A Neural Network Approach to Verb Phrase Ellipsis Resolution

Wei-Nan Zhang[†], Yue Zhang[‡*], Yuanxing Liu[†], Donglin Di[†], Ting Liu[†]
[†]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology
[‡]School of Engineering, Westlake University

Abstract

Verb Phrase Ellipsis (VPE) is a linguistic phenomenon, where some verb phrases as syntactic constituents are omitted and typically referred by an auxiliary verb. It is ubiquitous in both formal and informal text, such as news articles and dialogues. Previous work on VPE resolution mainly focused on manually constructing features extracted from auxiliary verbs, syntactic trees, etc. However, the optimization of feature representation, the effectiveness of continuous features and the automatic composition of features are not well addressed. In this paper, we explore the advantages of neural models on VPE resolution in both pipeline and end-to-end processes, comparing the differences between statistical and neural models. Two neural models, namely multi-layer perception and the Transformer, are employed for the subtasks of VPE detection and resolution. Experimental results show that the neural models outperform the state-of-the-art baselines in both subtasks and the end-to-end results.

Introduction

Ellipsis is a linguistic phenomenon where some syntactic constituents are omitted but can be reconstructed from context. One type of ellipsis, Verb Phrase Ellipsis (VPE), denotes the omission of verb phrases. In English, a VPE is usually associated with an auxiliary verb without a verb phrase. For example, in the sentence "*Not only is development of the new company's initial machine tied directly to Mr. Cray, so is its balanced sheet.*", the verb phrase "*tied directly to Mr. Cray*" is omitted for "*its balanced sheet*" with the auxiliary verb "*is*" being given. In the above instance, the auxiliary verb is usually called a **trigger** and the omitted verb phrase the **antecedent**. Given a sentence, there are two NLP tasks associated with VPE, namely **VPE detection**, which is to detect a trigger verb, and **VPE resolution**, which is to identify the antecedent of a given trigger (Hardt 1992; Nielsen 2003b; 2005; Van Craenenbroeck 2017). Figure 1 shows an example VPE resolution process.

VPE occurs frequently in both formal and informal texts, such as news articles and dialogues (Nielsen 2005). Thus resolving VPE is important for downstream NLP tasks such as event extraction, dialogue systems, etc (Kenyon-Dean, Cheung, and Precup 2016). Previous work on VPE resolution has addressed the sub-tasks of VPE detection (Hardt 1992; 1997; Hobbs and Kehler 1997; Nielsen 2003b; 2004a; 2004b) and VPE resolution (Hardt 1998; Nielsen 2003a; 2005; Bos 2005) separately, pipeline processing of VPE detection and VPE resolution (Bos and Spenader 2011; Bos 2012; Kawai 2013; Liu, Pellicer, and Gillick 2016; Bakhshandeh, Wellwood, and Allen 2016), and end-to-end modeling of the two steps (Kenyon-Dean, Cheung, and Precup 2016; McShane and Babkin 2016). Most existing work uses heuristic rules, manually constructed features extracted from auxiliaries and syntactic structures, and linguistic theories such as Discourse Representation Structure (DRS) (Bos 2012), Simple Parallel Configuration (SPC) (McShane and Babkin 2016), etc[1]. Despite the success of existing work, heuristic rules and manual features are sparse and cannot fully explored deep semantic information across a sentence. Such limitations can potentially be addressed by using neural network models (Collobert and Weston 2008).

In this paper, we explore neural network models for VPE resolution in both pipeline and end-to-end processes, com-

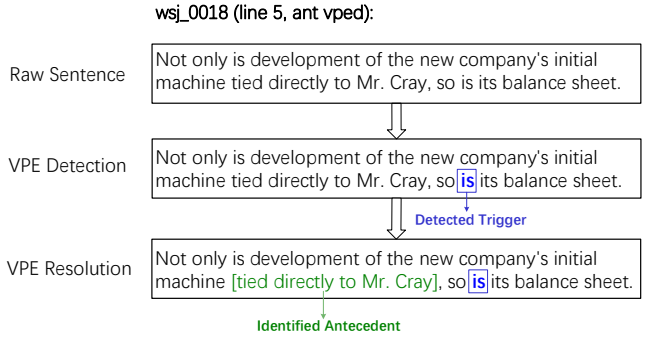

Figure 1: An example of VPE resolution process, where "**is**" and the VP in square bracket "tied directly to Mr. Cray" denote the trigger and antecedent of VPE, respectively.

[*]Corresponding author

[1]Here, DRS and SPC are proposed based on the discourse representation theory (Kamp, Reyle, and others 1993) and parallelism theory (Hobbs and Kehler 1997), respectively.

paring statistical models and neural models while investigating novel lexical and slot pattern features. For VPE detection, we choose an SVM model with non-linear kernel function as a state-of-the-art statistical model, a simple multilayer perception (MLP) and the Transformer (Vaswani et al. 2017) as our neural models. For VPE resolution, we apply a MLP and the Transformer models. Finally, for the end-to-end VPE resolution task, we propose a novel neural framework to uniformly integrate the two subtasks.

