Exploring the steps of Verb Phrase Ellipsis

Zhengzhong Liu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
liu@cs.cmu.edu

Edgar Gonzaĺez and **Dan Gillick**
Google Research
1600 Amphitheatre Parkway
Mountain View, CA 94043, USA
{edgargip, dgillick}@google.com

Abstract

Verb Phrase Ellipsis is a well-studied topic in theoretical linguistics but has received little attention as a computational problem. Here we propose a decomposition of the overall resolution problem into three tasks—target detection, antecedent head resolution, and antecedent boundary detection—and implement a number of computational approaches for each one. We also explore the relationships among these tasks by attempting joint learning over different combinations. Our new decomposition of the problem yields significantly improved performance on publicly available datasets, including a newly contributed one.

1 Introduction

Verb Phrase Ellipsis (VPE) is the anaphoric process where a verbal constituent is partially or totally unexpressed, but can be resolved through an antecedent in the context, as in the following examples:

(1) His wife also [antecedent *works for the paper*], as **did** his father.

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

In example 1, a light verb **did** is used to represent the verb phrase *works for the paper*; example 2 shows a much longer antecedent phrase, which in addition differs in tense from the elided one. Following Dalrymple et al. (1991), we refer to the full verb expression as the "antecedent", and to the anaphor as the "target".

VPE resolution is necessary for deeper Natural Language Understanding, and can be beneficial for instance in dialogue systems or Information Extraction applications.

Computationally, VPE resolution can be modeled as a pipeline process: first detect the VPE targets, then identify their antecedents. Prior work on this topic (Hardt, 1992; Nielsen, 2005) has used this pipeline approach but without analysis of the interaction of the different steps.

In this paper, we analyze the steps needed to resolve VPE. We preserve the target identification task, but propose a decomposition of the antecedent selection step in two subtasks. We use learning-based models to address each task separately, and also explore the combination of contiguous steps. Although the features used in our system are relatively simple, our models yield state-of-the-art results on the overall task. We also observe a small performance improvement from our decomposition modeling of the tasks.

There are only a few small datasets that include manual VPE annotations. While Bos and Spenader (2011) provide publicly available VPE annotations for Wall Street Journal (WSJ) news documents, the annotations created by Nielsen (2005) include a more diverse set of genres (e.g., articles and plays) from the British National Corpus (BNC).

We semi-automatically transform these latter annotations into the same format used by the former. The unified format allows better benchmarking and will facilitate more meaningful comparisons in the future. We evaluate our methods on both datasets, making our results directly comparable to those published by

Nielsen (2005).

2 Related Work

Considerable work has been done on VPE in the field of theoretical linguistics: e.g., (Dalrymple et al., 1991; Shieber et al., 1996); yet there is much less work on computational approaches to resolving VPE.

Hardt (1992; 1997) presents, to our knowledge, the first computational approach to VPE. His system applies a set of linguistically motivated rules to select an antecedent given an elliptical target. Hardt (1998) uses Transformation-Based Learning to replace the manually developed rules. However, in Hardt's work, the targets are selected from the corpus by searching for "empty verb phrases" (constructions with an auxiliary verb only) in the gold standard parse trees.

Nielsen (2005) presents the first end-to-end system that resolves VPE from raw text input. He describes several heuristic and learning-based approaches for target detection and antecedent identification. He also discusses a post-processing substitution step in which the target is replaced by a transformed version of the antecedent (to match the context). We do not address this task here because other VPE datasets do not contain relevant substitution annotations. Similar techniques are also described in Nielsen (2004b; 2004a; 2003a; 2003b).

Results from this prior work are relatively difficult to reproduce because the annotations on which they rely are inaccessible. The annotations used by Hardt (1997) have not been made available, and those used by Nielsen (2005) are not easily reusable since they rely on some particular tokenization and parser. Bos and Spenader (2011) address this problem by annotating a new corpus of VPE on top of the WSJ section of the Penn Treebank, and propose it as a standard evaluation benchmark for the task. Still it is desirable to use Nielsen's annotations on the BNC which contain more diverse text genres with more frequent VPE.

