In *Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL'01)*

Generation of VP Ellipsis: A Corpus-Based Approach

Daniel Hardt
Copenhagen Business School
Copenhagen, Denmark
dh@id.cbs.dk

Owen Rambow
AT&T Labs – Research
Florham Park, NJ, USA
Current: rambow@cs.columbia.edu

Abstract

We present conditions under which verb phrases are elided based on a corpus of positive and negative examples. Factor that affect verb phrase ellipsis include: the distance between antecedent and ellipsis site, the syntactic relation between antecedent and ellipsis site, and the presence or absence of adjuncts. Building on these results, we examine where in the generation architecture a trainable algorithm for VP ellipsis should be located. We show that the best performance is achieved when the trainable module is located after the realizer and has access to surface-oriented features (error rate of 7.5%).

1 Introduction

While there is a vast theoretical and computational literature on the interpretation of elliptical forms, there has been little study of the generation of ellipsis.[1] In this paper, we focus on Verb Phase Ellipsis (VPE), in which a verb phrase is elided, with an auxiliary verb left in its place. Here is an example:

(1) In 1980, 18% of federal prosecutions concluded at trial; in 1987, only 9% did.

Here, the verb phase *concluded at trial* is omitted, and the auxiliary *did* appears in its place. The basic condition on VPE is clear from the literature:[2] there must be an antecedent VP that is identical in meaning to the elided VP. Furthermore, it seems clear that the antecedent must be sufficiently close to the ellipsis site (in a sense to be made precise).

This basic condition provides a beginning of an account of the generation of VPE. However, there is more to be said, as is shown by the following examples:

(2) Ernst & Young said Eastern's plan would **miss projections** by $100 million. Goldman said Eastern would *miss the same mark* by at least $120 million.

In this example, the italicized VP could be elided, since it has a nearby antecedent (in bold) with the same meaning. Indeed the antecedents in this example is closer than in the following example in which ellipsis does occur:

(3) In particular Mr Coxon says businesses are **paying out** a smaller percentage of their profits and cash flow in the form of dividends than they have *VPE* historically.

In this paper, we identify factors which govern the decision to elide VPs. We examine a corpus of positive and negative examples; i.e., examples in which VPs were or were not elided. We find that, indeed, the distance between ellipsis site and antecedent is correlated with the decision to elide, as are the syntactic relation between antecedent

[1]We would like to thank Marilyn Walker, three reviewers for a previous submission, and three reviewers for this submission for helpful comments.

[2]The classic study is (Sag, 1976); for more recent work, see, eg, (Dalrymple et al., 1991; Kehler, 1993; Fiengo and May, 1994; Hardt, 1999).

and ellipsis site, and the presence or absence of adjuncts. Building on these results, we use machine learning techniques to examine where in the generation architecture a trainable algorithm for VP ellipsis should be located. We show that the best performance (error rate of 7.5%) is achieved when the trainable module is located after the realizer and has access to surface-oriented features.

In what follows, we first describe our corpus of negative and positive examples. Next, we describe the factors we coded for. Then we give the results of the statistical analysis of those factors, and finally we describe several algorithms for the generation of VPE which we automatically acquired from the corpus.

2 The Corpus

All our examples are taken from the Wall Street Journal corpus of the Penn Treebank (PTB). We collected both negative and positive examples from Sections 5 and 6 of the PTB. The negative examples were collected using a mixture of manual and automatic techniques. First, candidate examples were identified automatically if there were two occurrences of the same verb, separated by fewer than 10 intervening verbs. Then, the collected examples were manually examined to determine whether the two verb phrases had identical meanings or not.[3] If not, the examples were eliminated. This yielded 111 negative examples.

The positive examples were taken from the corpus collected in previous work (Hardt, 1997). This is a corpus of several hundred examples of VPE from the Treebank, based on their syntactic analysis. VPE is not annotated uniformly in the PTB. We found several different bracketing patterns and searched for these patterns, but one cannot be certain that no other bracketing patterns were used in the PTB. This yielded 15 positive examples in Sections 5 and 6. The negative and positive examples from Sections 5 and 6 – 126 in total – form our basic corpus, which we will refer to as SECTIONS5+6.