Results on a WSJ dataset (Bos and Spenader 2011) show that the proposed neural models outperform the state-of-the-art baselines. In addition, we further analyze the distribution of the VPE phenomenon through data annotation, which includes the extended annotation of sentences that 1) have trigger verb and VPE, 2) have trigger verb but no VPE, 3) have VPE but no trigger, 4) have no trigger and no VPE. We release the extended corpus and code for VPE resolution research.

Related Work

VPE Detection Hardt (1992) proposed a 3-step algorithm, which includes 3 operations to remove, assign and select on VPlist and VPE. Hardt (1997) was the first empirical study on VPE detection. They utilized syntactic-based matching pattern and proposed 4 preference factors to identify the VPE. Hobbs and Kehler (1997) introduced a parallelism theory in discourse to analyze the VPE phenomenon. Nielsen (2003b) verified the performance of different statistical machine learning approaches on VPE detection and explored the combination of these approaches. Nielsen (2004a) and Nielsen (2004b) verified the performance of VPE detection (or target detection) in the automatically parsed data and built a robust, accurate and domain independent VPE detection system. We follow Hardt (1997), Nielsen (2004a) and Nielsen (2004b) in utilizing lexical and syntactic features, but are the first apply neural models to explore deep semantic information across a sentence for VPE detection.

VPE Resolution Hardt (1998) employed a transformation learning-based approach to generated patterns for VPE resolution. Nielsen (2003a) first proposed a corpus based approach to VPE resolution. A 3-step end-to-end approach, which includes VPE detection, antecedent identification and VPE resolution, was proposed as a pipeline framework. Bos (2005) analyzed the VPE and sloppy identity through case study. All the above work uses statistical features, while we employ neural models to optimize the feature representation and composition for VPE resolution.

Although there are some previous research of VPE detection and resolution on the BNC dataset, it is difficult to use as it depends on a particular set of tools for preprocessing (Kenyon-Dean, Cheung, and Precup 2016). Recently, Bos and Spenader (2011) extended and released an annotated corpus of VPE in 25 sections of the WSJ corpus distributed with the Penn Treebank dataset. Bos (2012) utilized the Discourse Representation Structure (DRS) to resolve the VPE detection, location and resolution tasks. Kawai (2013) analyzed the identity condition on VPE and provided a preliminary formulation for the nondistinctness condition. Liu, Pellicer, and Gillick (2016) explored the steps of VPE and splitted the target detection and antecedent identification into three tasks as target detection, antecedent head resolution and antecedent boundary determination. Bakhshandeh, Wellwood, and Allen (2016) proposed a framework to jointly model the comparison and ellipsis as an interconnected structure of predicate-argument. McShane and Babkin (2016) considered the syntactic parallelism, modality correlation and sentence constituents for VPE detection and resolution. Kenyon-Dean, Cheung, and Precup (2016) explored auxiliary, lexical, syntactic features and proposed an alignment algorithm based MIRA approach for VPE resolution. All the above work mainly focuses on the heuristic rules and manual features under statistical models on both two separate tasks and the end-to-end task. In contrast, our work explores the advantages of neural models for modeling the deep semantics across a sentence on all the 3 tasks.

The Proposed Approach

We investigate an end-to-end learning framework to integrate the VPE detection and VPE resolution as shown in Figure 2. We provide an alternative choices of models in each steps. For example, for VPE detection, the classifier can be SVM, MLP or Transformer. For VPE resolution, the model can be either MLP or Transformer.

VPE Detection

Previous research (Nielsen 2003b; 2004a; 2004b; 2005; Bos 2012; Liu, Pellicer, and Gillick 2016; Kenyon-Dean, Cheung, and Precup 2016) regards VPE detection as a binary classification task over auxiliary verbs. In this paper, we follow the task specification. Given an auxiliary verb and the sentence where the auxiliary verb is in as input, VPE detection is to extract features and predict whether the auxiliary verb is a trigger or not.

For VPE detection, we follow the features proposed by Kenyon-Dean, Cheung, and Precup (2016), but investigate a range of novel features. Table 1 summarizes the proposed lexical features extended from (Kenyon-Dean, Cheung, and Precup 2016), and slot pattern features. For extended lexical features, we aim to capture the distributional semantics of words (No.1-3) and POS tag (No.4) as well as the sequential context information of POS tags (No.5-10). For the slot pattern feature, we proposed to explore the auxiliary related syntactical structures. The slot is set to generalize the matching scope. In particular, No.11-16 are related to the auxiliaries. No.17 and 18 are for a specific phenomenon "the same" reference and No.19 is for the phenomenon of "comparative deletion". Both phenomena are explored by Hobbs and Kehler (1997).

