Progress:

* Data exploration and serialization: parse the data to a format that is more conducive to making input samples for different prediction settings (predict predicate attributes, argument attributes, edge attributes). Each sentence is stored from the predicates’ “perspective”: each sentence has a number of predicates, each of which has the relevant text span, the predicate’s attributes, the connected arguments’ attributes, and these edges’ attributes (attached are code for parsing and a sample subset of the data).
* Model outline:
  + The model is designed for multiple prediction settings in mind
  + Due to nested and overlapping predicate-argument spans, the model cannot predict all attributes at once
  + Using the approach from Question Answering and special token, we can specify what attribute set or for which predicate span we want to predict
  + The idea is that the model will have an edge prediction mode (for predicting edge attributes) and a node prediction mode (for predicting predicate and argument attributes)
  + Question Answering: we can ‘prime’ the model for different predictions by appending a question (or some special tokens) at the beginning of the input and place a special token before the predicate head. The predictions will be tied to the tokens
    - Example: To predict edge attributes headed by specific predicate, the span text ‘From the AP comes this story’ will become ‘What are the edge attributes? From the AP [predicate] comes this story’, or for node attribute ‘What are the node attributes? From the AP [predicate] comes this story’.
    - To get the full set of semantics attributes, we have to run a sample through the model 2 (edge and node) x number of predicate heads
  + The advantage of this approach is we can predict all the semantics attributes without changing the model architecture since the differences is already reflected on the input level. This makes it easy to switch between different prediction settings, i.e., takes out the relevant samples from the training set.
  + The disadvantages include increased inference runtime (which might not matter since we are not concerned with fully predict a sample, rather see the difference in performance when the model is transferred between different sample subsets) and increased number of samples needed to train (debatable whether this is a disadvantage).
  + For model details, and model architecture that can perform seq2seq prediction can be used, some noteworthy architectures include Bi-LSTM and unmasked transformer based models (based on observing previous researches that mostly used bidirectional encoders)