CSCI 544 HOMEWORK 2

Asmita Chotani

TASK 1:

What is the selected threshold for unknown words replacement?
The selected threshold for unknown words replacement was 3. The words occurring less than 3 times were considered as 'unk'.

What is the total size of your vocabulary and what is the total occurrences of the special token '< unk >' after replacement?
After replacement of rare words with the keyword "unk" the total size of the vocabulary came down to 16920, with "unk" occurring 32537 times.

EXPLANATION
A dictionary was created for the number of occurrences of each word in the training data. Different threshold values were considered, and in the end "3" was considered as threshold as it performed better.
Entries of the dictionary with threshold value less than or equal to 3 were then dropped.

To create the "vocab.txt" file, a new dictionary was created, with the first key being "unk":number of unknown words and then the rest of the words were appended from the previous dictionary.

The vocab.txt has been used throughout to determine whether the word is unknown or not.

TASK 2

How many transition and emission parameters in your HMM?
For better efficiency, the transition probabilities were calculated for all possible combinations of the 45 tags that were mentioned in the training data. Total combinations were 2025, some of the combinations also had a probability of 0.

For the emission probabilities, the (word, tag) pairs that occur in the training data was considered. The number of combinations added up to 23019 entries.

EXPLANATION

Working on the transition probabilities, all the possibilities were considered and the probability of tag t2 occurring after tag t1 was calculated using the below formula:

CSCI 544 HOMEWORK 2

Asmita Chotani

$$\frac{number\ of\ times\ t2\ came\ after\ t1}{number\ of\ times\ t1\ occurs}$$

Since there are pair of tags which do not occur in the training dataset, laplace smoothing was applied to the transition dictionary values to get non zero probabilities.

For the emission probabilities, the (word, tag) pairs were considered on reading the training data. The number of occurrence of each pair was updated in a dictionary(emission dictionary), which was then divided by the occurrence of the respective tag(stored in separate dictionary) to give the probability.

For easy access, the dictionaries were written into a .txt file, that was used in a separate notebook, which performed the Greedy and Viterbi algorithms.

For submission purpose, the dictionaries have been added to hmm.json as well.

The keys are in the form

Transition-> (t1,t2):value where t2 is the current tag and t1 is the previous tag- the same terminology has been used throughout the tasks.
Emission-> (tag,word):value

FOR TASK 3 and TASK 4:
The test data output has been formatted based on training data, i.e

$$< index\ in\ sentence >/t < word >/t < tag >$$

TASK 3

What is the accuracy on the dev data?
The accuracy of the greedy algorithm on the dev data was coming out to be around 90.69%.

EXPLANATION
Reading each word from the dev data provided, the function verifies the presence of the word in the emission dictionary. Since the word can be paired up with different tags, the probability for each pair is considered and the tag with the highest probability is finalized for that word. The (word, tag) is then appended to a list which is returned by the function and is used to calculate the accuracy of the model.

Researching about HMM, resources mentioned that one of the ways to assign tags to unknown words is to give them the tag that occurs the most times with the unknown words. Calculating the occurrence “NNP” occurred the most along with unknown words, hence if the word was

not found in the vocabulary list, it was directly assigned "NNP" without checking probabilities (which would not exist).

TASK 4

What is the accuracy on the dev data?

The accuracy of the viterbi algorithm on the dev data was coming out to be around 91.7%.

EXPLANATION

To keep track of sentences so that we can backtrack the probability values, the input data was pre-processed to create a list of the sentences provided based on the indexes provided. The Viterbi function iterates over the list of sentences and for each sentence it creates two lists, one for forward propagation and one for backward propagation.

For the first word, it considers transition from '.' to the states as new sentences mostly start after a '.'. The Viterbi function then finds the transition and emission probabilities from the dictionaries for the different words. To find the best tag, the transition prob, emission prob and probability values obtained during forward propagating were considered.

Since in Viterbi we backtrack our way to find the highest probability, initially tags were assigned while backtracking and then the result was reversed for each sentence to obtain the sentence along with the tags in the correct order.

There are three conditions- for every condition the following process is followed:

1. The tag, word pair is verified to be in emission dictionary, and the emission probability is extracted
2. The tag is considered with the previous tag i.e the tag assigned to the previous word and the transition probability is calculated.
3. The current word probability is calculated using the emission and transition probabilities
4. The tag with the best current probability is assigned to the state in the back propagation list, which will then be considered in the future and
5. The probability of that tag is included in the forward propagation list which is used in current probability calculation of previous states.

Since the tags are being read from the set of tags present in train, it is important to verify whether the tag, word pair exist in training data, hence a check has been applied every time a new tag is considered for the word and in case of absence, the probability is assigned as 0. Similarly, there are pairs of tags that are not present in the training data A check is applied for verifying the the presence of the same in the transition dictionary.

CSCI 544 HOMEWORK 2

Asmita Chotani

If the word is considered as an unknown word, the occurence of tag with the keyword '<unk>' is considered. Based on the vocabulary created using training data, words occurring less than 3 times have been replaced with the '<unk>' keyword.
The tag for the uknown words were determined using predefined morphological rules to determine type of word based on the structure of word.
Also, based on research, it was determined that one of the ways to deal with unknown words is to assign the tag which occurs the most along the unknown words of the training dictionary. The most occuring tag,'<unk>' pair was found to be with the NNP tag. Hence, if the tag,'unk' does not exist in the emission dictionary, the value of NNP,'<unk>' is obtained from the emission dictionary and used.