We compare the effectiveness of SVM and an attention-based neural network model. For the SVM classifier, we used the scikit-learn[2] with 5-fold cross validation for training and test. For the attention-based neural network model, we utilized Transformer (Vaswani et al. 2017). The data s-

[2] http://scikit-learn.org/stable/modules/svm.html#svm

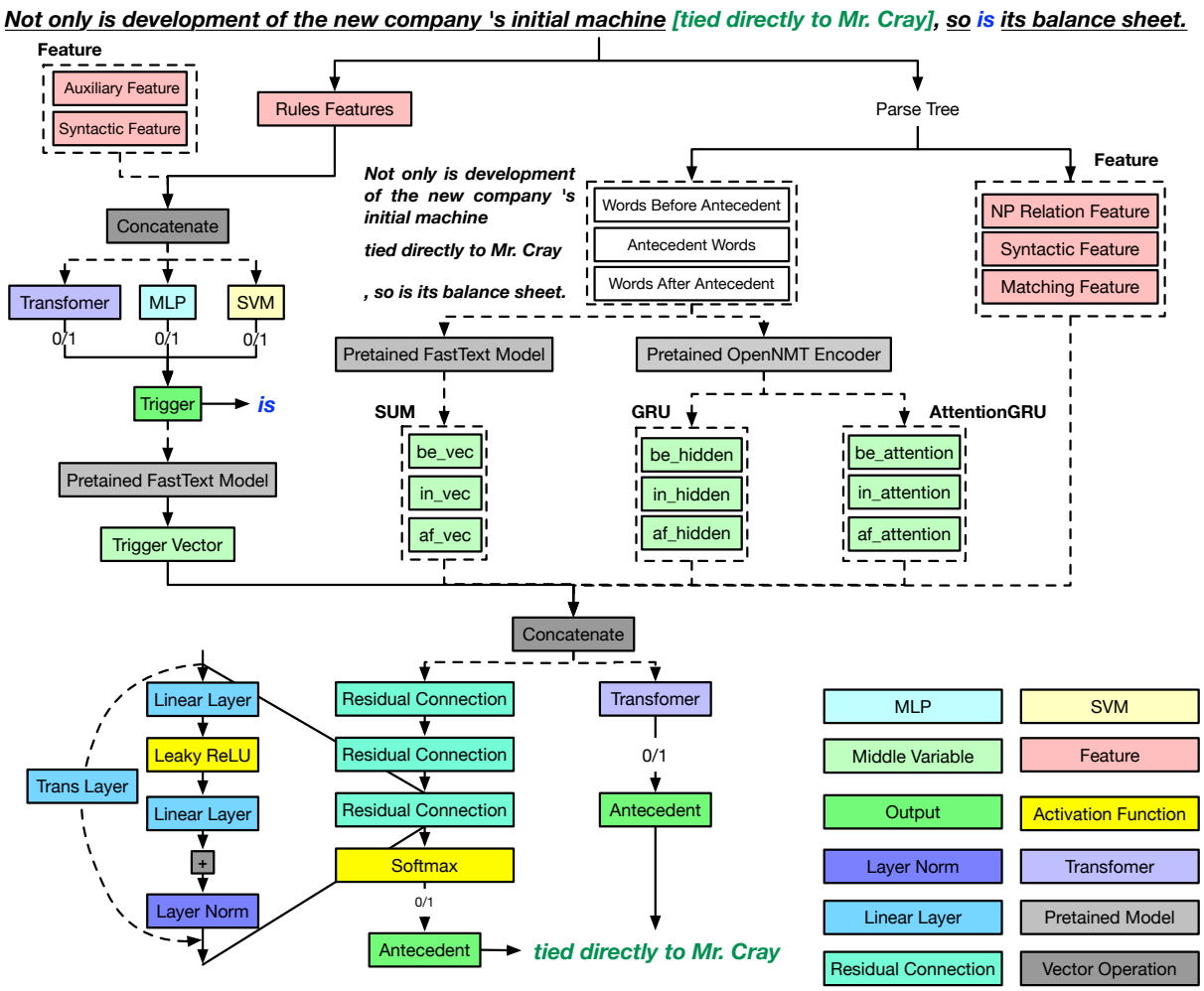

Figure 2: Framework of the proposed end-to-end approach to VPE resolution.

wsj_0018 (line 5, ant vped):

Not only is development of the new company's initial machine [tied directly to Mr. Cray], so **is** its balance sheet.

a_i t c_i

Figure 3: Example VPE training data. The VP in square bracket denotes a candidate antecedent a_i, the underlined text represents the context (c_i) of a_i and "**is**" is a trigger (t) for VPE.

plit and feature input to the Transformer model is the same as the SVM classifier.

VPE Resolution

VPE resolution is defined as a binary classification task over verb phrases and adverb phrases. Formally, given a sentence, let a_i, t and c_i denote the i-th candidate antecedent, trigger and the context, respectively, the VPE resolution problem is then described as:

$$f(a_i, t, c_i) \rightarrow \{0, 1\} \quad (1)$$

where 1 denotes that the candidate antecedent a_i is the correct antecedent of trigger t and 0 vice versa. Figure 3 is an example. In this paper, we first learn the joint representation of a_i, t and c_i and then resolve VPE with two neural network models.

