Name: Muhammad Mobeen

Registration No: 200901097

BSCS01-(B)

Presented to: Mam Reeda Saeed

Big Data Analytics

Assignment #3

Hidden Markov Model

1. Dataset

The dataset used in this assignment is the NER_dataset.csv file, which contains a collection of sentences and their corresponding Part-of-Speech (POS) tags. The dataset is loaded into a Pandas DataFrame, and any missing values are filled using the ffill method.

2. Preprocessing

The following preprocessing steps are performed on the dataset:

1. The column names are renamed: 'Sentence #' is changed to 'sentence'.
2. Duplicate words and tags are removed by converting them into sets.
3. The dataset is split into training and testing sets using GroupShuffleSplit from scikit-learn. This ensures that sentences are not split across the training and testing sets.
4. To increase the vocabulary size in the training set, a fraction of the training data is randomly selected, and their words are replaced with the 'UNKNOWN' token.
5. The words and tags are mapped to unique integer values using dictionaries.

3. Co-relation Table

The code does not explicitly create a co-relation table to show the relationships between columns. However, it calculates the following counts and probabilities:

- count_tags: The number of occurrences of each POS tag in the training set.
- count_tags_to_words: A dictionary mapping POS tags to another dictionary, which maps words to their counts for that tag.
- count_init_tags: The number of occurrences of each POS tag as the first tag in a sentence.
- count_tags_to_next_tags: A 2D matrix that stores the count of transitions from one POS tag to another within a sentence.

These counts and probabilities are used to calculate the initial probabilities (startprob_), transition probabilities (transmat_), and emission probabilities (emissionprob_) required for training the HMM.

4. Graphs of Connected Words

The code does not generate any graphs or visualizations of connected words or their occurrence probabilities.

5. Applying HMM

The HMM is implemented using the hmmlearn library. The following steps are performed:

1. The MultinomialHMM model is initialized with the number of components equal to the number of POS tags.
2. The model parameters (startprob_, transmat_, and emissionprob_) are calculated from the counts and probabilities obtained during preprocessing.
3. The testing data is preprocessed by replacing any unseen words with the 'UNKNOWN' token.
4. The testing data is transformed into a format suitable for the HMM's predict method.
5. The HMM's predict method is used to obtain the predicted POS tags for the testing data.

6. Hyperparameter Tuning

The code does not perform any hyperparameter tuning for the HMM model. The HMM is initialized with default hyperparameters, except for setting the algorithm parameter to 'viterbi'.

7. Testing on Unseen Data

The performance of the trained HMM model is evaluated on the testing dataset. The following metrics are calculated:

- Accuracy
- Precision (weighted average)
- Recall (weighted average)
- F1-score (weighted average)

Due to an issue with the size of the predicted output, the evaluation is performed only on the shorter length between the predicted and actual outputs.