOP (S (NP (PRP I))
        (VP (VBD made)
            (NP (PRP you))
            (NP (DT a)
                (NN man))))  ('' '')
    (. .))
(TOP (`` (S (X (RB Yes)) (, ,)
            (NP (NNP Gavin)) (, ,)
            (S (NP (PRP you))
               (VP (VBD did)))) ('' '')
```

## Antecedent Location: Overall Structure

The VPE antecedent location routine (VPEAL) has the following structure:

1. Generate Candidates
2. Syntactic Filter
3. Preference Ordering
4. Selection

The generation of candidates for VPE antecedents are all full VP's appearing within a certain window. This window is currently defined as the current sentence and the two preceding sentences. Next, impossible candidates are removed, through a syntactic filter. This filter removes all VP's that contain the VPE in an improper fashion. A preference ordering is imposed upon the remaining candidate antecedents, based on factors such as recency, structural position, priming, and parallelism. Finally, the highest rated candidate is selected.

## Syntactic Filter

The syntactic filter is based on the following constraint: the antecedent cannot contain the VPE occurrence. [2] An example of this is the following:

(1)
```
(S (NP (PRP she))
 (VP (VBD said)
     (SBAR (-NONE- 0)
           (S (NP (PRP she))
              (VPE (MD would))
              (NEG (RB not)))))))
```

Here, the VPE occurrence *would* cannot select as its antecedent the containing VP headed by *said.*

Pronoun resolution systems often incorporate a syntactic filter – a mechanism to remove certain antecedents based on syntactic structure. The basic syntactic constraint for pronouns is that they cannot take a "local" antecedent, as described by Condition B of the binding theory (Chomsky 1981)[3]. The sytactic filter for VPE also rules out "local" antecedents in a sense: it rules out antecedents in certain containment configurations.

The implementation of the syntactic filter is complicated by two factors: first, there are certain cases in which a containing antecedent is possible, where the VPE is contained in an NP argument of the containing VP, as in the following example[4]:

(2)
```
(S (NP (PRP she))
    (AUX (VBD was))
    (VP (VBG getting)
        (ADJP (RB too)
          (JJ old)
          (S (NP (-NONE- *))
             (AUX (TO to))
             (VP (VB take)
               (NP (NP (DT the)
                   (NN pleasure))
                (PP (IN from)
                    (NP (PRP it)))
                (SBAR-2 (WHNP (WDT that))
                    (S (NP (PRP she))
                       (VP (VBD used)
                           (S (NP (-NONE- *))
                              (VPE (TO to))))
```

Here, the VP headed by *take* is the antecedent for the VPE, despite the containment relation.

The second complication results from a basic limitation in Treebank parses; there is no distinction between arguments and adjuncts. A VP must be ruled out if the VPE is within a non-quantificational argument; When a VPE occurs in an adjunct positions, the "containing" VP is a permissible antecedent. The following is an example of this:

(3)
```
(S (NP (-NONE- *))
  (VP (VB get)
      (PP (TO to)
          (NP (DT the)
              (NN corner)
              (PP (IN of)
                  (NP (NP (NNP Adams))
                      (CC and)
                      (NP (NNP Clark))
      (ADVP (RB just)
            (RB as)
            (RB fast)
            (PP (IN as)
                (S (NP (PRP you))
                   (VPE (MD can)))))))
```

[2] This constraint is discussed in (Hardt 1992) as a way of ruling out antecedents for VPE.

[3] See (Brennan, Friedman, & Pollard 1987; Lappin & McCord 1990) for two approaches to pronoun resolution that incorporate a syntactic filter of this sort.

[4] See (Sag 1976; May 1985) for discussion. The analysis of these constructions is controversial; (Lappin & McCord 1990; Jacobson 1992) present alternative views.

In this case, the VP headed by *get* is that antecedent for the VPE, despite the appearance of containment. Since the VPE is contained in an adjunct (an adverbial phrase), there is in fact a non-maximal VP headed by *get* that does not contain the VPE: this is the VP *get to the corner of Adams and Clark*. However, because of the approach taken in annotating the Penn Treebank, this non-maximal VP is not displayed as a VP.

To capture the above data, the syntactic filter rules out VP's that contain the VPE in a *sentential complement*; any other antecedent-containment relation is permitted. Furthermore, a sentential complement is restricted to be an S category not immediately dominating certain other categories, such as WHNP. This correctly rules out the containing antecedent in (1), and permits it in (2) and (3).