Representation Learning To obtain a joint representation of a_i, t and c_i, we initially obtain word representations using fastText[3] on the WSJ corpus[4] except for the 554 training and test instances. The joint representation of a_i, t and c_i are then obtained using three different functions.

- **Sum Pooling** directly sums the word embeddings of a_i, t and c_i, respectively, to obtain their representations. $w_{a_{ij}}$

[3] https://fasttext.cc/
[4] https://catalog.ldc.upenn.edu/ldc99t42

No.	Extended Lexical Feature
1	Current word w_i
2	Previous word w_{i-1}
3	Next word w_{i+1}
4	The string of POS tags of w_i
5	The string of POS tags of $w_{i-3}, w_{i-2}, w_{i-1}$
6	The string of POS tags of w_{i-2}, w_{i-1}
7	The string of POS tags of w_{i-1}
8	The string of POS tags of w_{i+1}
9	The string of POS tags of w_{i+1}, w_{i+2}
10	The string of POS tags of $w_{i+1}, w_{i+2}, w_{i+3}$
	Slot Pattern Feature
11	,/. so/or/nor/but/while [*slot*] do/to/did/does
12	as [*slot*] were/do/does/did
13	if it is/does/isnt
14	,/. have [*slot*] ,/.
15	[*slot*] wasnt/ would/ do/ might/have to [*slot*] ,/.
16	all the/the way/that/who/and [*slot*] does/will/can
17	the same [*slot*] do
18	doing/do [*slot*] the same/so
19	than [*slot*] do/is/had/has

Table 1: Summary of the proposed extended lexical features and slot pattern features for VPE detection.

and $w_{c_{ik}}$ denote the vector representation of the j-th and k-th word in the i-th candidate antecedent and the i-th context, respectively.

$$v_{Sum} = \sum_j w_{a_{ij}} + \sum_k w_{c_{ik}} + t \tag{2}$$

- **GRU** uses a GRU layer (Chung et al. 2014) to obtain the representations of a_i and c_i, which are the outputs of the last hidden state of two separate GRU models, respectively. The inputs to the GRU models are word embeddings. We use x_t to represent the embedding of word in candidate antecedent, trigger and the context. Initially, for $t = 0$, the output vector is $h_0 = 0$. The GRU model can be formally represented as:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{3}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{4}$$

$$v_{GRU} = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{5}$$

  Here x_t, h_t, z_t, r_t denote the input vector, output vector, update gate vector and reset gate vector, respectively. W and U are parameter matrices and b_* is bias. σ_g and σ_h represent the sigmoid function and the hyperbolic tangent function, respectively.

- **Attention-based GRU** uses attention-based GRU (Bahdanau, Cho, and Bengio 2014) to obtain the representations of a_i and c_i, which are the outputs of the weighted sum of each hidden state on GRU model. Given the word embedding sequence $x_1, ..., x_t$, we measure the similarity between each word and the last word as follow:

$$s_i = \text{sim}(x_i, x_t) = \frac{x_i \cdot x_t}{\|x_i\| \cdot \|x_t\|} \tag{6}$$

  Then, we calculate the attention (weight) of each word.

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=0}^{n} \exp(s_j)} \tag{7}$$

  Finally, the encoded vector v_{AttGRU} of the sentence is calculated as:

$$v_{AttGRU} = \sum_{i=0}^{n} \alpha_i h_i \tag{8}$$

  Where h_i is the output vector of GRU at time step i. Note that the representation of t equals to its word embedding in the above three different functions.

Neural Model For VPE resolution, we utilized two neural network models. The first is a multi-layer perception (MLP). The linear layer is defined as:

$$\text{Linear}(x) = xA^T + b \tag{9}$$

where x is the input, A is the parameter matrix and b is a bias. Then we use LeakyReLU as activation function, which is defined as:

$$\text{LeakyRelu}(x) = \max(0, x) + ns * \min(0, x) \tag{10}$$

where ns controls the angle of the negative slope and its default value equals to $1e$-2. Our MLP is defined as:

$$f_{l_1} = \text{LeakyRelu}(\text{Linear}(x)) \tag{11}$$

$$f_{l_2} = \text{LeakyRelu}(\text{Linear}(f_{l_1})) \tag{12}$$

$$f_{l_3} = \text{LeakyRelu}(\text{Linear}(f_{l_2})) \tag{13}$$

$$f_{l_4} = \text{LeakyRelu}(\text{Linear}(f_{l_3})) \tag{14}$$

$$y = \text{softmax}(\text{Linear}(f_{l_4})) \tag{15}$$

where x denotes the input vector which is obtained by concatenating the representations of a_i, t and c_i, and y is the output vector.

