JRL336, QW79
CS 4740 Project 3

Proposal

We plan on implementing HMM for NER. For preprocessing, we will use standard python library functions to iterate through the training text file. We will parse the lines into appropriate data structures to capture all the information relevant to a word token while maintaining ordering.

In this case, the hidden variables are the NER tags (“O”, “B-PER”, “I-PER”, “B-LOC”, etc.). They need to be found out using HMM during testing for the test data.
The observed variables are the lexical word tokens and part-of-speech tags. They will be used in the HMM for training.

To be more explicit, some examples of the model parameters are:
states = ('PER', 'LOC', ‘ORG’, ‘MISC’)
observations = ('american', 'played', ‘baseball', ‘barcelona’)
start_probability = {'PER': 0.2, 'LOC': 0.2, ‘ORG’: 0.3, ‘MISC’: 0.3}
transition_probability = {
  'PER' : {'PER': 0.4, 'LOC': 0.2, ‘ORG’: 0.2, ‘MISC’: 0.2},
  'LOC' : {'LOC': 0.5, 'PER': 0.2, ‘ORG’: 0.2, ‘MISC’: 0.2},
…
  }
emission_probability = {
  'PER' : {'american': 0.7, 'played': 0.1, 'basebal': 0, ‘barcelona’: 0.2, … },
  'LOC' : {'american': 0.7, 'played': 0, 'basebal': 0, ‘barcelona’: 0.3, … },
…
  }

We are planning on experimenting with bigram and trigram based features and possibly MEMM for our extension.

We submitted our baseline system on Kaggle under the name kungFuPandas_hmm. We obtained an accuracy of 40.7%. Our baseline system is rather rudimentary. We built a lexicon of all the named entities in the training data and during testing only those named entities that are part of the lexicon. We didn’t deal with unknown words in the baseline system. In the actual implementation of HMM, nevertheless, we will use smoothing to take into account unknown words.