CSCI 5832: Homework 4
Named Entity Recognition

- By Payoj Jain

Overview

The aim of the assignment is to implement a learning based approach to named entity recognition. In this approach, we can cast the problem of finding named entities as a sequence labeling task using IOB tags. The particular NER task we'll be tackling is to find all the references to genes in a set of biomedical journal article abstracts.
The model has been trained using IOB-tagged data from a set of biomedical journal article abstracts with over 13000 sentences.
Below is the list of all the assumptions that have been taken into account while developing the model:

- New words may or may not occur in the test set.
- No new IOB tags will appear in the test set.
- While developing the model I have split the provided training data into *train set (90% of the provided training data) and development (dev) set (remaining 10%).* Also, I have randomized the training data while before splitting it into train set and dev set as lot sentences in the training data were repeated and to ensure that all possible type of sentences could be included in the training set

HMM and Viterbi

To implement Viterbi there were certain calculations and implementations that needed to be done. These were:

1) ***Extract the required counts from the training data to generate the required probability estimates for the model.***

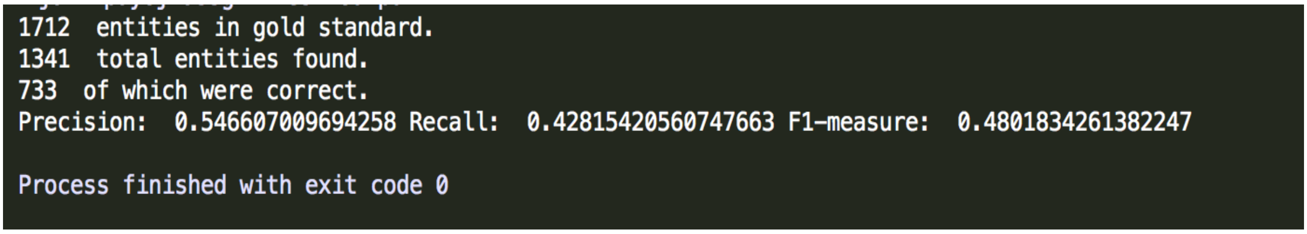
- List of all IOB tags, their total unigram counts, their bigram counts, count a particular tag appeared at the end of a bigram type, count a particular tag appeared at the beginning of a bigram type, count of a particular tag starting a sentence, count of a particular tag ending a sentence, probabilities corresponding to these counts which are useful for smoothing, calculating emission probabilities of words and transmission probabilities of tags.
- List of words, their total counts, tags which they have been assigned in the train set, count of tags a particular word has been tagged to.
- List of all the bigram types and their total counts appeared in the train set.

2) Deal with unknown words: In the train set, words which rarely appear (less than or equal to cutoff) are considered as "UNKNOWN WORDS". Any new/unseen word appearing in test/dev set will be given emission probability based on the count of "UNKNOWN WORDS" from train set.

3) Do some form of smoothing for emission probability.

Laplace smoothing: I have also handled the probability of a tag starting or ending a sentence by Laplace smoothing. Since, in the train set almost all sentences were ended by '.' but there is a possibility that some other tag may end the sentence in dev/test set. Also, there were many tags which were never at the start of a sentence in the train set but this doesn't mean that the same tag won't appear at the beginning of a sentence in dev/test set.

Implementing Viterbi decoding and backtracking the path on Viterbi matrix to get the most probable tags for the given set of test/dev data shows result. evaluation of these kinds of systems is not based on per tag accuracy. Performance is judged by optimizing precision and recall. Thus, the F1 measure which is the harmonic mean of precision and recall is the right metrics. Performance of HMM model for IOB tagging is 48.0183% (F1 measure).



```
1712  entities in gold standard.
1341  total entities found.
733  of which were correct.
Precision:  0.546607009694258 Recall:  0.42815420560747663 F1-measure:  0.4801834261382247

Process finished with exit code 0
```

Other methods tried

I have tried, RNN but the training set is not enough to get a good F1 measure (0.38) for IOB tagging. I have also tried to change my HMM by trying different smoothing methods for emission probability. Also, I tried to implement MEMM and CRFs models which should have worked better than HMMs but since there are not appropriate word embedding and features related to words which can help make model stronger. There are over 17000 words in the corpus with frequency 1 which makes task even more difficult to implement MEMMs.