Anjan Khatri

IT 320

Dr. Nathan Green

Lab 4 pos.py

#1) Why is the training accuracy higher than the testing accuracy?  
 Training accuracy is higher than testing accuracy because training accuracy takes 90% of data to train itself and testing on remaining 10%. As training accuracy are on some data which are different from the testing where the whole data is split to 90% training and 10% testing.

#2) Why is the training accuracy not perfect (100%)?

Training accuracy is not perfect because we are training a tagger on some data which are different from test data. A tagger that simply memorize its training data and made no attempt to construct a general model that would get the perfect score instead it would be useless for tagging new text. We split the data on 90% training and 10% testing. So, training accuracy is not perfect.

#3) Why does the accuracy score on the training data not go up but it does on the test data?

On second part to of the code, backoff used the default tagger. Backoff technique give simple context in which to present them. If unigram tagger is unable to find a tag, then default tagger was used when a tagger cannot assign a tag in the given context then backoff to more general model.   
  
#4) Create two new taggers, A BigramTagger that has not backoff and a BigramTagger that user a unigram tagger as backoff. Report the accuracies. Why is one so much lower than the other?

After using bigram tagger. One is so much lower than the other because bigram tagger take 2 gram and as bigram tagger manages to tag every word in the sentence, it will badly on an unseen sentence and as soon as it encounters a new word it is unable to assign a tag. It cannot a tag a word even if it was seen during training but it never saw it with the same two words which results to fail to tag the rest of the sentence so it is lower than the others.  
   
#5) Repeat #4 with a TrigramTagger using a Bigramtagger as backoff

In lab4 python file.