CS 6320

Project 1: Part-of-speech Tagging

Hidden Markov Model Tagger using Viterbi algorithm

The HMM Model is based on the assumption that current state(tag) depends only on the previous state(tag). The Viterbi algorithm uses a dynamic programming approach to construct the most probable state sequence. This was used to give the most probable POS tag sequence based on the probabilities computed for the given corpus. Due to its assumption, it may not always give the required sequence for long inputs. For ex:

'the secretariat is expected to race tomorrow .'

For the above input, this is the output given:

['DETERMINER', 'NOUN', 'VERB', 'VERB', 'PREPOSITION', 'NOUN', 'NOUN', 'PUNCT']

The pos tag for ’race’ is given incorrectly as ‘NOUN’ instead of ‘VERB’.

Recurrent Neural Networks

To predict the POS tag of a token, the RNN model also takes in some data that gives the context of the token in the sentence.

For the input

'the secretariat is expected to race tomorrow .',

The RNN tagger gives the output as

['DETERMINER', 'NOUN', 'VERB', 'VERB', 'X', 'VERB', 'NOUN', 'PUNCT']

The pos tag for ’race’ is given correctly. This can be attributed to the high size of the embedding vector and large number of LSTM units. However tag for ‘to’ is given as ‘X’. This is most probably due to training data declaring ‘to’ as ‘X’ tag in some cases.

Learning:

1. Use of keras library
2. There is still scope for improvement as I haven’t taken care of inputs having unknown words
3. Many steps such as padding, one-hot encoding need to be performed alongside creating the model.
4. RNNs can be used for other use cases like sentiment analysis or summary.