MODEL 1 :

HMM AND VITERBI ALGORITHM WITHOUT MORPH ANALYSIS

Experiment -
The utf file was converted to wx notation using the convertor.
The output file was used to train the model.
The train data contains 1000 sentences.
The test data contains 100 sentences.

Observations:
Correction of training data POS
- XC tag was not used correctly.
- Proper Nouns were tagged as Nouns
- VAUX were tagged as VM.

Emission Matrix -
Contains probability of a word(X) being tagged as Y.
P( Y | X ) = Count(X,Y) / Count(X).
These values are computed using the training data. It is a 2-D matrix.
Rows represent unique words and columns represent Tags.

Transition Matrix -
Contains probability of Tag1(T1) being followed by Tag2(T2).
P( T2 | T1 ) = Count(T1,T2) / Count(T1).
These values are computed using the training data. It is a 2-D matrix.
Rows represent T1 tags or previous tags and columns represent T2 tags ot current tags.

Maximum Likelihood of a Tag sequence -
Probability of tag sequence = prob of the word being tagged as T2 * probability of T2 following T1
This is computed for all words and the sequence maximising the probability is the best tag sequence.

Viterbi -
implemented using dynamic programming

HMM -
assumed to be dependent on the previous state only

Assumptions:
Handling of unseen words – unseen words are assumed to be nouns.

Results -
The unseen words were initially tagged as Nouns but the in best tag sequence result get coverted to other tags appropriately

MODEL 2

HMM AND VITERBI ALGORITHM WITH NOUN MORPH ANALYSIS

Experiment -
The utf file was converted to wx notation using the convertor.
The output file was used to train the model.
The train data contains 1000 sentences.
The test data contains 100 sentences.

Observations:
Correction of training data POS
- XC tag was not used correctly.
- Proper Nouns were tagged as Nouns
- VAUX were tagged as VM.

Creation of paradigm tables - 6 categories ( Masculine – 3 , feminine – 3) were observed.
Within each category, Direct and oblique cases were observed for singular and plural forms.

Emission Matrix -
Contains probability of a word(X) being tagged as Y.
P( Y | X ) = Count(X,Y) / Count(X). observed
These values are computed using the training data. It is a 2-D matrix.
Rows represent unique words and columns represent Tags.

Transition Matrix -
Contains probability of Tag1(T1) being followed by Tag2(T2).
P( T2 | T1 ) = Count(T1,T2) / Count(T1).
These values are computed using the training data. It is a 2-D matrix.
Rows represent T1 tags or previous tags and columns represent T2 tags ot current tags.

Maximum Likelihood of a Tag sequence -
Probability of tag sequence = prob of the word being tagged as T2 * probability of T2 following T1
This is computed for all words and the sequence maximising the probability is the best tag sequence.

Viterbi -
implemented using dynamic programming

HMM -
assumed to be dependent on the previous state only

SFST -
6 .lex and .fst files were created on the basis of hindi paradigm rules.

Feminine Nouns -

| | Type 1 | | Type 2 | | Type 3 | |
|---|---|---|---|---|---|---|
| | Direct | Oblique | Direct | Oblique | Direct | Oblique |
| Singular | ***null*** | ***null*** | ***null*** | ***null*** | ***null*** | ***null*** |
| Examples | *āg, pyās* | *āg, pyās* | *nədī, məNɪ* | *nədī, məNɪ* | *lətā, bāt* | *lətā, bāt* |
| | | | | | | |
| Plural | ***null*** | ***null*** | ***-yə̃*** | ***-yõ*** | ***-ẽ*** | ***-õ*** |
| Examples | *āg, pyās* | *āg, pyās* | *nədɪ-yə̃,məNɪ-yə̃* | *nədɪ-yõ,məNɪ-yõ* | *lətā-ẽ,bāt-ẽ* | *lətā-õ,bāt-õ* |

Masculine Nouns -

| | Type 1 | | Type 2 | | Type 3 | |
|---|---|---|---|---|---|---|
| | Direct | Oblique | Direct | Oblique | Direct | Oblique |
| Singular | ***null*** | ***null*** | ***null*** | ***-e*** | ***null*** | ***null*** |
| Examples | *krodh, pyār* | *krodh,pyār* | *lərkā, sāyā* | *lərk-e,sāy-e* | *ādmī, ghər* | *ādmī, ghər* |
| | | | | | | |
| Plural | ***null*** | ***null*** | ***-e*** | ***-õ*** | ***null*** | ***-õ/-yõ*** |
| Examples | *krodh, pyār* | *krodh,pyār* | *lərk-e,sāy-e* | *lərk-õ,sāy-õ* | *ādmī,ghər* | *ādmɪ-yõ, ghər-õ* |

Manually nouns were categorised into these 6 categories.

In the .fst file -
The states in the Finite State Transducers were identified on the basis of the suffixes of the input word.
Alphabet were defined according to occuring inflections.
Delete rules were formed according to each sub-cases within all the 6 categories

Also, the modifications required for inflections were taken care of – separately for all the unique occurences.

Assumptions:
Handling of unseen words – unseen words are assumed to be nouns.
Morph analysis is done only for nouns.

Results -
The unseen words were initially tagged as Nouns but the in best tag sequence result get coverted to other tags appropriately