The University of Memphis
Department of Computer Science
Dunn Hall 375, Memphis, TN 38152-3240

Project Report

"A JAVA Based HMM Tagger"

Submitted By:
Nabin Maharjan
[U00534753]

Due Date:
1st May 2015

A JAVA Based HMM Tagger:

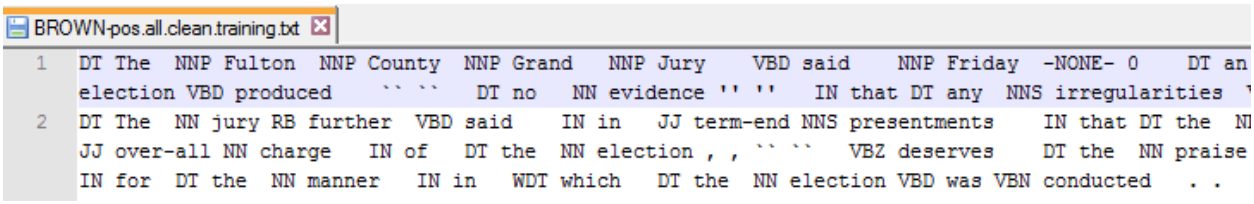
For a given sequence of observations, the HMM model has a sequence of hidden states that generates them. In HMM tagger, words represent observations and tags represent hidden states.

For our JAVA based HMM tagger, the Penn Tree Bank tag sets consisting of 45 tags were used as hidden states of the model.

Preprocessing:

The sentence parse trees in the Brown corpus were preprocessed to generate a preprocessed file containing sequences of sentences. Each sentence was written in a line in the following format.

pos1<SPACE>word1<TAB>pos2<SPACE>word2 (and so on)



This pre-processed file was then used for training and testing the HMM tagger. The words with 'NONE' tags were ignored. Each sentence in the above file was read and mapped to HMMData instance object.

A HMMData instance object consists of five different ArrayList objects to represent five different sequences as listed below:

1. Sequences of original words
2. Sequences of processed words
3. Sequences of coded words
4. Sequences of actual tag sequences
5. Sequences of predicted tag sequences

Each sentence in the preprocessed file was mapped to a sequence of original words and actual tags in a HMMData instance object.

Let's consider an example sentence:

```
DT The  NN couple   VBD was VBN married NNP Aug.    CD 2    , , CD 1913 . .
```

This sentence has a sequence of original words as *[The couple was married Aug. 2 , 1913 .]* and a sequence of actual tags as *[DT NN VBD VBN NNP CD , CD .]*

Not all words in all the sentences can be observed in the vocabulary. A special word "__unknown__" was used to represent all unseen words. So, a sequence of processed words for a sentence was represented as a sequence of original words where unseen words are replaced by special

"__unknown__" word. If word “couple” was an unseen word in the test sentence, the sequence of processed words would look like:

[The __unknown__ *was married Aug. 2 , 1913 .]*

The Brown Corpus data was used for building the HMM tagger application. The 60% of the data was used for building vocabulary consisting of 42752 unique words. These words might not be directly used as observation symbols to the HMM model as there are constraints to how many constants can be defined for a program depending upon the programming language. The JAVA allows maximum 65535 constants per program and implementing JAVA based HMM tagger would give **constants exceeded error** if words were used as observations. In order to overcome this issue, each word in the vocabulary was encoded by a unique integer and, these integers were used as observations for the HMM model. The sequence of coded words generated from the sequence of processed words for the above example might look like:

[20 115 *12 335 587 2360846 12223 24500 896]*

These sequences of coded words were used for training and testing the HMM tagger model. The predicted tags for the given sentence were outputted to a sequence of predicted tags in HMMData instance object of the sentence.

Building an initial HMM model:

A HMM model is represented by three parameters: initial state probabilities, transition probabilities and emission probabilities.

In order to build an initial HMM model, the Brown Corpus data (60% data) was used to compute all these HMM parameters.

