Robust VPE detection using Automatically Parsed Text

Leif Arda Nielsen
Department of Computer Science
King's College London
nielsen@dcs.kcl.ac.uk

Abstract

This paper describes a Verb Phrase Ellipsis (VPE) detection system, built for robustness, accuracy and domain independence. The system is corpus-based, and uses machine learning techniques on free text that has been automatically parsed. Tested on a mixed corpus comprising a range of genres, the system achieves a 70% F1-score. This system is designed as the first stage of a complete VPE resolution system that is input free text, detects VPEs, and proceeds to find the antecedents and resolve them.

1 Introduction

Ellipsis is a linguistic phenomenon that has received considerable attention, mostly focusing on its interpretation. Most work on ellipsis (Fiengo and May, 1994; Lappin, 1993; Dalrymple et al., 1991; Kehler, 1993; Shieber et al., 1996) is aimed at discerning the procedures and the level of language processing at which ellipsis resolution takes place, or ambiguous and difficult cases. The detection of elliptical sentences or the identification of the antecedent and elided clauses within them are usually not dealt with, but taken as given. Noisy or missing input, which is unavoidable in NLP applications, is not dealt with, and neither is focusing on specific domains or applications. It therefore becomes clear that a robust, trainable approach is needed.

An example of Verb Phrase Ellipsis (VPE), which is detected by the presence of an auxiliary verb without a verb phrase, is seen in example 1. VPE can also occur with semi-auxiliaries, as in example 2.

(1) John$_3$ $\{$loves his$_3$ wife$\}_2$. Bill$_3$ does$_1$ too.

(2) But although he was terse, he didn't $\{$rage at me$\}_2$ the way I expected him to$_1$.

Several steps of work need to be done for ellipsis resolution :

1. Detecting ellipsis occurrences. First, elided verbs need to be found.
2. Identifying antecedents. For most cases of ellipsis, copying of the antecedent clause is enough for resolution (Hardt, 1997).
3. Resolving ambiguities. For cases where ambiguity exists, a method for generating the full list of possible solutions, and suggesting the most likely one is needed.

This paper describes the work done on the first stage, the detection of elliptical verbs. First, previous work done on tagged corpora will be summarised. Then, new work on parsed corpora will be presented, showing the gains possible through sentence-level features. Finally, experiments using unannotated data that is parsed using an automatic parser are presented, as our aim is to produce a stand-alone system.

We have chosen to concentrate on VP ellipsis due to the fact that it is far more common than

other forms of ellipsis, but pseudo-gapping, an example of which is seen in example 3, has also been included due to the similarity of its resolution to VPE (Lappin, 1996). *Do so/it/that* and *so doing* anaphora are not handled, as their resolution is different from that of VPE (Kehler and Ward, 1999).

(3) John writes plays, and Bill does novels.

2 Previous work

Hardt's (1997) algorithm for detecting VPE in the Penn Treebank (see Section 3) achieves precision levels of 44% and recall of 53%, giving an F1[1] of 48%, using a simple search technique, which relies on the parse annotation having identified empty expressions correctly.

In previous work (Nielsen, 2003a; Nielsen, 2003b) we performed experiments on the British National Corpus using a variety of machine learning techniques. These earlier results are not directly comparable to Hardt's, due to the different corpora used. The expanded set of results are summarised in Table 1, for Transformation Based Learning (TBL) (Brill, 1995), GIS based Maximum Entropy Modelling (GIS-MaxEnt) (Ratnaparkhi, 1998), L-BFGS based Maximum Entropy Modelling (L-BFGS-MaxEnt)[2] (Malouf, 2002), Decision Tree Learning (Quinlan, 1993) and Memory Based Learning (MBL) (Daelemans et al., 2002).