The second VPE resolution model is the Transformer model (Vaswani et al. 2017), which is based on self-attention. In our task, we model the binary classification process as a self-attention based encoder as shown in Equation (16).

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{16}$$

Here Q, K and V denote the matrices of query, key and value, respectively, as described in Vaswani et al. (2017). d_k is the dimension of the vector of a key in K. In our task, Q, K and V is the same vector, which comes from the output of previous layer in encoder. The initialization of Q, K and V equals to the concatenation of the representations of a_i, t and c_i. Besides the scaled dot-product attention as shown in Equation (16), the Transformer model also leverages a multi-head attention mechanism as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{17}$$

Auxiliary Type	Trigger	VPE Freq	Sum (%)
DO	do	213	213(38.45%)
BE	be	108	108 (19.49%)
HAVE	have	44	44 (7.94%)
TO	to	29	29 (5.23%)
MODAL	can	29	93 (16.79%)
	will	26	
	would	14	
	could	11	
	should	7	
	might	4	
	may	1	
	must	1	
SO	so	54	67 (12.09%)
	same	8	
	likewise	3	
	opposite	2	
TOTAL			554

Table 2: Statistics of auxiliary categories and the corresponding trigger words and VPE in experimental dataset. **Auxiliary**, **Trigger** and **VPE Freq** denote the auxiliary type, trigger words and VPE frequency, respectively.

Here W^O, W_i^Q, W_i^K and W_i^V are parameter matrices. Then a feed-forward network (FFN) with two linear layers and ReLU activation function is used. Let $x = \text{LayerNorm}(\text{MultiHead}(Q, K, V) + \text{Sublayer}(PE))$, the FFN is as follows:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (18)$$

where PE denotes the result of positional encoding. Sublayer is the function implemented by the sub-layer itself (Vaswani et al. 2017). W_1, W_2 are parameter matrices and b_1, b_2 are biases. The output is then obtained by:

$$y = \text{softmax}(\text{Linear}(\text{FFN}(x) + \text{Sublayer}(x))) \quad (19)$$

where the form of Linear function is similar to Equation (9) but with different parameters[5]. The neural models can leverage manual features in addition to dense representation features which are obtained through the representation learning, but also can use the manually constructed features which are depended on the experience of experts. In addition to neural features, we also integrate the features of Kenyon-Dean, Cheung, and Precup (2016).

Experiments

We conduct experiments on standard benchmarks for verifying the effectiveness of neural models, the performance of the 3 functions on obtaining context representations, and the usefulness of traditional manual features on improving the performance of neural models. We analyze the additional VPE phenomena through manual annotating 820 sentences, and release the annotated sentences as an extended data.

[5]For more details of the Transformer model, please see the original paper (Vaswani et al. 2017).

Dataset

We use the dataset released by Bos and Spenader (2011). Following Kenyon-Dean, Cheung, and Precup (2016), we divide auxiliaries into six types, including DO, BE, HAVE, TO, MODAL and SO, as shown in Table 2. For the VPE detection task, the training set contains 554 sentences with trigger words as positive instances and 554 sentences. For the VPE resolution task, we first obtain the syntactic tree of each sentence using Berkeley parser[6]. We then use NLTK[7] to extract all the verb phrases (VP) and adjective phrases (ADJP) in a sentence as candidate antecedents of VPE. The training set also contains 554 sentences with ground truth antecedents as positive instances, and the randomly sampled antecedents from the rest of the extracted VPs and ADJPs as negative instances.

Baselines

Three state-of-the-art baselines are selected for VPE detection and VPE resolution, respectively. For VPE detection, the first two baselines include a rule-based approach (**Rule**) and a machine learning based approach (**ML**). Both are proposed by Kenyon-Dean, Cheung, and Precup (2016). The third baseline is a 3-step approach proposed by Liu, Pellicer, and Gillick (2016), which includes VPE detection, antecedent identification and VPE resolution.

For VPE resolution, the first baseline is a DR theory-based VPE resolution approach (**DRVPE**) proposed by Bos (2012). The second is a Margin-Infused-Relaxed-Algorithm based approach (**MIRA**) for VPE resolution, which is proposed by Kenyon-Dean, Cheung, and Precup (2016). The third is the 3-step VPE resolution approach of Liu, Pellicer, and Gillick (2016). We directly use the experimental results of VPE detection and VPE resolution from Kenyon-Dean, Cheung, and Precup (2016) and Liu, Pellicer, and Gillick (2016) for comparison.

Parameter Settings

VPE detection. For the SVM model, the hyper parameter $C = 100, \gamma = 0.5$ and the kernel function is "RBF" . For the MLP model, the size of hidden state is 1,024, the learning rate equals 0.005 with 1,000 training epochs and the batch size is 64.

VPE resolution. For the MLP model, the batch size equals to 64, learning rate and weight decay are both 0.005. We use a cross entropy loss and Adam optimization (Kingma and Ba 2014). For the Transformer model, we use the default parameter settings of Vaswani et al. (2018) except that the beam size equals to 4 and length penalty $\alpha = 0.5$ for both VPE detection and VPE resolution.

