HomeWork 3

Corpus:-

(a) “We had a nice party yesterday”

(b) “She came to visit me two days ago”

(c) “You may go now”

(d) “Their kids are not always naive”

The Grammar for these sentences:-

```
my_grammer_test = nltk.CFG.fromstring("""
 S -> NP VP | VP
 NP -> Prop | Det PP N |  ADJ N | Prop N | ADJ N ADV
 VP -> V NP | V NP NP | V PP | MD VP | V ADV PP | V ADV | AUX NP
 PP -> P NP | P VP | ADV ADJ | ADJ N | ADJ
 AUX -> "had"
 V -> "came" | "go" | "visit"  | "are" | "am" | "went" | "meet" | "is"
 ADJ -> "nice" | "naive" | "two" | "good" | "three" | "greatest"
 ADV -> "ago" | "now" | "not" | "always"
 Prop -> "We" | "She" | "You" | "Their" | "me" | "I" | "John" | "Maria" | "Rock"
 Det -> "a" | "the"
 MD -> "may"
 N -> "party" | "yesterday" | "kids" | "days" | "person" | "years" | "WWE" | "champion"
 P -> "to"""")
```

I have added extra words to parse the other 3 sentences which are listed below:-

- Rock is the greatest WWE champion
- John went to meet Maria three years ago
- I am a good person
- John meet to Maria

From these 3 sentences “John meet to Maria” doesn’t make sense.

In this one “John” is a Pronoun , “meet” is a verb , “to” is a P and “Maria” is another Pronoun. So based on the Grammar it is satisfied. But It doesn’t make any sense. If something like “John wants to buy.” This makes sense. So because of the grammer doesn’t understand the grammar rules it satisfies the sentence but cant make sense out of it.

Because of my grammar rules VP --> V NP NP. It redirects to the NP grammar and it doesn’t allow any other Nouns to consider rather than this.

Now PCFG:-

For Probabilistic CFG, We first have to calculate the frequency of words in our corpus that we are going to use:-

```
Counter({'We': 1, 'had': 1, 'a': 1, 'nice': 1, 'party': 1,
'yesterday': 1, 'She': 1, 'came': 1, 'to': 1, 'visit': 1, 'me':
1, 'two': 1, 'days': 1, 'ago': 1, 'You': 1, 'may': 1, 'go': 1,
'now': 1, 'Their': 1, 'kids': 1, 'are': 1, 'not': 1, 'always': 1,
'naive': 1})
```

So, the probability of each word will be equal in their respective categories.

Now for the probability for the Grammar is based on how frequent a Grammar Rule is used in the corpus.

For Example, The rule S -> NP VP | VP, there are two ways to go from S as one from NP VP and other as VP. In my grammar most of the time it chooses as the NP VP for parsing the sentence. So I

kept the Higher Probability for NP VP.

For NP there are 5 rules as

NP -> Prop [0.4]| Det ADJ N N [0.25]| ADJ N [0.15]| Prop N [0.1]| ADJ N ADV[0.1]

Here I set the probability of Prop higher compare to the rest of rules because in each sentence the First word is a Pronoun and it going to be called for each sentence. The other two are used more than the rule which has probability of 0.1 so their probability is higher.

As like in PP the grammar has 3 rules:-

PP -> P NP [0.5] | P VP [0.3]| ADV ADJ [0.2] , Here P NP has been used 2 times more than compare to others so I set the probability Higher. ADV ADJ rule has been used just once so its probability is lower than others.

