Exploring Statistical and Neural Models for Noun Ellipsis Detection and Resolution in English

by

Payal Khullar, Manish Shrivastava

Report No: IIIT/TR/2020/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2020
Exploring Statistical and Neural Models for Noun Ellipsis Detection and Resolution in English

Payal Khullar
International Institute of Information Technology Hyderabad
Gachibowli, Hyderabad
payal.khullar@research.iiit.ac.in

Abstract
Computational approaches to noun ellipsis resolution have been sparse, with only a naive rule-based approach that uses syntactic feature constraints for marking noun ellipsis licensors and selecting their antecedents. In this paper, we further the ellipsis research by exploring several statistical and neural models for both the subtasks involved in the ellipsis resolution process and addressing the representation and contribution of manual features proposed in previous research. Using the best performing models, we build an end-to-end supervised Machine Learning (ML) framework for this task that improves the existing F1 score by 16.55% for the detection and 14.97% for the resolution subtask. Our experiments demonstrate robust scores through pretrained BERT (Bidirectional Encoder Representations from Transformers) embeddings for word representation, and more so the importance of manual features—once again highlighting the syntactic and semantic characteristics of the ellipsis phenomenon. For the classification decision, we notice that a simple Multilayer Perceptron (MLP) works well for the detection of ellipsis; however, Recurrent Neural Networks (RNN) are a better choice for the much harder resolution step.

1 Introduction
Noun ellipsis is a linguistic phenomenon where the head noun of a noun phrase gets deleted, without making the sentence ungrammatical. For example, in the sentence in (1) from (Lobeck, 1995), the noun presentation is elided at "[e]".

1. John’s presentation on urban development was virtually ignored because [NP Mary’s [e]] was so much more interesting.

The elided information can be retrieved from the previous context as in (1) or with the knowledge of idiomatic usage of language as in I will be back in two [e]. It is also possible that the reference of the elided information comes from extra-linguistic, situational context. For example, consider a speaker pointing towards the roses in a shop and saying an utterance as in I will take two [e]. While human interlocutors resolve any such elided information by disambiguating from context, cognitive commonsense extension and reasoning (Chen, 2016), ellipsis resolution can be a hard task for Natural Language Processing (NLP) systems (Hardt, 1999). Resolution of ellipsis comprises two tasks - detection of the elided material and antecedent selection (Liu et al., 2016b; Nielsen, 2003). Ellipses occur in the environment of certain syntactical structures or trigger words, also known as licensors or triggers of ellipses. They are useful syntactic cues for the detection of noun ellipsis. See Figure 1 for an example of the noun ellipsis resolution process.

2 Related Work
Nominal ellipsis has been a topic of interest in theoretical linguistics for a very long time (Halliday and Hasan, 1976; Dalrymple et al., 1991; Lobeck, 1995; Lappin, 1996; Hobbs and Kehler, 1997; Hardt, 1999; Johnson, 2001; Wijnen et al., 2003; Merchant, 2004; Frazier, 2008; Chung et al., 2010; Mer-
The corpus marks a total of 946 annotations, of which 438 are described as endophoric, i.e. with a textual antecedent, and 508 exophoric, i.e. without a textual antecedent.

3.1 Noun Ellipsis Detection

From a given sentence, we first select all words belonging to the syntactic categories that can license noun ellipsis in English, i.e. cardinal and ordinal numbers, determiners and adjectives (Ross, 1967; Lobeck, 1995; Mitkov, 1999; Saito et al., 2008; Kim et al., 2019; Khullar et al., 2019) using a POS tag filter. The POS tags are obtained from state-of-the-art spaCy parser (Honnibal and Johnson, 2015). For simplicity, we refer to words with these categories as noun modifiers (although in strict linguistic terms, this might be problematic). For each of these selected noun modifiers, we follow the task specification for VPE detection used by (Nielsen, 2004a; Bos and Spenader, 2011; Liu et al., 2016a; Dean et al., 2016), Multilayer Perceptrons (MLP) and Transformers (Zhang et al., 2019). In recent times, there has been a surge in the computational research on nominal ellipsis and closely related phenomena (Khullar et al., 2020, 2019; Lapshinova-Koltunski et al., 2018; Menzel, 2017; Menzel and Lapshinova-Koltunski, 2014). For the resolution process, we previously proposed a rule based system (Khullar et al., 2019) that detects noun ellipsis using syntactic constraints on licensors of ellipsis and resolves them by matching Part-of-Speech (POS) tag similarity between the licensor of ellipsis and the modifier of the antecedent. It later fine tunes these syntactic rules on a small curated dataset that contains 234 instances of noun ellipsis along with some negative samples (Khullar et al., 2019). For the present paper, we further the research on noun ellipses by using the NoEI corpus annotated by us previously (Khullar et al., 2020) to experiment with state-of-the-art ML models.

