**Hidden Markov Model Part-of-Speech Tagger for Hindi Language**

**Overview**

A Hidden Markov Model part-of-speech tagger for Hindi language. The training data is tokenized and tagged; the test data is also tokenized and the taggers add the tags to the test data.

**Training and Development data:**

A tagged hindi file (hi\_test\_tagged.txt file) in the word/TAG format, with words separated by spaces and each sentence on a new line is used as training data. Another hindi file (hi\_test\_raw.txt file) with untagged development data, with words separated by spaces and each sentence on a new line is used as testing data . hmmmoutput.txt is the output file with tagged development data in the word/TAG format, with words separated by spaces and each sentence on a new line.

**Programs**

Consists of two programs:

hmmlearn.py learns a hidden Markov model from the training data((hi\_test\_tagged.txt), and hmmdecode.py uses the model to tag new data.

Initially the hmmlearn.py will run .The program will learn a hidden Markov model, and write the model parameters to a file called hmmmodel.txt.

Then, the hmmdecode.py will run .The program reads the parameters of a hidden Markov model from the file hmmmodel.txt, tag each word in the test data, and write the results to a text file called hmmmoutput.txt in the same format as the training data.

**TO EXECUTE THE PROJECT, RUN hmmlearn.py FILE.**

**Accuracy**

The accuracy of our tagger is determined by comparing the output of the tagger in hmmmoutput.txt file to a reference tagged text file named hi\_test\_tagged file using confusion matrix which takes two parameters named expect and predict.

Expect parameter is an array of tags of initial training data (hi\_test\_tagged.txt) file.

Predict parameter is an array of tags of final output data (hmmmoutput.txt) file.

**Results on Test Data**

Accuracy obtained for Hindi test data of 35,430 words using HMM algorithm is 97.63% .