3 Approaches

We focus on the problems of target detection and antecedent identification as proposed by Nielsen (2005). We propose a refinement of these two tasks, splitting them into these three:

1. **Target Detection (T)**, where the subset of VPE targets is identified.
2. **Antecedent Head Resolution (H)**, where each target is linked to the head of its antecedent.
3. **Antecedent Boundary Determination (B)**, where the exact boundaries of the antecedent are determined from its head.

The following sections describe each of the steps in detail.

3.1 Target Detection

Since the VPE target is annotated as a single word in the corpus[1], we model their detection as a binary classification problem. We only consider modal or light verbs (*be*, *do*, *have*) as candidates, and train a logistic regression classifier ($\mathbf{Log}^T$) with the following set of binary features:

1. The POS tag, lemma, and dependency label of the verb, its dependency parent, and the immediately preceding and succeeding words.
2. The POS tags, lemmas and dependency labels of the words in the dependency subtree of the verb, in the 3-word window, and in the same-size window after (as bags of words).
3. Whether the subject of the verb appears to its right (i.e., there is subject-verb inversion).

3.2 Antecedent Head Resolution

For each detected target, we consider as potential antecedent heads all verbs (including modals and auxiliaries) in the three immediately preceding sentences of the target word[2] as well as the sentence including the target word (up to the target[3]). This follows Hardt (1992) and Nielsen (2005).

We perform experiments using a logistic regression classifier ($\mathbf{Log}^H$), trained to distinguish correct antecedents from all other possible candidates. The set of features are shared with the Antecedent Boundary Determination task, and are described in detail in Section 3.3.1.

[1] All targets in the corpus of Bos and Spenader (2011) are single-word by their annotation guideline.

[2] Only 1 of the targets in the corpus of Bos and Spenader (2011), has an antecedent beyond that window.

[3] Only 1% of the targets in the corpus are cataphoric.

However, a more natural view of the resolution task is that of a ranking problem. The gold annotation can be seen as a partial ordering of the candidates, where, for a given target, the correct antecedent ranks above all other candidates, but there is no ordering among the remaining candidates. To handle this specific setting, we adopt a ranking model with domination loss (Dekel et al., 2003).

Formally, for each potential target t in the determined set of targets T, we consider its set of candidates C_t, and denote whether a candidate $c \in C_t$ is the antecedent for t using a binary variable a_{ct}. We express the ranking problem as a bipartite graph $\mathcal{G} = (V^+, V^-, E)$ where vertices represent antecedent candidates:

$$\begin{aligned} V^+ &= \{(t, c) \mid t \in T, c \in C_t, a_{ct} = 1\} \\ V^- &= \{(t, c) \mid t \in T, c \in C_t, a_{ct} = 0\} \end{aligned}$$

and the edges link the correct antecedents to the rest of the candidates for the same target[4]:

$$E = \{((t, c^+), (t, c^-)) \mid (t, c^+) \in V^+, (t, c^-) \in V^-\}$$

We associate each vertex i with a feature vector $\mathbf{x}_i$, and compute its score s_i as a parametric function of the features $s_i = g(\mathbf{w}, \mathbf{x}_i)$. The training objective is to learn parameters $\mathbf{w}$ such that each positive vertex $i \in V^+$ has a higher score than the negative vertices j it is connected to, $V_i^- = \{j \mid j \in V^-, (i, j) \in E\}$.

The combinatorial domination loss for a vertex $i \in V^+$ is 1 if there exists any vertex $j \in V_i^-$ with a higher score. A convex relaxation of the loss for the graph is given by (Dekel et al., 2003):

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log(1 + \sum_{j \in V_i^-} \exp(s_j - s_i + \Delta))$$

Taking $\Delta = 0$, and choosing $\mathbf{g}$ to be a linear feature scoring function $s_i = \mathbf{w} \cdot \mathbf{x_i}$, the loss becomes:

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log \sum_{j \in V_i^-} \exp(\mathbf{w} \cdot \mathbf{x}_j) - \mathbf{w} \cdot x_i$$

The loss over the whole graph can then be minimized using stochastic gradient descent. We will denote the ranker learned with this approach as **Rank**^{H}.