Algorithm	Recall	Precision	F1
TBL	69.63	85.14	76.61
Decision Tree	60.93	79.39	68.94
MBL	72.58	71.50	72.04
GIS-MaxEnt	71.72	63.89	67.58
L-BFGS-MaxEnt	71.93	80.58	76.01

Table 1: Comparison of algorithms

[1]Precision, recall and F1 are defined as :

$$Recall = \frac{No(\text{correct ellipses found})}{No(\text{all ellipses in test})} \quad (1)$$

$$Precision = \frac{No(\text{correct ellipses found})}{No(\text{all ellipses found})} \quad (2)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

[2]Downloadable from http://www.nlplab.cn/zhangle/maxent_toolkit.html

For all of these experiments, the training features consisted of lexical forms and Part of Speech (POS) tags of the words in a three word forward/backward window of the auxiliary being tested. This context size was determined empirically to give optimum results, and will be used throughout this paper. The L-BFGS-MaxEnt uses Gaussian Prior smoothing which was optimized for the BNC data, while the GIS-MaxEnt has a simple smoothing option available, but this deteriorates results and is not used. MBL was used with its default settings.

While TBL gave the best results, the software we used (Lager, 1999) ran into memory problems and proved problematic with larger datasets. Decision trees, on the other hand, tend to oversimplify due to the very sparse nature of ellipsis, and produce a single rule that classifies everything as non-VPE. This leaves Maximum Entropy and MBL for further experiments.

3 Corpus description

The British National Corpus (BNC) (Leech, 1992) is annotated with POS tags, using the CLAWS-4 tagset. A range of V sections of the BNC, containing around 370k words[3] with 645 samples of VPE was used as training data. The separate test data consists of around 74k words[4] with 200 samples of VPE.

The Penn Treebank (Marcus et al., 1994) has more than a hundred phrase labels, and a number of empty categories, but uses a coarser tagset. A mixture of sections from the Wall Street Journal and Brown corpus were used. The training section[5] consists of around 540k words and contains 522 samples of VPE. The test section[6] consists of around 140k words and contains 150 samples of VPE.

4 Experiments using the Penn Treebank

To experiment with what gains are possible through the use of more complex data such as

[3]Sections CS6, A2U, J25, FU6, H7F, HA3, A19, A0P, G1A, EWC, FNS, C8T

[4]Sections EDJ, FR3

[5]Sections WSJ 00, 01, 03, 04, 15, Brown CF, CG, CL, CM, CN, CP

[6]Sections WSJ 02, 10, Brown CK, CR

parse trees, the Penn Treebank is used for the second round of experiments. The results are presented as new features are added in a cumulative fashion, so each experiment also contains the data contained in those before it.

Words and POS tags

The Treebank, besides POS tags and category headers associated with the nodes of the parse tree, includes empty category information. For the initial experiments, the empty category information is ignored, and the words and POS tags are extracted from the trees. The results in Table 2 are seen to be considerably poorer than those for BNC, despite the comparable data sizes. This can be accounted for by the coarser tagset employed.

Algorithm	Recall	Precision	F1
MBL	47.71	60.33	53.28
GIS-MaxEnt	34.64	79.10	48.18
L-BFGS-MaxEnt	60.13	76.66	67.39

Table 2: Initial results with the Treebank

Close to punctuation

A very simple feature, that checks for auxiliaries close to punctuation marks was tested. Table 3 shows the performance of the feature itself, characterised by very low precision, and results obtained by using it. It gives a 2% increase in F1 for MBL, 3% for GIS-MaxEnt, but a 1.5% decrease for L-BFGS-MaxEnt.

This brings up the point that the individual success rate of the features will not be in direct correlation with gains in overall results. Their contribution will be high if they have high precision for the cases they are meant to address, and if they produce a different set of results from those already handled well, complementing the existing features. Overlap between features can be useful to have greater confidence when they agree, but low precision in the feature can increase false positives as well, decreasing performance. Also, the small size of the test set can contribute to fluctuations in results.

Heuristic Baseline

A simple heuristic approach was developed to form a baseline. The method takes all auxiliaries as possible candidates and then eliminates them using local syntactic information in a very simple way. It searches forwards within a short range of words, and if it encounters any other verbs, adjectives, nouns, prepositions, pronouns or numbers, classifies the auxiliary as not elliptical. It also does a short backwards search for verbs. The forward search looks 7 words ahead and the backwards search 3. Both skip 'asides', which are taken to be snippets between commas without verbs in them, such as : "... papers do, however, show ...". This feature gives a 4.5% improvement for MBL (Table 4), 4% for GIS-MaxEnt and 3.5% for L-BFGS-MaxEnt.