For computing initial state probabilities:

$$\prod(\mathrm{i}) = \frac{\mathrm{count(i)}\ at\ the\ beginning\ of\ sentence}{\sum_{i=1}^{N} count(i)\ at\ the\ beginning\ of\ sentence}\ \ where\ N\ is\ number\ of\ tags \quad (1)$$

For computing transition probabilities:

$$A(\mathrm{i,j}) = \frac{\mathrm{count(i,j)}}{\mathrm{count(i)}}\ \ where\ i,j\ are\ tags \qquad (2)$$

For computing emission probabilities:

$$Bi(\mathrm{o1}) = \frac{\mathrm{count(o1)}\ \ with\ tag\ i}{\sum_{i=1}^{V} count(oi)\ with\ tag\ i}\ \ where\ i,j\ are\ tags\ and\ oi\ are\ observations \qquad (3)$$

The above emission probabilities accounted for only seen words in the 60% of the corpus data. The emission probabilities of special "__unknown__" word by each tag is also required to be estimated. For this estimation, the remaining 40% data of corpus was used in which all the words not in found in vocabulary were replaced by "__unknown__" word and its emission probability for tag was estimated as:

For computing emission probabilities of **__unknown__** word observation:

$$Bi(\text{\_unknown\_}) = \frac{\text{count(\_unknown\_) } with\ tag\ i}{\sum_{i=1}^{V} count(oi)\ with\ tag\ i}\ where\ i,j\ are\ tags, oi\ are\ observations \quad (4)$$

The equation (3) is now readjusted to accommodate the emission probabilities of unknown word as:

$$Bi(\text{o1}) = \ Bi(\text{o1}) * (1 - Bi(\_unknown\_))where\ i\ are\ tags, oi\ \neq\ \_unknown\_ \quad (5)$$

Training HMM model

The initial HMM tagger model was then used for training using Baum-Welch learning algorithm. Training could not be done for more than 1 iteration as more than 1 iterations resulted in NaN probabilities. However, the initial HMM model alone was found to outperform lexical statistical tagger when tested on whole Brown Corpus data.

Experiments

Evaluation on Whole Brown Corpus

The HMM tagger was tested on the whole dataset. It predicted tags 95.62% accurately and outperformed lexical statistical tagger which reported 91.08% accuracy.

Table 1Comparision of Accuracy of taggers against Brown Corpus Data

S.N.	Tagger	Accuracy
1	Lexical Statistical Tagger	91.08
2	HMM tagger	95.62

Evaluation on Unseen Words

The vocabulary for HMM was built using only 60 percent of the Brown corpus data set. So, 17124 unseen words were observed when testing the model in the whole dataset. Below is an example sentence containing unseen words with HMM tagger prediction. The tagger predicted correct tags for unseen words **phonology** and **articulations**. Each word in the sentence is represented in format: <original word>/<processed word>/<expected tag>/<predicted tag>.

Of/Of/IN/IN course/course/NN/NN ,/,/,/, something/something/NN/NN of/of/IN/IN the/the/DT/DT same/same/JJ/JJ sort/sort/NN/NN occurs/occurs/VBZ/VBZ with/with/IN/IN other/other/JJ/JJ sectors/sectors/NNS/NNS of/of/IN/IN the/the/DT/DT **phonology/__unknown__/NN/NN** :/:/:/: consonantal/consonantal/JJ/JJ **articulations/__unknown__/NNS/NNS** have/have/VBP/VBP both/both/DT/PDT a/a/DT/DT linguistic/linguistic/JJ/JJ and/and/CC/CC an/an/DT/DT individual/individual/JJ/JJ component/component/NN/NN ././.

The HMM tagger reported 50.68% accuracy on unseen words.

Table 2Accuracy of Unseen Words in Brown Corpus

S.N.	Tagger	Accuracy
1	HMM Tagger	50.68

Conclusion

The tagger based on HMM model can be used to predict the tags of the word in a given text. The HMM tagger captures some context information as probability of seeing an observation at time also depends upon previous tag observed. As expected, the HMM tagger performed better than lexical statistical tagger.