Algorithm 1: Candidate generation

```
Data: $a$, the antecedent head
Data: $t$, the target
Result: $B$, the set of possible antecedent
        boundaries $(start, end)$
begin
    $a_s \longleftarrow \mathrm{SemanticHeadVerb}(a)$;
    $E \longleftarrow \{a_s\}$ // the set of ending positions;
    for $ch \in \mathrm{RightChildren}(a_s)$ do
        $e \leftarrow \mathrm{RightMostNode}(ch)$;
        if $e < t \wedge \mathrm{ValidEnding}(e)$ then
            $E \longleftarrow E \cup \{e\}$
    $B \longleftarrow \emptyset$;
    for $e \in E$ do
        $B \longleftarrow B \cup \{(a, e)\}$;
```

3.3 Antecedent Boundary Determination

From a given antecedent head, the set of potential boundaries for the antecedent, which is a complete or partial verb phrase, is constructed using Algorithm 1.

Informally, the algorithm tries to generate different valid verb phrase structures by varying the amount of information encoded in the phrase. To do so, it accesses the semantic head verb a_s of the antecedent head a (e.g., *paying* for *are* in Example 2), and considers the rightmost node of each right child. If the node is a valid ending (punctuation and quotation are excluded), it is added to the potential set of endings E. The set of valid boundaries B contains the cross-product of the starting position $S = \{a\}$ with E.

For instance, from Example 2, the following boundary candidates are generated for *are*:

- are paying
- are paying out
- are paying out a smaller percentage of their profits and cash flow
- are paying out a smaller percentage of their profits and cash flow in the form of dividends

We experiment with both logistic regression (**Log**^{B}) and ranking (**Rank**^{B}) models for this task. The set of features is shared with the previous task, and is described in the following section.

[4] During training, there is always 1 correct antecedent for each gold standard target, with several incorrect ones.

3.3.1 Antecedent Features

The features used for antecedent head resolution and/or boundary determination try to capture aspects of both tasks. We summarize the features in Table 1. The features are roughly grouped by their type. **Labels** features make use of the parsing labels of the antecedent and target; **Tree** features are intended to capture the dependency relations between the antecedent and target; **Distance** features describe distance between them; **Match** features test whether the context of the antecedent and target are similar; **Semantic** features capture shallow semantic similarity; finally, there are a few **Other** features which are not categorized.

On the last column of the feature table, we indicate the design purpose of the feature: head selection (H), boundary detection (B) or both (B&H). However, we use the full feature set for all three tasks.

4 Joint Modeling

Here we consider the possibility that antecedent head resolution and target detection should be modeled jointly (they are typically separate). The hypothesis is that if a suitable antecedent for a target cannot be found, the target itself might have been incorrectly detected. Similarly, the suitability of a candidate as antecedent head can depend on the possible boundaries of the antecedents that can be generated from it.

We also consider the possibility that antecedent head resolution and antecedent boundary determination should be modeled independently (though they are typically combined). We hypothesize that these two steps actually focus on different perspectives: the antecedent head resolution (**H**) focuses on finding the correct antecedent position; the boundary detection step (**B**) focuses on constructing a well-formed verb phrase. We are also aware that **B** might be helpful to **H**, for instance, a correct antecedent boundary will give us correct context words, that can be useful in determining the antecedent position.

We examine the joint interactions by combining adjacent steps in our pipeline. For the combination of antecedent head resolution and antecedent boundary determination (**H+B**), we consider simultaneously as candidates for each target the set of all potential boundaries for all potential heads. Here too, a logistic regression model (**Log**$^{H+B}$) can be used to distinguish correct `(target, antecedent start, antecedent end)` triplets; or a ranking model (**Rank**$^{H+B}$) can be trained to rank the correct one above the other ones for the same target.