Algorithm	Recall	Precision	F1
close-to-punctuation	30.06	2.31	4.30
MBL	50.32	61.60	55.39
GIS-MaxEnt	37.90	79.45	51.32
L-BFGS-MaxEnt	57.51	76.52	65.67

Table 3: Effects of using the close-to-punctuation feature

```
(SINV
   (ADVP-PRD-TPC-2   (RB so) )
   (VP  (VBZ is)
       (ADVP-PRD (-NONE- *T*-2)  ))
   (NP-SBJ (PRP$ its)
       (NN balance) (NN sheet) ))
```

Figure 1: Fragment of sentence from Treebank

Algorithm	Recall	Precision	F1
heuristic	48.36	27.61	35.15
MBL	55.55	65.38	60.07
GIS-MaxEnt	43.13	78.57	55.69
L-BFGS-MaxEnt	62.09	77.86	69.09

Table 4: Effects of using the heuristic feature

Surrounding categories

The next feature added is the categories of the previous branch of the tree, and the next branch. So in the example in Figure 1, the previous category of the elliptical verb is ADVP-PRD-TPC-2, and the next category NP-SBJ. The results of using this feature are seen in Table 5, giving a 3.5% boost to MBL, 2% to GIS-MaxEnt, and 1.6% to L-BFGS-MaxEnt.

Algorithm	Recall	Precision	F1
MBL	58.82	69.23	63.60
GIS-MaxEnt	45.09	81.17	57.98
L-BFGS-MaxEnt	64.70	77.95	70.71

Table 5: Effects of using the surrounding categories

Auxiliary-final VP

For auxiliary verbs parsed as verb phrases (VP), this feature checks if the final element in the VP is an auxiliary or negation. If so, no main verb can be present, as a main verb cannot be followed by an auxiliary or negation. This feature was used by Hardt (1993) and gives a 3.5% boost to performance for MBL, 6% for GIS-MaxEnt, and 3.4% for L-BFGS-MaxEnt (Table 6).

Algorithm	Recall	Precision	F1
Auxiliary-final VP	72.54	35.23	47.43
MBL	63.39	71.32	67.12
GIS-MaxEnt	54.90	77.06	64.12
L-BFGS-MaxEnt	71.89	76.38	74.07

Table 6: Effects of using the Auxiliary-final VP feature

Empty VP

Hardt (1997) uses a simple pattern check to search for empty VP's identified by the Treebank, (VP (-NONE- *?*)), which achieves 60% F1 on our test set. Our findings are in line with Hardt's, who reports 48% F1, with the difference being due to the different sections of the Treebank used.

It was observed that this search may be too restrictive to catch some examples of VPE in the corpus, and pseudo-gapping. Modifying the search pattern to be '(VP (-NONE- *?*)' instead improves the feature itself by 10% in F1 and gives the results seen in Table 7, increasing MBL's F1 by 10%, GIS-MaxEnt by 14% and L-BFGS-MaxEnt by 11.7%.

Algorithm	Recall	Precision	F1
Empty VP	54.90	97.67	70.29
MBL	77.12	77.63	77.37
GIS-MaxEnt	69.93	88.42	78.10
L-BFGS-MaxEnt	83.00	88.81	85.81

Table 7: Effects of using the improved Empty VP feature

Empty categories

Finally, including empty category information completely, such that empty categories are treated as words and included in the context. Table 8 shows that adding this information results in a 4% increase in F1 for MBL, 4.9% for GIS-MaxEnt, and 2.5% for L-BFGS-MaxEnt.

Algorithm	Recall	Precision	F1
MBL	83.00	79.87	81.41
GIS-MaxEnt	76.47	90.69	82.97
L-BFGS-MaxEnt	86.27	90.41	88.29

Table 8: Effects of using the empty categories

5 Experiments with Automatically Parsed data

The next set of experiments use the BNC and Treebank, but strip POS and parse information, and parse them automatically using two different parsers. This enables us to test what kind of performance is possible for real-world applications.