Results

VPE Detection We have two experimental settings according to the split of the training and test data. Table 3 shows the accuracies of VPE detection in 5-fold cross validation. We can see from Table 3 that **ML** and **SVM** give

[6]https://github.com/slavpetrov/berkeleyparser

[7]https://www.nltk.org/

Auxiliary	Rule	ML	SVM†	SVM+$\boldsymbol{F}^{\dagger}_{\ddagger}$	MLP	MLP+$\boldsymbol{F}^{\dagger}_{\ddagger}$	Transformer†	Transformer+$\boldsymbol{F}^{\dagger}_{\ddagger}$
DO	0.83	0.89	0.94	0.93	0.88	0.94	0.85	**0.96**
BE	0.34	0.63	0.71	0.76	0.66	0.89	0.60	**0.89**
HAVE	0.43	0.75	0.76	**0.90**	0.67	0.83	0.77	**0.90**
TO	0.76	0.79	0.64	0.86	0.72	0.91	0.79	**0.98**
MODAL	0.80	0.86	0.95	0.95	0.70	0.82	0.88	**0.97**
SO	0.67	0.86	0.91	0.90	0.80	0.90	0.88	**0.98**
ALL	0.71	0.82	0.87	0.90	0.87	0.95	0.88	**0.96**

Table 3: The F1 scores of VPE detection obtained with 5-fold cross validation. **Rule** and **ML** are baselines. **SVM**, **SVM+*F***, **MLP**, **MLP+*F***, **Transformer** and **Transformer+*F*** are the approaches in VPE detection. † and ‡ indicate the experimental results are statistically significant to the results of **Rule**, **ML**, respectively, with $p < 0.05$. The results in bold indicate the best performance.

Auxiliary		DO	BE	HAVE	TO	MODAL	SO	ALL
DRVPE		0.42	0.37	0.42	0.15	0.39	0.03	0.36
MIRA		0.71	0.63	0.67	0.53	0.61	0.58	0.65
MLP	Sum$^{\dagger}_{\ddagger}$	0.65	0.67	0.69	0.63	0.63	0.64	0.66
	Sum+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.76	0.76	0.78	0.74	0.73	0.74	0.76
	GRU†	0.50	0.52	0.51	0.45	0.51	0.48	0.51
	GRU+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.81	0.75	0.78	0.71	0.84	0.75	0.78
	AttGRU†	0.53	0.53	0.51	0.54	0.53	0.51	0.53
	AttGRU+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.89	0.84	0.86	0.84	0.89	0.86	0.87
Transformer	Sum†	0.77	0.65	0.56	0.60	0.54	0.67	0.83
	Sum+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.78	0.70	0.75	0.77	0.75	0.85	0.88
	GRU†	0.74	0.67	0.65	0.73	0.79	0.68	0.85
	GRU+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.87	0.84	**0.90**	0.87	0.84	**0.89**	0.90
	AttGRU$^{\dagger}_{\ddagger}$	0.84	0.71	0.74	0.76	0.68	0.64	0.84
	AttGRU+$\boldsymbol{F}^{\dagger}_{\ddagger}$	**0.93**	**0.93**	0.89	**0.91**	**0.93**	0.85	**0.94**

Table 4: VPE resolution results on accuracy obtained with 5-fold cross validation. Here, **DRVPE** and **MIRA** are two baselines. **Sum**, **GRU** and **AttGRU** denote the three encoding mechanisms, namely sum-pooling, GRU and attention-based GRU, respectively. **Sum+*F***, **GRU+*F*** and **AttGRU+*F*** represent the three encoding mechanisms with the manually constructed features, respectively. † and ‡ denote the performance is statistical significant over the baselines of **DRVPE** and **MIRA**, respectively, with $p < 0.05$. The results in bold indicate the best performance.

comparable performance, where the **SVM** model uses our proposed features as shown in Table 1. The Transformer model outperforms the **ML** and **SVM** models with the same features demonstrating that the neural model (Transformer) can better explore the composition of features and improve the performance of prediction.

In addition to the proposed features shown in Table 1, for empirical comparison, we also integrate the features proposed by Kenyon-Dean, Cheung, and Precup (2016)(**ML**), denoted by ***F***, We can see that both the performance of SVM and Transformer models improve by integrating the features used by **ML** model. We conclude that: first, the features proposed by Kenyon-Dean, Cheung, and Precup (2016) can be integrated into the neural model (Transformer) to further improve the performance of VPE detection; second, to compare the performance of **SVM+*F*** and **Transformer+*F***, the neural model (Transformer) can better utilize the features than the statistical model (SVM) on the VPE detection task.