While not pathologically peripheral, VPE is a fairly rare phenomenon, and 15 positive examples is a fairly small number. We created a second corpus by extending SECTIONS5+6 with positive examples from other sections of the PTB so that the number of positive examples equals that of the negative examples. Specifically, we included all positive examples from Section 8 through 13. The result is a corpus with 111 negative examples – those from SECTIONS5+6 – and 121 positive examples (including the 15 positive examples from SECTIONS5+6). We call this corpus BALANCED; clearly BALANCED does not reflect the distribution of VPE in naturally occurring text, as does SECTIONS5+6; we therefore use it only in examining factors affecting VPE in Section 4, and we do not use it in algorithm evaluation in Section 5.

3 Factors Examined

Each example was coded for several features, each of which has figured implicitly or explicitly in the research on VPE. The following **surface-oriented features** were added automatically.

- **Sentential Distance** (sed): Measures distance between possible antecedent and candidate, in sentences. A value of 0 means that the VPs are in the same sentence.
- **Word Distance** (vpd): Measures distance between possible antecedent and candidate, in words.
- **Antecedent VP Length**(anl): Measures size of the antecedent VP, in words.

All subsequent features were coded by hand by two of the authors. The following **morphological features** were used:

- **Auxiliaries** (in1 and in2): Two features, for antecedent and candidate VP. The value is the list of full forms of the auxiliaries (and verbal particle *to*) on the antecedent and candidate verbs. This information can be annotated reliably ($\kappa_{\text{in1}} = 0.82$ and $\kappa_{\text{in2}} = 0.89$).[4]

[3] The proper characterization of the identity condition licensing VPE remains an open area of research, but it is known to permit various complications, such as "sloppy identity" and "vehicle change" (see (Fiengo and May, 1994) and references therein).

[4] Following (Carletta, 1996), we use the κ statistic to estimate reliability of annotation. We assume that values $\kappa \geq .8$ show reliability, and values $0.67 < \kappa < 0.8$ show sufficient reliability for drawing conclusions, given that the other variable we are comparing these variables to (VPE) is coded 100% correctly.

The following **syntactic features** were coded:

- **Voice** (vox): Grammatical voice (active/passive) of antecedent and candidate. This information can be annotated reliably ($\kappa = 0.83$).
- **Syntactic Structure** (syn): This feature describes the syntactic relation between the head verbs of the two VPs, i.e., *conjunction* (which includes "conjunction" by juxtaposition of root sentences), *subordination*, *comparative constructions*, and *as-appositive* (for example, *the index maintains a level below 50%, as it has for the past couple of months*). This information can be annotated reasonably reliably ($\kappa = 0.73$).
- **Subcategorization frame** for each verb. Standard distinctions between intransitive and transitive verbs with special categories for other subcategorization frames (total of six possible values). These two features can be annotated highly reliably ($\kappa = 0.93$).

We now turn to **semantic and discourse features**.

- **Adjuncts** (adj): that the arguments have the same meaning is a precondition for VPE, and it is also a precondition for us to include a negative example in the corpus. Therefore, semantic similarity of arguments need not be coded. However, we do need to code for the semantic similarity of adjuncts, as they may differ in the case of VPE: in (3) above, the second (elided) VP has the additional adverb *historically*. We distinguish the following cases: adjuncts being identical in meaning, similar in meaning (of the same semantic category, such as temporal adjuncts), only the antecedent or candidate VP having an adjunct, the adjuncts being different, there being no adjuncts at all. This information can be annotated reliably at a satisfactory level ($\kappa = 0.68$).
- **In-Quotes** (qut): Is the antecedent and/or the candidate within a quoted passage, and if yes, is it semantically the same quote. This information can be annotated highly reliably ($\kappa = 1$).
- **Discourse Structure** (dst): Are the discourse segments containing the antecedent and candidate directly related in the discourse structure? Possible values are Y and N. Here, "directly related" means that the two VPs are in the same segment, the segments are directly related to each other, or the segments are both directly related to the same third discourse segment. For this feature, inter-annotator agreement could not be achieved to a satisfactory degree ($\kappa = 0.54$), but the feature was not identified as useful during machine learning anyway. In future research, we hope to use independently coded discourse structure in order to investigate its interaction with ellipsis decisions.
- **Polarity** (pol): Does the antecedent or candidate sentence contain the negation marker *not* or one of its contractions. This information can be annotated highly reliably ($\kappa = 1$).