## Preference Factors

Remaining candidates are ordered according to the following four preference factors:

1. recency
2. syntactic position
3. priming
4. similar parallel elements

Each candidate is initialized with a *weight* of 1. This weight is modified by any applicable preference factors.

### Recency

The simplest and most important factor is *recency*: If no other preference factors obtain, the (syntactically possible) antecedent closest to the ellipsis site is always chosen. The weights are modified as follows: the first VP weight is unchanged. Moving rightward, toward the VPE, the weight of each subsequent VP is incremented with a *recency increment*, whose level has been set at .35. Each subsequent VP receives one more recency increment than the previous. Thus, if there are three VP's preceding the VPE, we begin with the following weights (1 1 1). After the recency increments, we have (1 1.35 1.7). If a VP contains another VP, the two VP's receive the same number of recency increments. For example, if we have four VP's, and the second contains the third, we would have (1 1.35 1.35 1.7). Finally, VP's *following* the VPE are penalized in a symmetrical fashion; successive backwards-penalties are subtracted from VP's following the VPE.

### Syntactic Position

Another preference factor is *syntactic position*: if a VP contains another VP, the containing VP is preferred. An example of this is the following:

```
(TOP (SQ (AUX (VBP Do))
         (NP (PRP you))
         (VP (VB love)
             (S (NP (-NONE- *))
                (AUX (TO to))
                (VP (VP (VB run)
                        (PRT (RP up))
                        (NP (DT a)
                            (NN hem)))
                    (, ,)
                    (VP (VB sew)
                        (PRT (RP on))
                        (NP (NNS buttons)))
                    (, ,)
                    (VP (VB make)
                        (NP (JJ neat)
                            (NNS buttonholes)))))))
      (. ?)
      (. ?))
(TOP (S (SBAR (IN If)
              (S (NP (PRP you))
                 (VPE (VBP do))))
```

The VP headed by *love* is selected as the antecedent, rather than the contained VP's, headed by *run*, *sew*, and *make*. This preference is implemented by multiplying each contained VP by a *containment penalty value* of .8.

If the contained VP contains the VPE occurrence the situation is reversed: here the *contained* VP is preferred. Thus, if we have VP1 containing VP2, and VP2 contains VPE, the VP2 weight is *divided by* the *containment penalty value*, thus increasing its weight.

### Priming

Another preference factor is associated with "priming": that is, an antecedent that has been recently accessed is preferred over one that has not.[5] This is illustrated by the following example:

(4) A: And if I ever hear you say "Mist Laban" again, I'll scream. And don't tell me you didn't at church Sunday. I heard you.

(5) B: He really hadn't meant to, he assured her.

The first VPE accesses the VP *say "Mist Laban"*. This antecedent is also selected by the second VPE *meant to*.

### Parallelism

Finally, there is a preference for similar *parallel elements*. Parallel elements are the elements surrounding the ellipsis site, and the elements that correspond to them surrounding the antecedent.[6] In the case of

[5] A similar preference factor has been applied to pronoun resolution, as discussed, for example, in (Walker 1989; Lappin & Leass 1994).

[6] The term "parallel elements" is from (Dalrymple, Shieber, & Pereira 1991), where parallelism is emphasized in the interpretation of ellipsis. Parallelism is also important in many other treatments of ellipsis, such as (Prüst, Scha, & van den Berg 1991) and (Fiengo & May 1994).

VPE, the subject and auxiliary are parallel elements. In (Hardt 1992) a preference for VPE with coreferential subjects is suggested. This information is not available in the Penn Treebank; instead a preference for *exact match* of subjects is implemented. In addition, two features of the auxiliary are examined: tense and aspect.

There is a penalty for a VP whose aspect conflicts with the VPE; in particular, a be-form auxiliary conflicts with a do-form, [7] as in the following example:

```
(TOP (      (S (NP (PRP I))
               (AUX (VBP do))
               (NEG (RB  n't))
               (VP (VB smoke)))
            ( _cquote )
            (   )))
(TOP (S (NP (PRP She))
        (AUX (VBD was))
        (VP (VBN horrified)))
     (   ))
(TOP (      (SQ (VPE (VBP Do))
                (NP (PRP you)))
            ( _cquote )
            (  ?)
```

The VP "was horrified" is penalized because it is a be-form, and the VP "smoke" is correctly selected as the antecedent. This is implemented by multiplying the VP by an *aspect penalty* value of .5.

There is also a smaller penalty for a VP whose tense conflicts with the VPE.[8] Here, the *tense penalty* is .7.