We also compare the precision (**P**), recall (**R**) and F1 score (**F**) of VPE detection in the train-test data split used by Bos and Spenader (2011), Liu, Pellicer, and Gillick (2016) and Kenyon-Dean, Cheung, and Precup (2016) (ML). The experimental results are shown in Table 5. We can see that SVM, MLP and Transformer outperform the baseline models (**Rule** and **ML**). It again verifies that the features proposed by Kenyon-Dean, Cheung, and Precup (2016) can further improve the performance of both the statistical models and neural models.

VPE Resolution We follow the setting of Kenyon-Dean, Cheung, and Precup (2016) on the split of train and test data. As presented in the above section, the representations of the antecedent a_i, the context (c_i) of a_i and the trigger (t) are obtained using three different functions. Besides the proposed three measures, we also consider to integrate the features that proposed by Kenyon-Dean, Cheung, and Precup (2016) to enrich the representations of a_i, c_i and t. Therefore, for

Test Set Results	P	R	F
Liu et al. (2016)	0.8022	0.6135	0.6953
ML	0.7574	0.8655	0.8078
SVM†	0.8803	0.8782	0.8780
SVM+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.9048	0.9034	0.9033
MLP	0.8268	0.8824	0.8537
MLP+$\boldsymbol{F}^{\dagger}$	0.9304	0.8992	0.9145
Transformer$_{\ddagger}$	0.8504	0.9076	0.8780
Transformer+$\boldsymbol{F}^{\dagger}_{\ddagger}$	**0.9569**	**0.9328**	**0.9447**

Table 5: The precision (**P**), recall (**R**) and F1 score (**F**) of VPE detection obtained with the train-test split used by Bos and Spenader (2011), Liu, Pellicer, and Gillick (2016) (Liu et al. (2016)) and (Kenyon-Dean, Cheung, and Precup 2016) (ML). † and ‡ denote the performance is statistical significant over the baselines of Liu et al. (2016)) and ML, respectively, with $p < 0.05$. The results in bold indicate the best performance.

End-to-end Results		P	R	F
Liu et al. (2016)		0.5482	0.4192	0.4751
MIRA		0.4871	0.5567	0.5196
MLP	Sum$_{\ddagger}$	0.5834	0.8035	0.6760
	Sum+$\boldsymbol{F}^{\dagger}_{\ddagger}$	0.5724	0.9200	0.7057
	GRU$^{\dagger}_{\ddagger}$	**0.6396**	0.6801	0.6592
	GRU+$\boldsymbol{F}$	0.5507	0.8936	0.6814
	AttGRU$_{\ddagger}$	0.6052	0.8348	0.7017
	AttGRU+$\boldsymbol{F}_{\ddagger}$	0.6024	**0.9226**	**0.7289**
Transformer	Sum	0.4464	0.8739	0.5909
	Sum+$\boldsymbol{F}$	0.4856	0.8487	0.6177
	GRU	0.4414	0.8235	0.5748
	GRU+$\boldsymbol{F}$	0.4673	0.8403	0.6006
	AttGRU	0.4696	0.9076	0.6189
	AttGRU+$\boldsymbol{F}$	0.5354	0.8908	0.6688

Table 6: The end-to-end results of precision (**P**), recall (**R**) and F1 score (**F**) of VPE resolution obtained with the train-test split used by Bos and Spenader (2011), Liu, Pellicer, and Gillick (2016) (**Liu et al. (2016)**) and (Kenyon-Dean, Cheung, and Precup 2016) (**MIRA**).† and ‡ denote the performance is statistical significant over the baselines of **DRVPE** and **MIRA**, respectively, with $p < 0.05$. The results in bold indicate the best performance.

each neural model, there are 6 experimental settings for VPE resolution. Accuracy, precision (**P**), recall (**R**) and F1 score (**F**) are used for evaluation. Table 4 shows the VPE resolution results on accuracy obtained with 5-fold cross validation. We can see that the neural models outperform the baselines significantly. It also reveals that the features ***F*** can further improve the performance of both MLP and Transformer models.

Similar to VPE detection, we also compare the precision (**P**), recall (**R**) and F1 score (**F**) of VPE resolution in the train-test data split used by Bos and Spenader (2011), Liu, Pellicer, and Gillick (2016) (Liu et al. (2016)) and (Kenyon-Dean, Cheung, and Precup 2016) (ML) for comparisons. The experimental results are shown in Table 6. We find that MLP outperforms the baselines significantly in three different functions, namely **Sum Pooling**, **GRU** and **Attention-based GRU**. Transformer also outperforms the baselines in F score. It again verifies that the features ***F*** can further improve the performance of both MLP and Transformer models for end-to-end VPE resolution.

	Trigger	Non-Trigger
VPE	0	1
Non-VPE	2	3

Table 7: Labels for the annotation of extended data.

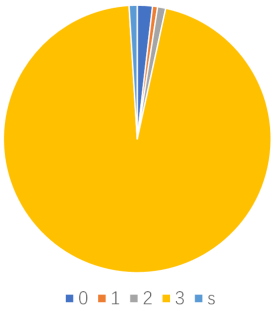

Figure 4: Proportion of the extended data.