3 The Proposed Approach

Following the VPE resolution framework presented by (Zhang et al., 2019), we investigate a similar framework for noun ellipsis resolution in English and present alternative choices of the models at each step as shown in Figure 2. We use the NoEI corpus (Khullar et al., 2020) that marks noun ellipsis instances as a separate layer (using the stand-off annotation scheme) on the Cornell Movie Dialogs corpus (Danescu-Niculescu-Mizil and Lee, 2011).
simple MLP and a biLSTM (Bidirectional Long Short Term Memory) as our state-of-the-art neural classifiers.

**BERT** We separate the sentence and the licensor with a [SEP] token and keep the sequence length to 300 as this is the maximum sentence length in the training data. After creating the concatenated set of tokens, if the number of tokens is greater than 300, we clip it to 300, otherwise we add [PAD] tokens which correspond to the embedding of 768 dimensional zero-vector. The [CLS] output of the BERT model (Devlin et al., 2019) is then fed into Naive Bayes, Linear SVM, MLP and bi-LSTM networks as above.

**Manual Syntactic Features** For each of these models, we additionally experiment with manual syntactic features. We use the lexical features proposed by (Dean et al., 2016) and extended lexical features by (Zhang et al., 2019), and take the five syntactic constraints on licensors of ellipsis explored by (Khullar et al., 2019) for their rule-based approach as our slot pattern features. We concatenate all these features to the embeddings from the previous step and check if they improve the classification decision.

### 3.1 Noun Ellipsis Detection

We define noun ellipsis detection as a binary classification task where given a licensor, antecedent candidate and their context, the goal of the classifier is to predict whether the antecedent candidate is the resolution of the ellipsis licensed by the licensor. Formally, given a sentence $s$, the licensor $l_i$ from the detection step, and the antecedent candidate $a_j$, the noun ellipsis resolution task can be defined as follows:

$$f(a_j, l_i, s) \rightarrow \{0, 1\}$$

where 1 denotes that the antecedent candidate $a_j$ is the actual resolution of the ellipsis licensed by $l_i$, and 0 otherwise.

**Embeddings** Similar to the detection step, we take pretrained FastText word embeddings for the licensor, antecedent candidate and context, and sum pool to obtain a single vector. In case of BERT, we separate the sentence, the licensor and the antecedent candidate with a [SEP] token and follow the same steps as in the detection step.

**Manual Syntactic and Semantic Features** We use POS tags of the licensor and modifier of the antecedent as our syntactic features and cosine similarity between their POS tags as our semantic features, following (Khullar et al., 2019). We concatenate these features to the embeddings to explore the efficacy of adding manual features on resolution.

### Experiments

We choose Naive Bayes and linear SVM as our statistical models, and Multilayer Perceptron (MLP) and bidirectional Long Short Term Memory (bi-LSTM) network as our neural models. For the
Table 1: Precision, Recall and F1-Score values of different statistical and neural models on the noun ellipsis detection task. FT stands for fastText and +F denotes concatenation of manual features to the embeddings. Values in bold depict best performance.