The combination of target detection with antecedent head resolution (**T+H**) requires identifying the targets. This is not straightforward when using a ranking model since scores are only comparable for the same target. To get around this problem, we add a “null” antecedent head. For a given target candidate, the null antecedent should be ranked higher than all other candidates if it is not actually a target. Since this produces many examples where the null antecedent should be selected, random subsampling is used to reduce the training data imbalance. The “null” hypothesis approach is used previously in ranking-based coreference systems (Rahman and Ng, 2009; Durrett et al., 2013).

Most of the features presented in the previous section will not trigger for the null instance, and an additional feature to mark this case is added.

The combination of the three tasks (**T+H+B**) only differs from the previous case in that all antecedent boundaries are considered as candidates for a target, in addition to the potential antecedent heads.

5 Experiments

5.1 Datasets

We conduct our experiments on two datasets (see Table 2 for corpus counts). The first one is the corpus of Bos and Spenader (2011), which provides VPE annotation on the WSJ section of the Penn Treebank. Bos and Spenader (2011) propose a train-test split that we follow[5].

To facilitate more meaningful comparison, we converted the sections of the British National Corpus annotated by Nielsen (2005) into the format used by Bos and Spenader (2011), and manually fixed conversion errors introduced during the process[6] (Our version of the dataset is publicly available for research[7].) We use a train-test split similar to Nielsen

[5] Section 20 to 24 are used as test data.

[6] We also found 3 annotation instances that could be deemed errors, but decided to preserve the annotations as they were.

[7] `https://github.com/hunterhector/VerbPhraseEllipsis`

Type	Feature Description	Purpose
Labels	The POS tag and dependency label of the antecedent head	H
	The POS tag and dependency label of the antecedent's last word	B
	The POS tag and lemma of the antecedent parent	H
	The POS tag, lemma and dependency label of within a 3 word around around the antecedent	B
	The pair of the POS tags of the antecedent head and the target, and of their auxiliary verbs	H
	The pair of the lemmas of the auxiliary verbs of the antecedent head and the target.	H
Tree	Whether the antecedent and the target form a comparative construction connecting by *so*, *as* or *than*	H&B
	The dependency labels of the shared lemmas between the parse tree of the antecedent and the target	H
	Label of the dependency between the antecedent and target (if exists)	H
	Whether the antecedent contains any descendant with the same lemma and dependency label as a descendant of the target.	H
	Whether antecedent and target are dependent ancestor of each other	H
	Whether antecedent and target share prepositions in their dependency tree	H
Distance	The distance in sentences between the antecedent and the target (clipped to 2)	H
	The number of verb phrases between the antecedent and the target (clipped to 5)	H
Match	Whether the lemmas of the heads, and words in the the window (=2) before the antecedent and the target match respectively	H
	Whether the lemmas of the ith word before the antecedent and $i-1$th word before the target match respectively (for $i \in \{1, 2, 3\}$, with the 0th word of the target being the target itself)	H&B
Semantic	Whether the subjects of the antecedent and the target are coreferent	H
Other	Whether the lemma of the head of the antecedent is *be* and that of the target is *do* (be-do match, used by Hardt and Nielsen)	H
	Whether the antecedent is in quotes and the target is not, or vice versa	H&B

Table 1: Antecedent Features

	Documents		VPE Instances	
	Train	Test	Train	Test
WSJ	1999	500	435	119
BNC	12	2	641	204

Table 2: Corpus statistics

(2005)[8].

5.2 Evaluation

We evaluate and compare our models following the metrics used by Bos and Spenader (2011).

VPE target detection is a per-word binary classification problem, which can be evaluated using the conventional precision (Prec), recall (Rec) and F1 scores.

Bos and Spenader (2011) propose a token-based evaluation metric for antecedent selection. The antecedent scores are computed over the correctly identified tokens per antecedent: precision is the number of correctly identified tokens divided by the number of predicted tokens, and recall is the number of

[8] Training set is CS6, A2U, J25, FU6, H7F, HA3, A19, A0P, G1A, EWC, FNS, C8T; test set is EDJ, FR3

correctly identified tokens divided by the number of gold standard tokens. Averaged scores refer to a "macro"-average over all antecedents.