5.1 Parsers used

Charniak's parser (2000) is a combination probabilistic context free grammar and maximum entropy parser. It is trained on the Penn Treebank, and achieves a 90.1% recall and precision average for sentences of 40 words or less.

Robust Accurate Statistical Parsing (RASP) (Briscoe and Carroll, 2002) uses a combination of statistical techniques and a hand-crafted grammar. RASP is trained on a range of corpora, and uses a more complex tagging system (CLAWS-2), like that of the BNC. This parser, on our data, generated full parses for 70% of the sentences, partial parses for 28%, while 2% were not parsed, returning POS tags only.

5.2 Reparsing the Treebank

The results of experiments using the two parsers (Table 9) show generally similar performance. Compared to results on the original treebank with similar data (Table 6), the results are 4-6% lower, or in the case of GIS-MaxEnt, 4% lower or 2% higher, depending on parser. This drop in performance is not surprising, given the errors introduced by the parsing process. As the parsers

do not generate empty-category information, their overall results are 14-20% lower, compared to those in Table 8.

The success rate for the features used (Table 10) stay the same, except for auxiliary-final VP, which is determined by parse structure, is only half as successful for RASP. Conversely, the heuristic baseline is more successful for RASP, as it relies on POS tags, which is to be expected as RASP has a more detailed tagset.

	Feature	Rec	Prec	F1
Charniak	close-to-punct	34.00	2.47	4.61
	heuristic baseline	45.33	25.27	32.45
	auxiliary-final VP	51.33	36.66	42.77
RASP	close-to-punct	71.05	2.67	5.16
	heuristic baseline	74.34	28.25	40.94
	auxiliary-final VP	22.36	25.18	23.69

Table 10: Performance of features on re-parsed Treebank data

5.3 Parsing the BNC

Experiments using parsed versions of the BNC corpora (Table 11) show similar results to the original results (Table 1) - except L-BFGS-MaxEnt which scores 4-8% lower - meaning that the added information from the features mitigates the errors introduced in parsing. The performance of the features (Table 12) remain similar to those for the re-parsed treebank experiments.

	Feature	Rec	Prec	F1
Charniak	close-to-punct	48.00	5.52	9.90
	heuristic baseline	44.00	34.50	38.68
	auxiliary-final VP	53.00	42.91	47.42
RASP	close-to-punct	55.32	4.06	7.57
	heuristic baseline	84.77	35.15	49.70
	auxiliary-final VP	16.24	28.57	20.71

Table 12: Performance of features on parsed BNC data

5.4 Combining BNC and Treebank data

Combining the re-parsed BNC and Treebank data diversifies and increases the size of the test data, making conclusions drawn empirically more reliable, and the wider range of training data makes it more robust. This gives a training set of 1167 VPE's and a test set of 350 VPE's. The results in Table 13 show little change from the previous experiments.

6 Conclusion and Future work

This paper has presented a robust system for VPE detection. The data is automatically tagged and parsed, syntactic features are extracted and machine learning is used to classify instances. Three different machine learning algorithms, Memory Based Learning, GIS-based and L-BFGS-based maximum entropy modeling are used. They give similar results, with L-BFGS-MaxEnt generally giving the highest performance. Two parsers were used, Charniak's and RASP, achieving similar results.

To summarise the findings :

- Using the BNC, which is tagged with a complex tagging scheme but has no parse data, it is possible to get 76% F1 using lexical forms and POS data alone
- Using the Treebank, the coarser tagging scheme reduces performance to 67%. Adding extra features, including sentence-level ones, raises this to 74%. Adding empty category information gives 88%, compared to previous results of 48% (Hardt, 1997)
- Re-parsing the Treebank data , top performance is 63%, raised to 68% using extra features
- Parsing the BNC, top performance is 71%, raised to 72% using extra features
- Combining the parsed data, top performance is 67%, raised to 71% using extra features

The results demonstrate that the method can be applied to practical tasks using free text. Next, we will experiment with an algorithm (Johnson, 2002) that can insert empty-category information into data from Charniak's parser, allowing replication of features that need this. Cross-validation experiments will be performed to negate the effects the small test set may cause.