The following (constructed) examples illustrate the preference for same tense:

(6) John said Harry will leave. Susan did too. (*say Harry will leave*)

(7) John said Harry will leave. Susan will too. (*leave*)

## Test Results

VPEAL is implemented in Common Lisp, and operates on Treebank parse trees. A blind test of the VPEAL program was performed on 103 examples from the Penn Treebank. The system selected the correct antecedent in 86 of these examples, for a success rate of 83.5%.

An alternative test was performed, in which the most recent antecedent was always selected. Here, the correct antecedent was selected in 71 of 103 examples, for a success rate of 69%. Thus, VPEAL performs significantly better than the simple recency-based approach.

Further investigation is required to assess the performance of the subparts of VPEAL.

[7] This constraint is suggested in (Hardt 1992).

[8] Observations along these lines are made in (Fiengo & May 1994), page 254-256.

## Conclusion

A system for automatically identifying and resolving VPE has been described. Automatic identification of VPE is performed by using *tgrep* to match tree patterns in the Penn Treebank. The antecedent is determined through the use of a syntactic filter together with preference factors such as recency, structural prominence, priming, and parallelism. The system selects the correct antecedent in 83.5% of VP ellipsis occurrences, based on a test on examples selected from the Penn Treebank. The system performs significantly better than a simple recency-based approach. In future work, aspects of discourse structure will be incorporated, and the system will be extended to other elliptical forms.

## References


Brennan, S. E.; Friedman, M. W.; and Pollard, C. J. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the ACL.*

Chao, W. 1987. *On Ellipsis.* Ph.D. Dissertation, University of Massachusetts-Amherst.

Chomsky, N. 1981. *Lectures on Government and Binding.* Foris.

Dalrymple, M.; Shieber, S.; and Pereira, F. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy* 14(4).

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

Hankamer, J., and Sag, I. 1976. Deep and surface anaphora. *Linguistic Inquiry* 7(3).

Hardt, D. 1992. An algorithm for vp ellipsis. In *Proceedings, 27th Annual Meeting of the ACL.*

Hardt, D. 1993. *Verb Phrase Ellipsis: Form, Meaning, and Processing.* Ph.D. Dissertation, University of Pennsylvania.

Harper, M. P. 1990. *The Representation of Noun Phrases in Logical Form.* Ph.D. Dissertation, Brown University.

Jacobson, P. 1992. Antecedent contained deletion in a variable-free semantics. In *Proceedings of the Second Conference on Semantics and Linguistic Theory.*

Kehler, A. 1993. The effect of establishing coherence in ellipsis and anaphora resolution. In *Proceedings, 28th Annual Meeting of the ACL.*

Kitagawa, Y. 1990. Deriving and copying predication. University of Rochester.

Lappin, S., and Leass, H. J. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics.*

Lappin, S., and McCord, M. 1990. Anaphora resolution in slot grammar. *Computational Linguistics* 16(4).

Lappin, S. 1984. Vp anaphora, quantifier scope, and logical form. *Linguistic Analysis* 13(4):273–315.

Lappin, S. 1992. The syntactic basis of ellipsis resolution. In *Proceedings of the Stuttgart Ellipsis Workshop.*

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19(2).

May, R. 1985. *Logical Form: Its Structure and Derivation.* Cambridge, MA: MIT Press.

Prüst, H.; Scha, R.; and van den Berg, M. 1991. A formal discourse grammar tackling verb phrase anaphora. Department of Computational Linguistics.

Ristad, E. 1990. *Computational Structure of Human Language.* Ph.D. Dissertation, Massachusetts Institute of Technology.

Ross, H. 1967. Constraints on variables in syntax. MIT Department of Linguistics and Philosophy.

Sag, I., and Hankamer, J. 1984. Towards a theory of anaphoric processing. *Linguistics and Philosophy* 7:325–345.

Sag, I. A. 1976. *Deletion and Logical Form.* Ph.D. Dissertation, Massachusetts Institute of Technology. (Published 1980 by Garland Publishing, New York).

Walker, M. 1989. Evaluating discourse processing algorithms. In *Proceedings, 27th Annual Meeting of the ACL.*

Webber, B. L. 1978. *A Formal Approach to Discourse Anaphora.* Ph.D. Dissertation, Harvard University. (Published 1979 by Garland Publishing, New York).

Williams, E. 1977. Discourse and logical form. *Linguistic Inquiry* 8(1):101–139.