NLP Assignment-2

Akshala Bhatnagar
2018012

1. Markov Length = 1

	Precision	Recall	F1 Score
Fold 1	0.7164713389	0.7542372827	0.7533900533
Fold 2	0.6898825106	0.7550126580	0.7877468870
Fold 3	0.5892765658	0.7478900800	0.7314389159
Average	0.665210138	0.752380007	0.757525285

Brown_train.txt is read into a string. The whole text is divided into sentences. The start of sentence token is added in front of each sentence (<s>_<s>). Then the sentence is divided into words. Each word has two parts, the word token and the tag. These are separated by _. A list of sentences is made which contains a list of tuples that have the word and the tag. Only the main tags are taken, the prefixes and suffixes separated by - are removed. So there are 189 tags. The data is divided into train and test using a 3 fold method. For the train data, all unique tags and unique words (vocabulary) are found. An unknown token is added for unseen words in the test data. A dictionary is made

which contains the word as key and all possible tags this word can have as the values. For the unknown token, all possible tags are included. The count of all tags is calculated.

Transition matrix is made which contains transition probabilities. This is the number of occurrences of current tag given the previous tag divided by number of occurrences of previous tag. Emission matrix is calculated which contains the emission probabilities. This is the number of occurrences of the current word given the tag divided by number of occurrences of the tag. For the unknown token, probabilities are calculated on the basis of tag frequency.

Then the viterbi algorithm is called for the test set. For each word, the previous word is taken into consideration. For all possible tags of current and previous word, the product of transition and emission probabilities is taken. The current tag is assigned on the basis of the maximum value of this product.

Then the confusion matrix is made with column headings as the actual tag value and row headings as the predicted tag values. Using this precision, recall and F1 score is calculated for each tag and average is taken for the entire dataset. Precision is true positive divided by the total occurrences of the predicted tag. Recall is true positive divided by the total occurrences of the actual tag. F1 score is the harmonic mean of precision and recall. This is done for each of the 3 folds and then average is taken.

Markov Length = 2

	Precision	Recall	F1 Score
Fold 1	0.7328882924	0.7401858777	0.7481548035
Fold 2	0.7248014352	0.7339199465	0.7293321908
Fold 3	0.7323478604	0.7359661191	0.7341525316
Average	0.730012529	0.736690648	0.737213175

Brown_train.txt is read into a string. The whole text is divided into sentences. Two start of sentence tokens are added in front of each sentence (<s>_<s>). Then the sentence is divided into words. Each word has two parts, the word token and the tag. These are separated by _. A list of sentences is made which contains a list of tuples that have the word and the tag. Only the main tags are taken, the prefixes and suffixes separated by - are removed. So there are 189 tags.. The data is divided into train and test using a 3 fold method. For the train data, all unique tags and unique words (vocabulary) are found. An unknown token is added for unseen words in the test data. A dictionary is made which contains the word as key and all possible tags this word can have as the values. For the unknown token, all possible tags are included. The count of all tags is calculated.

Transition matrix is made which contains transition probabilities. This is the number of occurrences of current tag given the previous tag and

previous of previous tag divided by number of occurrences of previous tag and previous of previous tag. Emission matrix is calculated which contains the emission probabilities. This is the number of occurrences of the current word given the tag divided by number of occurrences of the tag. For the unknown token, probabilities are calculated on the basis of tag frequency.

Then the viterbi algorithm is called for the test set. For each word, the previous two words are taken into consideration. For all possible tags of current and previous two words, the product of transition and emission probabilities is taken. The current tag is assigned on the basis of the maximum value of this product.

Then the confusion matrix is made with column headings as the actual tag value and row headings as the predicted tag values. Using this precision, recall and F1 score is calculated for each tag and average is taken for the entire dataset. Precision is true positive divided by the total occurrences of the predicted tag. Recall is true positive divided by the total occurrences of the actual tag. F1 score is the harmonic mean of precision and recall. This is done for each of the 3 folds and then average is taken.

