Experiment No 04: N gram Modelling

Rebecca Dias

Roll no: 18 BE CMPN A2

Pid: 182027

Aim : To implement the Ngram model from a text corpus and do adjacent word prediction in Python

Bigrams

```
string = "<s> I am hungry </s> <s> I am honest </s> <s> I love chocolate </s>".split(" ")
string
```

```
['<s>',
 'I',
 'am',
 'hungry',
 '</s>',
 '<s>',
 'I',
 'am',
 'honest',
 '</s>',
 '<s>',
 'I',
 'love',
 'chocolate',
 '</s>']
```

```
def calcProb(firstWord, secondWord):
  countFollowedBy = 0
  for i in range(len(string)-1):
    if string[i] == firstWord and string[i+1] == secondWord:
      countFollowedBy += 1

  return countFollowedBy/string.count(firstWord)

d1 = {}
bigrams = []
for i in range(len(string) - 1):
  pair = string[i]+", "+string[i+1]
  if pair == "</s>, <s>":
    continue

  if pair not in bigrams:
    bigrams.append(pair)
    d1[pair] = calcProb(string[i], string[i+1])
```

```
bigrams
```

```
['<s>, I',
 'I, am',
 'am, hungry',
 'hungry, </s>',
 'am, honest',
 'honest, </s>',
 'I, love',
 'love, chocolate',
 'chocolate, </s>']
```

```
d1
```

```
{'<s>, I': 1.0,
 'I, am': 0.6666666666666666,
 'I, love': 0.3333333333333333,
 'am, honest': 0.5,
 'am, hungry': 0.5,
 'chocolate, </s>': 1.0,
 'honest, </s>': 1.0,
 'hungry, </s>': 1.0,
 'love, chocolate': 1.0}
```

```
inputWord = input("Enter a word: ")
```

```
    Enter a word: I
```

```
dNew = {}

d1List = d1.keys()

for pair in d1List:
  if pair.split(", ")[0] == inputWord:
    dNew[pair.split(", ")[1]] = d1[pair]

print("The bigram of '"+inputWord+"' would be:\n")
for word in dNew:
  print(word + ": " + str(dNew[word]))
```

```
    The bigrams of 'I' would be:

    am: 0.6666666666666666
    love: 0.3333333333333333
```

Trigrams

```
string = "<s> I am hungry </s> <s> I am honest </s> <s> I love chocolate </s>".split(" ")
string
```

```
    ['<s>',
     'I',
     'am',
     'hungry',
     '</s>',
     '<s>',
     'I',
     'am',
     'honest',
     '</s>',
     '<s>',
     'I',
     'love',
     'chocolate',
     '</s>']
```

```
def calcProb(firstWord, secondWord, thirdWord):
  countFollowedBy = 0
  for i in range(len(string)-2):
    if string[i] == firstWord and string[i+1] == secondWord and string[i+2] == thirdWord:
      countFollowedBy += 1

  return countFollowedBy/string.count(firstWord)

d2 = {}
trigrams = []
for i in range(len(string) - 2):
  trigram = string[i]+", "+string[i+1]+", "+string[i+2]
  if "</s>, <s>" in trigram:
    continue

  if trigram not in trigrams:
    trigrams.append(trigram)
    d2[trigram] = calcProb(string[i], string[i+1], string[i+2])
```

```
trigrams
```

```
    ['<s>, I, am',
     'I, am, hungry',
     'am, hungry, </s>',
     'I, am, honest',
     'am, honest, </s>',
     '<s>, I, love',
     'I, love, chocolate',
     'love, chocolate, </s>']
```

```
d2
```

```
    {'<s>, I, am': 0.6666666666666666,
     '<s>, I, love': 0.3333333333333333,
     'I, am, honest': 0.3333333333333333,
     'I, am, hungry': 0.3333333333333333,
     'I, love, chocolate': 0.3333333333333333,
     'am, honest, </s>': 0.5,
     'am, hungry, </s>': 0.5,
     'love, chocolate, </s>': 1.0}
```

```
inputWords = input("Enter two words: ")
```

```
    Enter two words: I am
```

```
dNew = {}

d1List = d2.keys()

for trigram in d1List:
  if trigram.split(", ")[0] + " " + trigram.split(", ")[1] == inputWords:
    dNew[trigram.split(", ")[2]] = d2[trigram]

print("The trigram of '"+inputWords+"' would be:\n")
for word in dNew:
  print(word + ": " + str(dNew[word]))
```

```
    The trigram of 'I am' would be:

    hungry: 0.3333333333333333
    honest: 0.3333333333333333
```

Conclusion:

In this experiment we learned about N Gram Modelling using python programming language.