As machine learning is used to combine various features, this method can be extended to other forms of ellipsis, and other languages. However, a number of the features used are specific to English VPE, and would have to be adapted to such cases. It is difficult to extrapolate how successful

		MBL			GIS-MaxEnt			L-BFGS-MaxEnt		
		Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1
Charniak	Words + POS	54.00	62.30	57.85	38.66	79.45	52.01	56.66	71.42	63.19
	+ features	58.00	65.41	61.48	50.66	73.78	60.07	65.33	72.05	68.53
RASP	Words + POS	55.92	66.92	60.93	43.42	56.89	49.25	51.63	79.00	62.45
	+ features	57.23	71.31	63.50	61.84	72.30	66.66	62.74	73.84	67.84

Table 9: Results on re-parsed data from the Treebank

		MBL			GIS-MaxEnt			L-BFGS-MaxEnt		
		Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1
Charniak	Words + POS	66.50	63.63	65.03	55.00	75.86	63.76	71.00	70.64	70.82
	+ features	67.50	67.16	67.33	65.00	75.58	69.89	71.00	73.19	72.08
RASP	Words + POS	61.92	63.21	62.56	64.46	54.04	58.79	65.34	70.96	68.04
	+ features	71.06	73.29	72.16	73.09	61.01	66.51	70.29	67.29	68.76

Table 11: Results on parsed data from the BNC

		MBL			GIS-MaxEnt			L-BFGS-MaxEnt		
		Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1
Charniak	Words + POS	62.28	69.20	65.56	54.28	77.86	63.97	65.14	69.30	67.15
	+ features	65.71	71.87	68.65	63.71	72.40	67.78	70.85	69.85	70.35
RASP	Words + POS	63.61	67.47	65.48	59.31	55.94	57.37	57.46	71.83	63.84
	+ features	68.48	69.88	69.17	67.61	71.47	69.48	70.14	72.17	71.14

Table 13: Results on parsed data using the combined dataset

such approaches would be based on current work, but it can be expected that they would be feasible, albeit with lower performance.

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

E. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Meeting of the North American Chapter of the ACL*, page 132.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2002. Tilburg memory based learner, version 4.3, reference guide. Downloadable from http://ilk.kub.nl/downloads/pub/papers/ilk0210.ps.gz.

Mary Dalrymple, Stuart M. Shieber, and Fernando Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14:399–452.

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

Daniel Hardt. 1993. *VP Ellipsis: Form, Meaning, and Processing*. Ph.D. thesis, University of Pennsylvania.

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

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Andrew Kehler and Gregory Ward. 1999. On the semantics and pragmatics of 'identifier so'. In Ken Turner, editor, *The Semantics/Pragmatics Interface from Different Points of View (Current Research in the Semantics/Pragmatics Interface Series, Volume I)*. Amsterdam: Elsevier.

Andrew Kehler. 1993. A discourse copying algorithm for ellipsis and anaphora resolution. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL-93)*, Utrecht, the Netherlands.

Torbjorn Lager. 1999. The mu-tbl system: Logic programming tools for transformation-based learning. In *Third International Workshop on Computational Natural Language Learning (CoNLL'99)*. Downloadable from http://www.ling.gu.se/ lager/mutbl.html.

Shalom Lappin. 1993. The syntactic basis of ellipsis resolution. In S. Berman and A. Hestvik, editors, *Proceedings of the Stuttgart Ellipsis Workshop, Arbeitspapiere des Sonderforschungsbereichs 340, Bericht Nr. 29-1992*. University of Stuttgart, Stuttgart.

Shalom Lappin. 1996. The interpretation of ellipsis. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 145–175. Oxford: Blackwell.

G. Leech. 1992. 100 million words of english : The British National Corpus. *Language Research*, 28(1):1–13.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, M. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Human Language Technology Workshop*. Morgan Kaufmann, San Francisco.

Leif Arda Nielsen. 2003a. A corpus-based study of verb phrase ellipsis. In *Proceedings of the 6th Annual CLUK Research Colloquium*.

Leif Arda Nielsen. 2003b. Using machine learning techniques for VPE detection. In *Proceedings of RANLP*.

R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Stuart Shieber, Fernando Pereira, and Mary Dalrymple. 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19(5):527–552.