ASSIGNMENT: To find POS tags of every word of a given sentence using hidden Markov model

We have implemented this using three techniques viz.

a. Simple Naive Bayes Algorithm

b. Variable Elimination (Forward-Backward algorithm)

c. Viterbi Algorithm.

a**. Simple Naive Bayes Algorithm**

Formula used:

ARGMAX P(POSTAG|WORD Wi) = P(WORD Wi|POSTAG) * P(POSTAG)

where P(POSTAG) is the prior probability for any pos tag

and P(WORD WI|POSTAG) is the emission probability for a word given POSTAG.

- For this implementation, we use simple bayes net, where every word is independent of other word.

b. **Variable Elimination**

In Variable elimination we take all other states into account and eliminate them by performing summation over all possible values.

Lets say We need to find P(POSTAG PTi| WORD Wi)

Step 1:

Compute tow_table(alpha) using forward algorithm from first word upto the desired word. i.e PT1...PTi

Step 2:

Compute tow_table(Beta) using backward algorithm from last letter upto the desired letter. i.e PTn....PTi

Step 3:

Finally we multiply alpha(PT1...i)*emission(PTi|Word Wi) * beta(PTi....n)

c**. Viterbi Algorithm**

Formula used:

Step 1:

In this we created a viterbi matrix (I*J) where i = number of words in sentence and j = number of POS tags.

vit(i,j) stores viterbi values and a backpointer to the previous state.

Formula :

vit(W0,j) = P(Word W0|POSTAG) * P(POSTAG)

vit(Wi,j) = max(vit(Wi-1,j) * P(POSTAG|Prev POSTAG)) * P(Word Wi|POSTAG)

Step 2:

After creating the viterbi matrix, we begin with the last letter and traverse along the path till the first letter using the backpointer maintained.

PROBABILITY CALCULATIONS:

1. Initial Probability:

P(postag PT) = Number of occurences of the postag PT in training file/ total Number of postags

Note: If No. of occurences of POSTAG PT is zero then P(Letter L1) = 10e-14

2. Transition Probability:

P(POSTAG P2|POSTAG P1) : No. of occurences where P2 succeeds P1 in training file/ No. of occurences of P1

Note: If No. of occurences of POSTAG P1 is zero then P(L2|L1) = 10e-14

3. Emission Probability:

P(Word Wi| POSTAG Pi) : No. of occurences where Word Wi has POSTAG Pi in training set/ No. of occurences of POSTAG Pi.

POSTERIOR PROBABILITY:

Calculate the log of the posterior probability of a given sentence with a given part-of-speech labeling

The posterior probability is the probability where we need to calculate the probability for each word in the sentence given the corresponding tag for that sentence.

That is posterior probability = P(S1......Sn|W1.......Wn).

Applying naive bayes and bayes law, we will get this probability equal to P(W1|S1)........P(Wn|Sn)*P(S1)P(S2/S2).......P(Sn|Sn-1)/P(W1...Wn)

(ignoring denominator)

For example: Sentence- The house is big

label- Det noun pron adj

In the above bayes net, we can calculate the posterior probability as follows:

P(The/Det)*P(Det)*P(house/noun)*P(noun)*P(noun/Det)*P(is/pron)*P(pron)*P(pron/noun)*P(big/adj)*P(adj)*P(adj/pron)

OBSERVATIONS

1. Viterbi & Variable elimination performs much better than simple.

2. Viterbi & VE give almost equal results.

3. We observed that for some really long sentences(like 519th sentence in bc.test set) the issue of underflow takes place.This leads into wrong predication.In order to handle this, we have performed scaling up by multiplying 10e5 while computing the tow table. This inturn increased the word accuracy from 94.48% to 95.32% in HMM VE.

RESULTS

Scored 2000 sentences with 29442 words.

	Words correct:	Sentences correct:
0. Ground truth:	100.00%	100.00%
1. Simplified:	93.92%	47.45%
2. HMM VE:	95.32%	56.05%
3. HMM MAP:	95.31%	55.30%