4 Analysis of Data

In this section, we analyze the data to find which factors correlate with the presence of absence of VPE. We use the ANOVA test (or a linear model in the case of continuous-valued independent variables) and report the probability of the F value. We follow general practice in assuming that a value of $p < .05$ means that there is significant correlation.

We present results for both of our corpora: the SECTIONS5+6 corpus consisting only of examples from Sections 5 and 6 of the Penn Tree Bank, and the BALANCED corpus, containing a balanced number of negative and positive examples. Recall that BALANCED is derived from SECTIONS5+6 by adding positive examples, but no negative examples. Therefore, when summarizing the data, we report three figures: for the negative cases (No VPE), all from SECTIONS5+6; for the positive cases in SECTIONS5+6 (SEC VPE); and for the positive cases in BALANCED (BAL VPE).

4.1 Numerical Features

The two distance measures (based on words and based on sentences) both are significantly correlated with the presence of VPE while the length

of the antecedent VP is not. The results are summarized in Figure 1.

4.2 Morphological Features

For the two auxiliaries features, we do not get significant correlation for the auxiliaries on the antecedent VP, with either corpus. The situation does not change if we distinguish only two classes, namely the presence or absence of auxiliaries

4.3 Syntactic Features

When VPE occurs, the voice of the two VPs is the same, an effect that is significant only in BALANCED ($p = .024$) but not in SECTIONS5+6 ($p = 0.27$), presumably because of the small number of data points. The counts are shown in Figure 2.

The syntactic structure also correlates with VPE, with the different forms of subordination favoring VPE, and the absence of a direct relation disfavoring VPE ($p < .00001$ for both SECTIONS5+6 and BALANCED). The frequency distributions are shown in Figure 2.

Features related to argument structure are not significantly correlated with VPE. However, whether the two argument structures are identical is a factor approaching significance: in the two cases where they differ, no VPE happens ($p = .051$). More data may make this result more robust.

4.4 Semantic and Discourse Features

If the adjuncts of the antecedent and candidate VPs (matched pairwise) are the same, then VPE is more likely to happen. If only one VP or the other has adjuncts, or if the VPs have different adjuncts, VPE is unlikely to happen. The correlation is significant for both corpora ($p < .00001$). The distribution is shown in Figure 2.

Feature In-Quotes correlates significantly with VPE in both corpora ($p = .007$ for SEC and $p = .0008$ for BAL). We see that VPE does not often occur across quotes, and that it occurs unusually frequently within quotes, suggesting that it is more common in spoken language than in written language (or, at any rate, in the WSJ).

The binary discourse structure feature correlates significantly with VPE ($p = .0039$ for SECTIONS5+6 and $p < .00001$ for BAL), with presence of a close relation correlating with VPE. Since inter-annotator agreement was not achieved at a satisfactory level, the value of this feature remains to be confirmed.

5 Algorithms for VPE

The previous section has presented a corpus-based static analysis of factors affecting VPE. In this section, we take a computational approach. We would like to use a trainable module that learns rules to decide whether or not to perform VPE. Trainable components have the advantage of easily being ported to new domains. For this reason we use the machine learning system Ripper (Cohen, 1996). However, before we can use Ripper, we must discuss the issue of how our new trainable VPE module fits into the architecture of generation.