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Classification</th>
<th>Representation</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>Naive Bayes</td>
<td>FT</td>
<td>0.4192</td>
<td>0.5807</td>
<td>0.4869</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.4923</td>
<td>0.6275</td>
<td>0.5517</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.5511</td>
<td>0.6203</td>
<td>0.5837</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.5990</td>
<td>0.6213</td>
<td>0.6099</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>FT</td>
<td>0.4502</td>
<td>0.5560</td>
<td>0.4975</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.4824</td>
<td>0.6400</td>
<td>0.5501</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.5766</td>
<td>0.6503</td>
<td>0.6112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.5799</td>
<td>0.6771</td>
<td>0.6247</td>
</tr>
<tr>
<td></td>
<td>MLP</td>
<td>FT</td>
<td>0.6504</td>
<td>0.6974</td>
<td>0.6731</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.7123</td>
<td>0.8977</td>
<td>0.7943</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.6999</td>
<td>0.7398</td>
<td>0.7193</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.9116</td>
<td>0.8901</td>
<td>0.9007</td>
</tr>
<tr>
<td></td>
<td>bi-LSTM</td>
<td>FT</td>
<td>0.6010</td>
<td>0.6531</td>
<td>0.6260</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.6901</td>
<td>0.8745</td>
<td>0.7714</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.7001</td>
<td>0.7503</td>
<td>0.7243</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.8655</td>
<td>0.8163</td>
<td>0.8402</td>
</tr>
</tbody>
</table>

Detection task, we take the annotated 946 positive samples (exophoric) and randomly choose 946 negative samples. Similarly, for the resolution task, we take 438 positive samples (endophoric) and 438 randomly chosen negative samples. We perform a standard 70-10-20 split to obtain the train, development and test set respectively, and follow the 5-fold cross validation procedure to capture both classes properly in each case. For MLP, we take a simple, two-layer feedforward network (FFNN) or two layers of multiple computational units interconnected in a feed-forward way without loops. We have a single hidden layer with 768 neurons and a sigmoid function. A unidirectional weight connection exists between the two successive layers. The classification decision is made by turning the input vector representations of a word with its context into a score. The network has a softmax output layer. For bi-LSTM, we have embedding layer, time-distributed translate layer, Bi-LSTM (RNN) layer, batch normalization layer, dropout layer and prediction layer. The activation used is Softmax. The loss function is calculated with cross entropy. We train in batch sizes of 16 and early stopping with max epochs of 100. In early stopping the patience is kept to be 10 and the optimizer used is Adam. We use default values for the learning rate. We use Keras (Chollet et al., 2015) for coding the models.

5 Results

We evaluate the performance of our models in terms of F1-score, computed by taking an average F1-scores obtained from the 5-folds results. We experiment with sixteen models each for the noun ellipsis detection and resolution. The results on the testset for Precision, Recall and F1-Score values are presented in Table.2. As expected, the neural models perform significantly better than the statistical ones for both the subtasks. Our experiments show that for the detection task, BERT embeddings with a simple MLP gives best scores. This is expected because, BERT currently provides the most powerful contextual word representations, using 12 separate attention mechanism for each layer, where, at each layer, each token can focus on 12 distinct aspects of other tokens. Since Transformers (Vaswani et al., 2017) use many distinct attention heads (12*12=144 for the base BERT model), each head can focus on a different kind of constituent combinations, making BERT broadly attending over the whole sentence. In our task, the
Table 2: Precision, Recall and F1-Score values of different statistical and neural models on the noun ellipsis resolution subtask. FT stands for fastText and +F denotes concatenation of manual features to the embeddings.

