Experiment - 5

Aim: To calculate emission and transition matrix for tagging Parts of Speech using Hidden Markov Model.
Find POS tag of given sentence using HMM.

Theory:

POS tagging or part-of-speech tagging is the procedure of assigning a grammatical category like noun, verb, adjective etc. to a word. In this process both the lexical information and the context play an important role as the same lexical form can behave differently in a different context.

For example the word "Park" can have two different lexical categories based on the context.

The boy is playing in the park. ('Park' is Noun)
Park the car. ('Park' is Verb)

Assigning part of speech to words by hand is a common exercise one can find in an elementary grammar class. But here we wish to build an automated tool which can assign the appropriate part-of-speech tag to the words of a given sentence.

One can think of creating handcrafted rules by observing patterns in the language, but this would limit the system's performance to the quality and number of patterns identified by the rule crafter. Thus, this approach is not practically adopted for building POS Taggers.

Instead, a large corpus annotated with correct POS tags for each word is given to the computer and algorithms then learn the patterns automatically from the data and store them in form of a trained model. Later this model can be used to POS tag new sentences.

POS Tagging - Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible.

Hidden Markov Model has two important components-

1)Transition Probabilities: The one-step transition probability is the probability of transitioning from one state to another in a single step.

2)Emission Probabilities: : The output probabilities for an observation from state. Emission probabilities B = { bi,k = bi(ok) = P(ok | qi) }, where ok is an Observation. Informally, B is the probability that the output is ok given that the current state is qi

For POS tagging, it is assumed that POS are generated as a random process, and each process randomly generates a word.

Hence, the transition matrix denotes the transition probability from one POS to another and the emission matrix denotes the probability that a given word can have a particular POS. Words act as the observations. Some of the basic assumptions are:

Calculating the Probabilities

Consider the given toy corpus

```
EOS/eos
                                        They/pronoun
                                        cut/verb
                                        the/determiner
                                        paper/noun
                                        EOS/eos He/pronoun
                                        asked/verb
                                        for/preposition
                                        his/pronoun
                                        cut/noun.
                                        EOS/eos
                                        Put/verb
                                        the/determiner

                                        paper/noun
                                        in/preposition
                                        the/determiner
                                        cut/noun
                                        EOS/eos
```

Calculating Emission Probability Matrix

Count the no. of times a specific word occus with a specific POS tag in the corpus. Here, say for **"cut"**

```
count(cut,verb)=1
          count(cut,noun)=2
          count(cut,determiner)=0
```

and so on zero for other tags too.

```
count(cut) = total count of cut = 3
```

Now, calculating the probability
Probability to be filled in the matrix cell at the intersection of cut and verb

```
P(cut/verb)=count(cut,verb)/count(cut)=1/3=0.33
```

Similarly,
Probability to be filled in the cell at he intersection of cut and determiner

```
P(cut/determiner)=count(cut,determiner)/count(cut)=0/3=0
```

Repeat the same for all the word-tag combination and fill the

Calculating Transition Probability Matrix

Count the no. of times a specific tag comes after other POS tags in the corpus.
Here, say for **'determiner'**

```
count(verb,determiner)=2
          count(preposition,determiner)=1
          count(determiner,determiner)=0
          count(eos,determiner)=0
          count(noun,determiner)=0
```

and so on zero for other tags too.

```
count(determiner) = total count of tag 'determiner' = 3
```

Now, calculating the probability Probability to be filled in the cell at he intersection of determiner(in the column) and verb(in the row)

```
P(determiner/verb)=count(verb,determiner)/count(determiner)=2/3=0.66
```

Similarly,
Probability to be filled in the cell at he intersection of determiner(in the column) and noun(in the row)

```
P(determiner/noun)=count(noun,determiner)/count(determiner)=0/3=0
```

Repeat the same for all the tags

Note: EOS/eos is a special marker which represents ***End Of Sentence***.

Input: Text Corpus of sufficient length. For example movie reviews, newspaper articles, etc.
Output: For the given corpus fill the emission and transition matrix and find the POS tags for a given sentence.

Curiosity Questions

1. List a few ways for tagging parts of speech?
2. How do you find the most probable sequence of POS tags from a sequence of text?
3. Differentiate between Markov chain and Markov model?
4. How you can identify whether a system follows a Markov Process?
5. Explain the use of Markov Chains in text generation algorithms.

References:

- https://analyticsindiamag.com/a-guide-to-hidden-markov-model-and-its-applications-in-nlp/

- https://www.mygreatlearning.com/blog/pos-tagging/#:~:text=HMM%20(Hidden%20Markov%20Model)%20is,%2C%20partial%20discharges%2C%20and%20bioinformatics.
- https://medium.com/hackernoon/building-a-bigram-hidden-markov-model-for-part-of-speech-tagging-1b784a87ab2c