Finally, in order to asses the performance of antecedent head resolution, we compute precision, recall and F1 where credit is given if the proposed head is included inside the golden antecedent boundaries.

5.3 Baselines and Benchmarks

We begin with simple, linguistically motivated baseline approaches for the three subtasks. For target detection, we reimplement the heuristic baseline used by Nielsen (2005): take all auxiliaries as possible candidates and eliminate them using part-of-speech context rules (we refer to this as $\mathbf{Pos}^T$). For antecedent head resolution, we take the first non-auxiliary verb preceding the target verb. For antecedent boundary detection, we expand the verb into a phrase by taking the largest subtree of the verb such that it does not overlap with the target. These two baselines are also used in Nielsen (2005) (and we refer to them as $\mathbf{Prev}^H$ and $\mathbf{Max}^B$, respectively).

To upper-bound our results, we include an oracle for the three subtasks, which selects the highest scoring candidate among all those considered. We denote these as $\mathbf{Ora}^T$, $\mathbf{Ora}^H$, $\mathbf{Ora}^B$.

We also compare to the current state-of-the-art target detection results as reported in Nielsen (2005) on the BNC dataset ($\mathbf{Nielsen}^T$)[9].

6 Results

The results for each one of the three subtasks in isolation are presented first, followed by those of the end-to-end evaluation. We have not attempted to tune classification thresholds to maximize F1.

6.1 Target Detection

Table 3 shows the performance of the compared approaches on the Target Detection task. The logistic regression model $\mathbf{Log}^T$ gives relatively high precision compared to recall, probably because there are so many more negative training examples than positive ones. Despite a simple set of features, the F1 results are significantly better than Nielsen's baseline $\mathbf{Pos}^T$.

Notice also how the oracle $\mathbf{Ora}^T$ does not achieve 100% recall, since not all the targets in the gold data are captured by our candidate generation strategy. The loss is around 7% for both corpora.

The results obtained by the joint models are low on this task. In particular, the ranking models $\mathbf{Rank}^{T+H}$ and $\mathbf{Rank}^{T+H+B}$ fail to predict any target in the WSJ corpus, since the null antecedent is always preferred. This happens because joint modeling further exaggerates the class imbalance: the ranker is asked to consider many incorrect targets coupled with all sorts of hypothesis antecedents, and ultimately learns just to select the null target. Our initial attempts at subsampling the negative examples did not improve the situation. The logistic regression models $\mathbf{Log}^{T+H}$ and $\mathbf{Log}^{T+H+B}$ are most robust, but still their performance is far below that of the pure classifier $\mathbf{Log}^T$.

6.2 Antecedent Head Resolution

Table 4 contains the performance of the compared approaches on the Antecedent Head Resolution task, assuming oracle targets ($\mathbf{Ora}^T$).

First, we observe that even the oracle $\mathbf{Ora}^H$ has low scores on the BNC corpus. This suggests that some phenomena beyond the scope of those observed in the WSJ data appear in the more general corpus (we developed our system using the WSJ annotations and then simply evaluated on the BNC test data).

Second, the ranking-based model $\mathbf{Rank}^H$ consistently outperforms the logistic regression model $\mathbf{Log}^H$ and the baseline $\mathbf{Prev}^H$. The ranking model's advantage is small in the WSJ, but much more pronounced in the BNC data. These improvements suggest that indeed, ranking is a more natural modeling choice than classification for antecedent head resolution.

Finally, the joint resolution models $\mathbf{Rank}^{H+B}$ and $\mathbf{Log}^{H+B}$ give poorer results than their single-task counterparts, though $\mathbf{Rank}^{H+B}$ is not far behind $\mathbf{Rank}^H$. Joint modeling requires more training data and we may not have enough to reflect the benefit of a more powerful model.