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Classification</th>
<th>Representation</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Naive Bayes</td>
<td>FT</td>
<td>0.2992</td>
<td>0.5112</td>
<td>0.3775</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.3379</td>
<td>0.5750</td>
<td>0.4257</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.3919</td>
<td>0.5503</td>
<td>0.4578</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.4558</td>
<td>0.6230</td>
<td>0.5264</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>FT</td>
<td>0.4001</td>
<td>0.4960</td>
<td>0.4429</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.5024</td>
<td>0.6040</td>
<td>0.5485</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.4966</td>
<td>0.6003</td>
<td>0.5435</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.5372</td>
<td>0.6611</td>
<td>0.5927</td>
</tr>
<tr>
<td></td>
<td>MLP</td>
<td>FT</td>
<td>0.5901</td>
<td>0.6577</td>
<td>0.6221</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.7005</td>
<td>0.8122</td>
<td>0.7522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.6951</td>
<td>0.7798</td>
<td>0.7350</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td>0.8015</td>
<td>0.8531</td>
<td>0.8265</td>
</tr>
<tr>
<td></td>
<td>bi-LSTM</td>
<td>FT</td>
<td>0.6009</td>
<td>0.6321</td>
<td>0.6161</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT+F</td>
<td>0.7029</td>
<td>0.8044</td>
<td>0.7502</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT</td>
<td>0.7195</td>
<td>0.7383</td>
<td>0.7288</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BERT+F</td>
<td><strong>0.8652</strong></td>
<td><strong>0.8166</strong></td>
<td><strong>0.8401</strong></td>
</tr>
</tbody>
</table>

BERT with MLP model is robust and efficiently makes generalisations on the syntactic and semantic dependency between the elided noun and the pre-modifiers and modifiers (licensors). For the resolution task, however, bi-LSTMs work better than MLP. Unlike MLPs, Bi-LSTMs take into consideration left to right and right to left context, capturing long range dependencies in a sentence and, hence, are better suited for handling the resolution of a cohesive discourse device like ellipsis where the distance between elided phrase and antecedent can be several words. The sufficient neurons in the hidden layer with sigmoidal function ensures the MLP approximate the nonlinear relationships between the input and output, but they are not innately designed to capture temporal relationships within a sentence. Hence, although they perform well for a task like detection that needs local information, they are outperformed by bi-LSTMs on the resolution task that requires capturing a deeper relationship between the antecedent and the elided noun. We also note that manual feature addition boosts results greatly for all models, highlighting that ellipsis is a syntactically constrained phenomenon. We finally integrate the best models for each subtask into an end-to-end pipeline, as in Figure 2. Now, instead of the gold vectors (from the annotations), the resolution model is fed the output licensor vector from the detection model. This obviously results into error propagation into the second model, and lowers the precision value to 82.52%, recall to 78.66% and consequently, the F1-score to 80.55% of the final system. The error in the final system comes from failing to detect actual licensors, wrongly identifying non-licensor words and correct licensor detection but failed antecedent resolution. We run our final system on the curated dataset prepared by (Khullar et al., 2019) and compare the results with their rule-based approach. As expected, this model improves the F1-score by 16.55% for noun ellipsis detection and 14.97% for noun ellipsis res-

Table 3: Precision (P), Recall (R) and F1-Score (F) values (%) of the rule-based approach by (Khullar et al., 2019) and the neural model presented in this paper.
olution. See Table 3. The even higher accuracy on the curated dataset can be explained by the nature of the sentences in this dataset which are from textbooks, and, hence, free of grammatical errors, etc. – resulting into improved parser performance in the pre-processing step. Although, the presented models achieve high scores on both the tasks separately and in the pipeline process, the results can be further improved with hyper-parameter tuning and additional regularization.

6 Conclusion

We explored statistical and neural models for noun ellipsis detection and resolution, presenting a strong results for this task. As expected, neural classifiers perform significantly better than the statistical with the same input representation. As with several other NLP tasks, the contextual nature of BERT is useful for noun ellipsis resolution too, making robust predictions with simple neural classifiers. Finally, addition of manual features boosts the performance of all classifiers including those that use BERT, highlighting that ellipsis is a syntactically constrained phenomenon.

References


François Chollet et al. 2015. Keras. https://keras.io.


Tilbe Goksun, Tom W. Roeper, Kathy Hirsh-Pasek, and Roberta Michnick Golinkoff. 2010. From noun-phrase ellipsis to verbphrase ellipsis: The acquisition path from context to abstract reconstruction.


Jason Merchant. 2010. Three Kinds of Ellipsis: Syntactic, Semantic, Pragmatic?


Frank Wijnen, Tom W. Roeper, and Hiske van der Meulen. 2003. Discourse binding: Does it begin with nominal ellipsis?