- The confusion matrices and stats are present in the respective folders named Markov_length_1 and Markov_length_2
- These folders further have the folders Fold1, Fold2, Fold3 and Average

2. Formulation of HMM, Markov length = 2

S : state

O : observation

Goal: Maximize $P(S|O)$ tag sequence by choosing best sequence

$$S^* = \text{argmax}_S P(S|O)$$

$$P(S|O) = P(\{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\} | \{O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8\}) \quad \text{①}$$

Observation	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
State	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8

Applying chain rule to ①

$$P(S|O) = P(S_1|O)\, P(S_2|S_1, O)\, P(S_3|S_2, S_1, O) \ldots P(S_8|S_7 \ldots S_2, S_1, O)$$

Applying Markov Assumption

↳ length 2

$$P(S|O) = P(S_1|O)\, P(S_2|S_1, O)\, P(S_3|S_2, S_1, O)\, P(S_4|S_3, S_2, O) \ldots P(S_8|S_7, S_6, O)$$

Applying Bayes Theorem

$$S^* = \text{argmax}_S P(S|O)$$
$$= \text{argmax}_S P(O|S).P(S)$$

Prior,

$$P(S) = P(S_1).P(S_2|S_1).P(S_3|S_2, S_1).P(S_4|S_3, S_2) \ldots P(S_8|S_7, S_6)$$

Likelihood,

$$P(O|S) = P(O_1|S).P(O_2|O_1, S)\, P(O_3|O_2, S)\, P(O_4|O_3, S) \ldots P(O_8|O_7, S)$$

~~observation depends on current state only~~

[observation depends on current state only]

$$P(O|S) = P(O_1|S_1)\, P(O_2|S_2) P(O_3|S_3) \ldots P(O_8|S_8)$$

$$P(S|O) = P(O|S)\,P(S)$$
$$= P(S_1)\,P(S_2|S_1)\,P(S_3|S_2,S_1)\,P(S_4|S_3,S_2)\ldots P(S_8|S_7,S_6)\cdot P(O_1|S_1)\,P(O_2|S_2)\,P(O_3|S_3)\ldots P(O_8|S_8)$$

Introducing states S_{O1}, S_{O2} (initial) and S_9 (final)
ε: start of the observation

$$P(S|O) = P(O|S)\,P(S)$$

observation	O_{01}	O_{02}	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	
state	S_{01}	S_{02}	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9

$$P(S|O) = P(O_1|S_1)\cdot P(S_1|S_{02},S_{01})\cdot P(O_2|S_2)\,P(S_2|S_1,S_{02})\cdot P(O_3|S_3)\,P(S_3|S_2,S_1)\cdot P(O_4|S_4)\,P(S_4|S_3,S_2)\cdot P(O_5|S_5)\,P(S_5|S_4,S_3)\cdot P(O_6|S_6)\,P(S_6|S_5,S_4)\cdot P(O_7|S_7)\,P(S_7|S_6,S_5)\cdot P(O_8|S_8)\,P(S_8|S_7,S_6)$$

$$P(S|O) = \prod_{k=1} P(O_k|S_k)\,P(S_k|S_{k-1},S_{k-2})$$

3. Word types frequently tagged incorrectly are:

AP+AP, AT+NN, AT+NP, BEM*, DO+PPSS, DTS+BEZ, EX+HVZ, IN+IN, IN+NP, IN+PPO, JJ$, JJ+JJ, JJR+CS, MD+HV, MD+PPSS, NIL, NN+BEZ, NN+HVD, NN+HVZ, NN+IN, NN+MD, NN+NN, NNS+MD, NP+BEZ,, NP+HVZ, NP+MD, NPS$, NR+MD, PN+BEZ, PN+HVD, PN+HVZ, PN+MD, PPL+VBZ, PPO+IN, PPSS+BEZ, PPSS+BEZ*, PPSS+VB, RB+CC, RB+CS, RBR+CS, RP+IN, TO+VB, VB+AT, VB+JJ, VB+RP, VB+VB, WDT+BER, WDT+BER+PP, WDT+DO+PPS, WDT+DOD, WPS+HVD, WPS+HVZ, WPS+MD, WRB+BER, WRB+DO, WRB+DOD, WRB+DOD*, WRB+DOZ, WRB+IN, WRB+MD

These word types have 0 precision this means that they are incorrectly tagged. This happens because they are not occurring frequently enough in the training set.