6.3 Antecedent Boundary Determination

Table 5 shows the performance of the compared approaches on the Antecedent Boundary Determination task, using the soft evaluation scores (the results for

[9] The differences in the setup make the results on antecedent resolution not directly comparable.

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{T}	100.00	93.28	96.52	100.00	92.65	96.18
Log^{T}	80.22	61.34	69.52	80.90	70.59	75.39
Pos^{T}	42.62	43.7	43.15	35.47	35.29	35.38
Log^{T+H}	23.36	26.89	25.00	12.52	38.24	18.86
Rank^{T+H}	0.00	0.00	0.00	15.79	5.88	8.57
Log^{T+H+B}	25.61	17.65	20.90	21.50	32.35	25.83
Rank^{T+H+B}	0.00	0.00	0.00	16.67	11.27	13.45
Nielsen^{T}	-	-	-	72.50	72.86	72.68

Table 3: Results for Target Detection

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}	94.59	88.24	91.30	79.89	74.02	76.84
Rank^{H}	70.27	65.55	67.83	52.91	49.02	50.89
Prev^{H}	67.57	63.03	65.22	39.68	36.76	38.17
Log^{H}	59.46	55.46	57.39	38.62	35.78	37.15
Rank^{H+B}	68.47	63.87	66.09	51.85	48.04	49.87
Log^{H+B}	39.64	36.97	38.26	30.16	27.94	29.01

Table 4: Results for Antecedent Head Resolution

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^{B}	89.47	83.46	86.36	81.10	75.13	78.00
Rank^{B}	83.96	78.32	81.04	75.68	70.12	72.79
Max^{B}	78.97	73.66	76.22	73.70	68.28	70.88

Table 5: Soft results for Antecedent Boundary Determination

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Rank^{H}**+Log**^{B}	64.11	59.8	61.88	47.04	43.58	45.24
Rank^{H}**+Rank**^{B}	63.90	59.6	61.67	49.11	45.5	47.24
Log^{H}**+Log**^{B}	53.49	49.89	51.63	34.77	32.21	33.44
Log^{H}**+Rank**^{B}	53.27	49.69	51.42	36.26	33.59	34.88
Rank^{H+B}	67.55	63.01	65.20	50.68	46.95	48.74
Log^{H+B}	40.96	38.20	39.53	30.00	27.79	28.85

Table 6: Soft results for Antecedent Boundary Determination with non-gold heads

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^T**+Ora**^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^T**+Rank**^{H}**+Rank**^{B}	52.68	40.28	45.65	43.03	37.54	40.10
Log^T**+Rank**^{H}**+Log**^{B}	52.82	40.40	45.78	40.21	35.08	37.47
Log^T**+Log**^{H}**+Rank**^{B}	49.45	37.82	42.86	33.12	28.90	30.86
Log^T**+Log**^{H}**+Log**^{B}	49.41	37.79	42.83	31.32	27.33	29.19
Pos^T**+Prev**^{H}**+Max**^{B}	19.04	19.52	19.27	12.81	12.75	12.78
Log^T**+Rank**$^{H+B}$	54.82	41.92	47.51	41.86	36.52	39.01
Log^T**+Log**$^{H+B}$	38.85	29.71	33.67	26.11	22.78	24.33

Table 7: Soft end-to-end results

the strict scores are omitted for brevity, but in general look quite similar). The systems use the output of the oracle targets (**Ora**^{T}) and antecedent heads (**Ora**^{H}).

Regarding boundary detection alone, the logistic regression model **Log**^{B} outperforms the ranking model **Rank**^{B}. This suggests that boundary determination is more a problem of determining the compatibility between target and antecedent extent than one of ranking alternative boundaries. However, the next experiments suggest this advantage is diminished when gold targets and antecedent heads are replaced by system predictions.

