Supplemental Material:
Machine Coding of Policy Texts with the Institutional Grammar

September 7, 2020
Contents

1 Model Features 2
2 Model Estimation Diagnostics 3
3 XGBoost 4
1 Model Features

Our application of CoreNLP to policy documents yields four nominal / categorical variables that we employ for our analysis. For purposes of analysis, we convert each of the categories into indicator variables. Given the number of categories in particular for word, the result is a high-dimensional dataframe and rich feature set. Below, we offer a brief description of the four primary variables used to generate the features, and examples of values the variables can take on.

- **word**: the identified token.
  - *Examples*: "there", "is", "hereby", "established"

- **pos**: the part-of-speech.
  - *Examples*: "DET", "VERB", "ADV", "VERB"

- **word source**: the source term in the sentence for the dependency relation.
  - *Examples*: "is", "established", "is", "board"

- **relation**: relational dependency.
  - *Examples*: "expl", "root", "advmad", "xcomp"
2 Model Estimation Diagnostics

Figure 1: *Evaluation of Trained Three Layer Neural Network Model of Institutional Grammar.* This figure plots the loss metrics (top panel) and accuracy (bottom panel) for the trained models both within the training sample (blue) and within a held-out evaluation set (green) across the 10 training epochs.
3 XGBoost

In addition to evaluating the predictive capacity of the trained neural network, we also leverage the predictions in what might be termed a “meta-model.” Here, we leverage the predicted classification probabilities from the three layer neural network in a second stage classifier. Specifically, we add the classification probabilities for each of the six categories to our training dataset, and estimate a gradient boosting (Friedman et al., 2000; Friedman 2001) model using extreme gradient tree boosting, also known as XGBoost (Chen and Guestrin, 2016); using XGBoost in combination with neural networks has been regularly employed in winning data science challenge solutions (Chen 2016).

The underlying motivation for the second stage is two-fold. First, and pragmatically, the first stage might have a tendency to assign a high overall probability to the most commonly assigned IG component. This maximizes the predictive accuracy at this stage, but has the downside of ignoring the small increases in probability associated with particular sub-categories of components that may be substantively meaningful in classification. The second stage, however, can incorporate that additional signal and results in a substantial increase in predictive accuracy. Second, and more theoretically valuable as we consider avenues for development, we envision this stage as later incorporating the probabilities for context terms; that is, those terms which immediately follow or precede the word we are attempting to classify. In doing so, the classifier will naturally incorporate a type of smoothing that will yet further increase both its accuracy and its utility for practical applications.1 The results of the XGBoost model appear in Table 1. Interestingly, the boosting approach yields only a very slight improvement in accuracy overall.

---

1We train an XGBoost model using the xgboost package in R. We train a model with a softmax objective to be evaluated by a multiclass log-loss function, where the max depth of the trees is 20, we complete 30 training passes on the dataset, the step size for each boosting step is 0.1, and a regularization parameter on the weight of 0.08.
<table>
<thead>
<tr>
<th>Level</th>
<th>Accuracy</th>
<th>MCC</th>
<th>Precision</th>
<th>Recall</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>0.75</td>
<td>0.62</td>
<td>-</td>
<td>-</td>
<td>922</td>
</tr>
<tr>
<td>Aim</td>
<td>-</td>
<td>-</td>
<td>0.88</td>
<td>0.87</td>
<td>69</td>
</tr>
<tr>
<td>Attribute</td>
<td>-</td>
<td>-</td>
<td>0.65</td>
<td>0.61</td>
<td>81</td>
</tr>
<tr>
<td>Condition</td>
<td>-</td>
<td>-</td>
<td>0.71</td>
<td>0.68</td>
<td>328</td>
</tr>
<tr>
<td>Deontic</td>
<td>-</td>
<td>-</td>
<td>0.95</td>
<td>1.00</td>
<td>41</td>
</tr>
<tr>
<td>Object</td>
<td>-</td>
<td>-</td>
<td>0.75</td>
<td>0.80</td>
<td>394</td>
</tr>
<tr>
<td>Or/Else</td>
<td>-</td>
<td>-</td>
<td>0.67</td>
<td>0.22</td>
<td>9</td>
</tr>
</tbody>
</table>

Note: Results based on classification of held-out test set using neural network classifier trained on a set of 8,120 randomly sampled words.

Table 1: Out-of-sample Performance of XGBoost Classifier for Predicting Institutional Grammar Components from Nineteen Policy Documents.