Sentence	Label
The government [includes money spent on residential renovation]$_{antecedent}$; Dodge [does]$_{trigger}$n't.	0
Steam [may be changed]$_{antecedent}$ into water and water into ice.	1
She has a dog and I also [have]$_{trigger}$ too.	2

Table 8: Examples of extended data.

Data Extension for VPE

Recent VPE resolution models are mainly working on a benchmark dataset (Bos and Spenader 2011) in which each sentence has an antecedent and a trigger. However, considering to explore more VPE phenomena, we further extended the data in four conditions as shown in Table 7, where 0, 1, 2, 3 indicate the sentence that has a trigger and VPE, no trigger but has a VPE, has a trigger but no VPE, no trigger and VPE, respectively. The proportion of the extended data is shown in Figure 4, where the number of sentences that are labeled in 0, 1, 2, 3, s are 15, 5, 8, 784, 8, respectively. The total number of the extended data equals 820. The corresponding examples are shown in Table 8.

In addition to the four conditions, we observed a phenomenon where the antecedent is not a continuous sequence. We treat this as a special case of VPE and annotate it with a

label "s". For example, in the sentence "*Since the reforms went in place, for example, no state has posted a higher rate of improvement on the Scholastic Aptitude Test than South Carolina, although the state still posts the lowest average score of the about 21 states who use the as the primary college entrance examination.*" The antecedent is "*posted a rate of improvement*". This is due to the comparative form of the sentence. We will explore all the VPE phenomena in the extended data in future work.

Conclusion

We investigated neural network models for VPE resolution, proposing a novel framework for end-to-end processing of VPE detection and VPE resolution. Results show that the neural models outperforms the baselines on both VPE detection and the VPE resolution and the end-to-end process gives higher results than the baselines. In addition, traditional manual features are still useful for improving neural models. We release an extended dataset for VPE detection and resolution.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their insightful comments. We would like to thank Ms. Xi Chen and Mr. Caihai Zhu for their help on the annotation of the extended data. The paper is supported by the NSFC (No. 61502120, 61472105, 61772153).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bakhshandeh, O.; Wellwood, A. C.; and Allen, J. 2016. Learning to jointly predict ellipsis and comparison structures. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 62–74.

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

Bos, J. 2005. Verb phrase ellipsis and sloppy identity: a corpus-based investigation. *Mining for Parsing Failures*.

Bos, J. 2012. Robust vp ellipsis resolution in dr theory. *From quantification to conversation* 19:145–159.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. ACM.

Hardt, D. 1992. An algorithm for vp ellipsis. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, 9–14. Association for Computational Linguistics.

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

Hardt, D. 1998. Improving ellipsis resolution with transformation-based learning. In *Aaai fall symposium*, volume 1998.

Hobbs, J. R., and Kehler, A. 1997. A theory of parallelism and the case of vp ellipsis. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 394–401. Association for Computational Linguistics.

Kamp, H.; Reyle, U.; et al. 1993. From discourse to logic: Introduction to model-theoretic semantics of natural language, formal logic and discourse representation theory, volume 42 of. *Studies in linguistics and philosophy*.

Kawai, M. 2013. On vp ellipsis and the identity condition. *Proceedings of the 2013 annual conference of the Canadian Linguistic Association.*

Kenyon-Dean, K.; Cheung, J. C. K.; and Precup, D. 2016. Verb phrase ellipsis resolution using discriminative and margin-infused algorithms. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1734–1743.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liu, Z.; Pellicer, E. G.; and Gillick, D. 2016. Exploring the steps of verb phrase ellipsis. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, 32–40.

McShane, M., and Babkin, P. 2016. Detection and resolution of verb phrase ellipsis. *LiLT (Linguistic Issues in Language Technology)* 13.

Nielsen, L. A. 2003a. A corpus-based study of verb phrase ellipsis. In *Proceedings of the 6th Annual cluk Research Colloquium*, 109–115.

Nielsen, L. A. 2003b. Using machine learning techniques for vpe detection. In *Proceedings of RANLP*, 339–346.

Nielsen, L. A. 2004a. Robust vpe detection using automatically parsed text. In *Proceedings of the ACL 2004 workshop on Student research*, 25. Association for Computational Linguistics.

Nielsen, L. A. 2004b. Verb phrase ellipsis detection using automatically parsed text. In *Proceedings of the 20th international conference on Computational Linguistics*, 1093. Association for Computational Linguistics.

Nielsen, L. A. 2005. *A corpus-based study of verb phrase ellipsis identification and resolution*. Ph.D. Dissertation, King's College London.

Van Craenenbroeck, J. 2017. Vp-ellipsis. *The Wiley Blackwell Companion to Syntax, Second Edition* 1–35.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Vaswani, A.; Bengio, S.; Brevdo, E.; Chollet, F.; Gomez, A. N.; Gouws, S.; Jones, L.; Kaiser, Ł.; Kalchbrenner, N.; Parmar, N.; et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.