6.3.1 Non-Gold Antecedent Heads

Table 6 contains Antecedent Boundary Determination results for systems which use oracle targets, but system antecedent heads. When **Rank**^{H} or **Log**^{H} are used for head resolution, the difference between **Log**^{B} and **Rank**^{B} diminishes, and it is even better to use the latter in the BNC corpus. The models were trained with gold annotations rather than system outputs, and the ranking model is somewhat more robust to noisier inputs.

On the other hand, the results for the joint resolution model **Rank**$^{H+B}$ are better in this case than the combination of **Rank**^{H}**+Rank**^{B}, whereas **Log**$^{H+B}$ performs worse than any 2-step combination. The benefits of using a ranking model for antecedent head resolution seem thus to outperform those of using classification to determine its boundaries.

6.4 End-to-End Evaluation

Table 7 contains the end-to-end performance of different approaches, using the soft evaluation scores.

The trends we observed with gold targets are preserved: approaches using the **Rank**^{H} maintain an advantage over **Log**^{H}, but the improvement of **Log**^{B} over **Rank**^{B} for boundary determination is diminished with non-gold heads. Also, the 3-step approaches seem to perform slightly better than the 2-step ones. Together with the fact that the smaller problems are easier to train, this appears to validate our decomposition choice.

7 Conclusion and Discussion

In this paper we have explored a decomposition of Verb Phrase Ellipsis resolution into subtasks, which splits antecedent selection in two distinct steps. By modeling these two subtasks separately with two different learning paradigms, we can achieve better performance then doing them jointly, suggesting they are indeed of different underlying nature.

Our experiments show that a logistic regression classification model works better for target detection and antecedent boundary determination, while a ranking-based model is more suitable for selecting the antecedent head of a given target. However, the benefits of the classification model for boundary determination are reduced for non-gold targets and heads. On the other hand, by separating the two steps, we lose the potential joint interaction of them. It might be possible to explore whether we can bring the benefits of the two side: use separate models on each step, but learn them jointly. We leave further investigation of this to future work.

We have also explored jointly training a target detection and antecedent resolution model, but have not been successful in dealing with the class imbalance inherent to the problem.

Our current model adopts a simple feature set, which is composed mostly by simple syntax and lexical features. It may be interesting to explore more semantic and discourse-level features in our system. We leave these to future investigation.

All our experiments have been run on publicly available datasets, to which we add our manually aligned version of the VPE annotations on the BNC corpus. We hope our experiments, analysis, and more easily processed data can further the development of new computational approaches to the problem of Verb Phrase Ellipsis resolution.

Acknowledgments

The first author was partially supported DARPA grant FA8750-12-2-0342 funded under the DEFT program. Thanks to the anonymous reviewers for their useful comments.

References

Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, 45(4):463–494.

Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452.

Ofer Dekel, Yoram Singer, and Christopher D. Manning. 2003. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems*, page None.

Greg Durrett, David Hall, and Dan Klein. 2013. Decentralized Entity-Level Modeling for Coreference Resolution. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 114–124.

Daniel Hardt. 1992. An algorithm for VP ellipsis. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, number January, pages 9–14.

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

Daniel Hardt. 1998. Improving Ellipsis Resolution with Transformation-Based Learning. *AAAI Fall Symposium*, pages 41–43.

Leif Arda Nielsen. 2003a. A corpus-based study of Verb Phrase Ellipsis Identification and Resolution. In *Proceedings of the 6th Annual CLUK Research Colloquium*, page Proceedings of the 6th Annual CLUK Research Colloq.

Leif Arda Nielsen. 2003b. Using Machine Learning techniques for VPE detection. In *Proceedings of Recent Advances in Natural Language Processing*, pages 339–346.

Leif Arda Nielsen. 2004a. Robust VPE detection using automatically parsed text. In *Proceedings of the Student Workshop, ACL 2004*, pages 31–36.

Leif Arda Nielsen. 2004b. Verb phrase ellipsis detection using automatically parsed text. In *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*.

Leif Arda Nielsen. 2005. *A corpus-based study of Verb Phrase Ellipsis Identification and Resolution*. Doctor of philosopy, King's College London.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, number August, pages 968–977.

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