5.1 VPE in the Generation Architecture

Tasks in the generation process have been divided into three stages (Rambow and Korelsky, 1992): the **text planner** has access only to information about communicative goals, the discourse context, and semantics, and generates a non-linguistic representation of text structure and content. The **sentence planner** chooses abstract linguistic resources (meaning-bearing lexemes, syntactic constructions) and determines sentence boundaries. It passes an abstract lexico-syntactic specification[5] to the **Realizer**, which inflects, adds function words, and linearizes, thus producing the surface string. The question arises where in this architecture the decision about VPE should be made. We will investigate this question in this section by distinguishing three places for making the VPE decision: in or just after the text planner; in or just after the sentence planner; and in or just after the realizer (i..e, at the end of the whole generation process if there are no modules after realization, such as prosody). We will refer to these three architecture options as **TP**, **SP**, and **Real**.

From the point of view of this study, the three options are distinguished by the subset of the fea-

[5]The interface between sentence planner and realizer differs among approaches and can be more or less semantic; we will assume that it is an abstract syntactic interface, with structures marked for grammatical function, but which does not represent word order.

Measure	No VPE	SEC VPE	BAL VPE	SEC Prob	BAL Prob
Word Distance	35.5	6.5	7.2	$p < .0001$	$p < .0001$
Sentential Distance	1.6	0.1	0.2	$p < .0001$	$p < .0001$
Antecedent VP length	3.6	3.9	3.3	$p = .76$	$p = .55$

Figure 1: Means and linear model analysis of correlation for numerical features

Voice Feature (vox)	No VPE	SEC VPE	BAL VPE
Both active	87	15	97
Antecedent active,candidate passive	13	0	0
Antecedent passive, candidate active	3	0	0
Both passive	8	0	4
Syntactic Feature (syn)	No VPE	SEC VPE	BAL VPE
as appositive	1	4	16
Comparative	0	6	24
Other Subordination	5	2	24
Conjunction	7	2	21
Other or no relation	98	1	15
Adjunct Feature (adj)	No VPE	SEC VPE	BAL VPE
Adjunct only on antecedent VP	10	0	0
Adjunct only on candidate VP	23	1	4
Different adjuncts	15	0	1
Neither VP has adjunct	33	7	56
VPs have same adjuncts	3	6	33
VPs have adjuncts of similar type	24	0	6
Quote Feature (qut)	No VPE	SEC VPE	BAL VPE
No quotes	91	9	75
Antecedent only in quotes	2	0	1
Candidate only in quotes	6	1	1
Both in different quotes	6	0	1
Both in same quotes	6	5	23
Binary Discourse Structure Feature (dst)	No VPE	SEC VPE	BAL VPE
Close discourse relation	70	15	96
No close discourse relation	41	0	5
Total	111	15	101

Figure 2: Counts for different features

tures as identified in Section 3 that the algorithm has access to: **TP** only has access to discourse and semantic features; **SP** can also use syntactic features, but not morphological features or those that relate to surface ordering. **Real** can access all features. We summarize the relation between architecture option and features in Figure 3.

5.2 Using a Machine Learning Algorithm

We use Ripper to automatically learn rule sets from the data. Ripper is a rule learning program, which unlike some other machine learning programs supports bag-valued features.[6] Using a set of attributes, Ripper greedily learns rule sets that choose one of several classes for each data set. We use two classes, vpe and novpe. By using different parameter settings for Ripper, we obtain different rule sets. These parameter settings are of two types: first, parameters internal to Ripper, such as the number of optimization passes; and second, the specification of which attributes are used. To determine the optimal number of optimization passes, we randomly divided our SECTIONS5+6 corpus into a training and test part, with the test corpus representing 20% of the data. We then ran Ripper with different settings for the optimization pass parameter. We determined that best results are obtained with six passes. We then used this setting in all subsequent work with Ripper. The test/training partition used to determine this setting was not used for any other purpose.