The PCFG is given below:-

```
prob_grammar_prac = nltk.PCFG.fromstring("""
 S -> NP VP [0.9]| VP [0.1]
 VP -> TranV NP [0.17]| TranV NP NP [0.166] | TranV VP[0.166]
 VP -> InV ADV [0.166]
 VP -> DatV ADV PP [0.166]| DatV PP [0.166]
 PP -> P NP [0.5] | P VP [0.3]| ADV ADJ [0.2]
 TranV -> "visit" [0.34] | "had" [0.33] | "may" [0.33]
 InV -> "go" [1.0]
 DatV -> "came" [0.5] | "are"[0.5]
 NP -> Prop [0.4]| Det ADJ N N [0.25]| ADJ N [0.15]| Prop N [0.1]| ADJ N ADV[0.1]
 ADJ -> "nice" [0.34] | "naive" [0.33] | "two" [0.33]
 ADV -> "ago" [0.25] | "now" [0.25] | "not" [0.25] | "always" [0.25]
 Prop -> "We" [0.2] | "She" [0.2] | "You" [0.2] | "Their" [0.2] | "me" [0.2]
 Det -> "a" [1.0]
 N -> "party" [0.25] | "yesterday" [0.25] | "kids" [0.25] | "days" [0.25]
 P -> "to" [1.0]
 """)
```

Appendix:-

```
Sentence 1
(S
  (NP (Prop We))
  (VP (AUX had) (NP (Det a) (PP (ADJ nice) (N party)) (N yesterday))))
Sentence 2
(S
  (NP (Prop She))
  (VP
    (V came)
    (PP
      (P to)
      (VP (V visit) (NP (Prop me)) (NP (ADJ two) (N days) (ADV ago))))))
Sentence 3
(S (NP (Prop You)) (VP (MD may) (VP (V go) (ADV now))))
Sentence 4
(S
  (NP (Prop Their) (N kids))
  (VP (V are) (ADV not) (PP (ADV always) (ADJ naive))))

Extra 1
(S
  (NP (Prop Rock))
  (VP (V is) (NP (Det the) (PP (ADJ greatest) (N WWE)) (N champion))))
Extra 2
(S
  (NP (Prop John))
  (VP
    (V went)
    (PP
      (P to)
      (VP
        (V meet)
        (NP (Prop Maria))
        (NP (ADJ three) (N years) (ADV ago))))))
Extra 3
(S (NP (Prop I)) (VP (V am) (NP (Det a) (PP (ADJ good)) (N person))))
Sense less Sentence
(S (NP (Prop John)) (VP (V meet) (PP (P to) (NP (Prop Maria)))))
```

```
['We', 'had', 'a', 'nice', 'party', 'yesterday', 'She', 'came', 'to', 'visit', '
me', 'two', 'days', 'ago', 'You', 'may', 'go', 'now', 'Their', 'kids', 'are', 'n
ot', 'always', 'naive']
Counter({'We': 1, 'had': 1, 'a': 1, 'nice': 1, 'party': 1, 'yesterday': 1, 'She'
: 1, 'came': 1, 'to': 1, 'visit': 1, 'me': 1, 'two': 1, 'days': 1, 'ago': 1, 'Yo
u': 1, 'may': 1, 'go': 1, 'now': 1, 'Their': 1, 'kids': 1, 'are': 1, 'not': 1, '
always': 1, 'naive': 1})
```

```
Probabilistic CFG
Sentence 1
(S
  (NP (Prop We))
  (VP
    (TranV had)
    (NP (Det a) (ADJ nice) (N party) (N yesterday)))) (p=2.14583e-05)
Sentence 2
(S
  (NP (Prop She))
  (VP
    (DatV came)
    (PP
      (P to)
      (VP
        (TranV visit)
        (NP (Prop me))
        (NP (ADJ two) (N days) (ADV ago)))))) (p=1.66956e-08)
Sentence 3
(S
  (NP (Prop You))
  (VP (TranV may) (VP (InV go) (ADV now)))) (p=0.000163683)
Sentence 4
(S
  (NP (Prop Their) (N kids))
  (VP (DatV are) (ADV not) (PP (ADV always) (ADJ naive)))) (p=1.54069e-06)
```

Note:- This is the output of the given code. If you want to check you can simply run the code.

Conclusion:-

In this Homework, we learned how to make a grammar for different sentences and from that grammar we can generate other sentences. The PCFG is a nice way to predict the grammar rules probability as which rule is frequently used and which ones are useful for most of the sentences.