In the next subsection (Section 5.3), we present and discuss several rule sets, as they bring out different properties of ellipsis. We discuss rule sets trained on and evaluated against the entire set of data from SECTIONS5+6: since our data set is relatively small, we decided not to divide it into distinct training and test sets (except for determining the internal parameter; see above). The fact that these rule sets are obtained by a machine learning algorithm is in some sense incidental here, and while we give the coverage figures for the training corpus, we consider them of mainly qualitative interest. We present three rule sets, one each for each of three architecture options, each one with its own set of attributes. We start out with a full set of attributes, and successively eliminate the more surface-oriented and syntactic ones. As we will see, the earlier the VPE decision is made, the less reliable it is.

In the subsection after next (Section 5.4), we present results using ten-fold cross-validation, for which the quantitative results are meaningful. However, since each run produces ten different rule sets, the qualitative results, in some sense, are not meaningful. We therefore do not give any rule sets; the cross-validation demonstrates that effective rule sets can be learned even from relatively small data sets.

5.3 Algorithms for VP Ellipsis Generation

We will present three different rule sets for the three architecture options. All rule sets must be used in conjunction with a basic screening algorithm, which is the same one that we used in order to identify negative examples: there must be two identical verbs with at most ten intervening verbs, and the arguments of the verbs must have the same meaning. Then the following rule sets can be applied to determine whether a VPE should be generated or not.

We start out with the **Real** set of features, which is available after realization has completed, and thus all surface-oriented and morphological features are available. Of course, we also assume that all other features are still available at that time, not *just* the surface features. We obtain the following rule set:

```
Choose VPE if sed<=0 and syn=com (6/0).
Choose VPE if vpd<=14, sed<=0,
                        and anl>=3 (7/1).
Otherwise default to no VPE (110/2).
```

Each rule (except the first) only applies if the preceding ones do not. The first rule says that if the distance in sentences between the antecedent VP and candidate VP (sed) is less than or equal to 0, i.e., the candidate and the antecedent are in the same sentence, and the syntactic construction is a comparative, then choose VPE. This rule accounts for 6 cases correctly and misclassified none. The second rule says that if the distance in words between antecedent VP and candidate VP is less than or equal to 14, and the VPs are in the same sentence, and the antecedent VP contains 3 or more words, then the candidate VP is elided. This rule accounts for 7 cases correctly but misclassified one. Finally, all other cases are

[6] Our only bag-valued set of features is the set of auxiliaries, which is not used in the rules we present here.

Short Name	VPE Module After	Features Used
TP	Text planner	quotes, polarity, adjuncts, discourse structure
SP	Sentence planner	all from TP plus voice, syntactic relation, subcat, size of antecedent VP, and distance in sentences
Real	Realizer	all from SP plus auxiliaries and distance in words

Figure 3: Architecture options and features

not treated as VPE, which misses 2 examples but classifies 110 correctly. This yields an overall training error rate of 2.4% (3 misclassified examples). (Recall that we are here comparing the performance against the training set.)

We now consider the examples from the introduction, which are repeated here for convenience.

(4) In 1980, 18% of federal prosecutions concluded at trial; in 1987, only 9% did.

(5) Ernst & Young said Eastern's plan would **miss projections** by $100 million. Goldman said Eastern would *miss the same mark* by at least $120 million.

(6) In particular Mr Coxon says businesses are **paying out** a smaller percentage of their profits and cash flow in the form of dividends than they have *VPE* historically.

Consider example (4). The first rule does not apply (this is not a comparative), but the second does, since both VPs are in the same sentence, and the antecedent has three words, and the distance between them is fewer than 14 words. Thus (4) would be generated as a VPE. The first rule does apply to example (6), so it would also be generated as a VPE. Example (5), however, is not caught by either of the first two rules, so it would not yield a VPE. We thus replicate the data in the corpus for these three examples.

We now turn to **SP**. We assume that we are making the VPE decision before realization, and therefore have access only to syntactic and semantic features, but not to surface features. As a result, distance in words is no longer available as a feature.

```
Choose VPE if sed<=0 and anl>=3 (10/3).
Choose VPE if sed<=0 and adj=sam (3/0).
Otherwise default to no VPE (108/2).
```

Here, we first choose VPE if the antecedent and candidate are in the same sentence and the antecedent VP length is greater than three, or if the two VPs are in the same sentence and they have the same adjuncts. In all other cases, we choose not to elide. The training error rate goes up to 3.97%.

With this rule set, we can correctly predict a VPE for examples (4) and (6), using the first rule. We do not generate a VPE for (5), since it does not match either of the two first rules.

Finally, we consider architecture option **TP**, in which the VPE decision is made right after text planning, and only semantic and discourse features are available. The rule set is simplified:

```
Choose VPE if adj=sam (6/3).
Otherwise default to no VPE (108/9).
```

VPE is only chosen if the adjuncts are the same; in all other cases, VPE is avoided. The training error rate climbs to 9.52%.

For our examples, only example (4) generates a VPE since the adjuncts are the same on the two VPS[7] (6) fails to meet the requirements of the first rule since the second VP has an adjunct of its own, *historically*.

5.4 Quantitative Analysis

In the previous subsection we presented different rule sets. We now show that rule sets can be derived in a consistent manner and tested on a held-out test set with satisfactory results. We take these results to be indicative of performance on unseen data (which is in the WSJ domain and genre, of course). We use ten-fold cross-validation for this purpose, with the same three sets of possible attributes used above.

The results for the three attribute sets are shown in Figure 4 (average error rates for the tenfold

[7]The adjunct is elided on the second VP, of course, but present in the input representation, not shown here.

Architecture Option	Mean Error Rate	Error Reduction
TP	11.7%	0%
SP	9.2%	23%
Real	7.5%	35%
Baseline	11.9%	—-

Figure 4: Results for 10-fold cross validation for different architectures: after realizer, after sentence planner, after text planner

cross-validations). The baseline is obtained by never choosing VPE (which, recall, is relatively rare in the SECTIONS5+6 corpus). We see that the **TP** architecture does not do better than the baseline, while **SP** results in an error reduction of 23% and the **Real** architecture in an error reduction of 35%, for an average error rate of 7.5%.

6 Conclusion

We have found that the decision to elide VPs is statistically correlated with several factors, including distance between antecedent and candidate VPs by word or sentence, and the presence or absence of syntactic and discourse relations. These findings provide a strong foundation on which to build algorithms for the generation of VPE. We have explored several possible algorithms with the help of a machine learning system, and we have found that these automatically derived algorithms perform well on cross-validation tests.

We have also seen that the decision whether or not to elide can be better made later in the generation process: the more features are available, the better. It is perhaps not surprising that the decision cannot be made very well just after after text planning: it is well known that VPE is subject to syntactic constraints, and the relevant information is not yet available. It is perhaps more surprising that the surface-oriented features appear to contribute to the quality of the decision, pushing the decision past the realization phase. One possible explanation is that there are in fact other features, which we have not yet identified, and for which the surface-oriented features are stand-ins. If this is the case, further work will allow us to define algorithms so that the decision on VPE can be made after sentence planning. However, it is also possible that decisions about VPE (and related pronominal constraints) cannot be made before the text is linearized, presumably because of the processing limitations of the hearer/reader (and of the speaker/writer). Walker (1996) has argued in favor of the importance of limited attention in processing discourse phenomena, and the surface-oriented features can be argued to model such cognitive constraints.

References

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*. AAAI.

Mary Dalrymple, Stuart Shieber, and Fernando Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4), August.

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

Daniel Hardt. 1997. An empirical approach to vp ellipsis. *Computational Linguistics*, 23(4):525–541.

Daniel Hardt. 1999. Dynamic interpretation of verb phrase ellipsis. *Linguistics and Philosophy*, 22(2):187–221.

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

Owen Rambow and Tanya Korelsky. 1992. Applied text generation. In *Third Conference on Applied Natural Language Processing*, pages 40–47, Trento, Italy.

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

Marilyn A. Walker. 1996. Limited attention and discourse structure. *Computational Linguistics*, 22-2:255–264.