دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

گزارش پروژه‌ی چهارم درس مبانی هوش مصنوعی

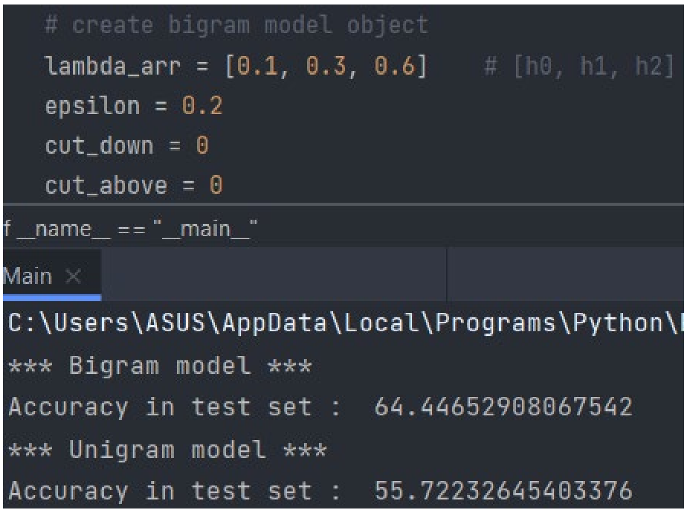
استاد روشن‌فکر

آرمین ذوالفقاری داریانی ۹۷۳۱۰۸۲ و امیرحسین رجب‌پور ۹۷۳۱۰۸۵

بهار ۱۴۰۰

تاثیر حذف کلمات پرتکرار و کم تکرار در دقت بدست آمده:

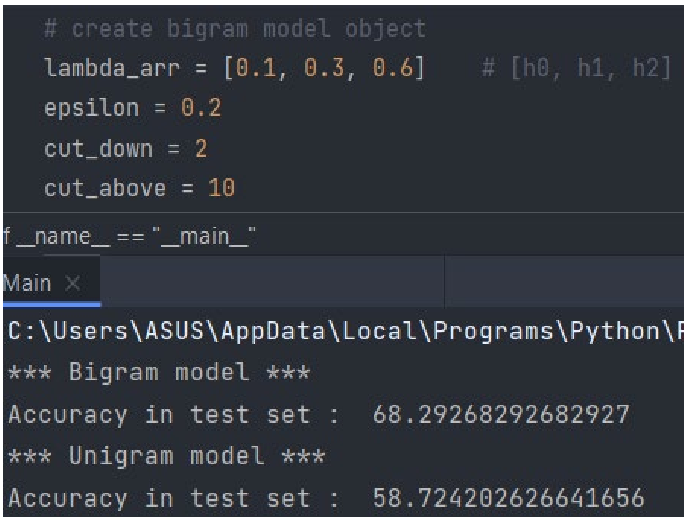
- Epsilon: 0.2
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]
  Cut_down = 0
  Cut_above = 0



```
# create bigram model object
lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 0
cut_above = 0
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  64.44652908067542
*** Unigram model ***
Accuracy in test set :  55.72232645403376
```

- Epsilon: 0.2
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]
  Cut_down = 2
  Cut_above = 10



```
# create bigram model object
lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 2
cut_above = 10
```

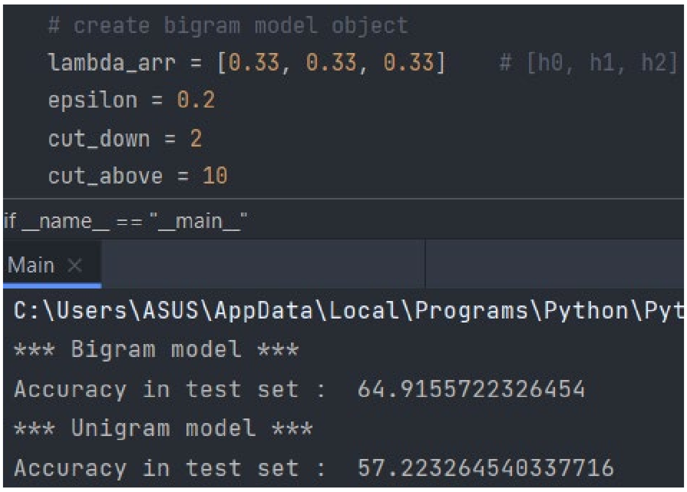
```
C:\Users\ASUS\AppData\Local\Programs\Python\F
*** Bigram model ***
Accuracy in test set :  68.29268292682927
*** Unigram model ***
Accuracy in test set :  58.724202626641656
```

تحلیل : می توان دید که با حذف کردن کلمات با تکرار بسیار کم (۲ به پایین (که معمولا اسامی خاص هستند و تاثیر چندانی در بار مثبت یا منفی داشتن جملات ندارند)) و تکرار بسیار زیاد

۱۰) کلمه با بیشترین تکرار (این کلمات عموما stop words هستند و تاثیری در بار مثبت یا منفی داشتن جملات ندارند)) بازدهی بهتری داریم و مدل می‌تواند بهتر بار مثبت و یا منفی داشتن کامنت‌ها را تشخیص دهد.

تاثیر مقدار λ و اپسیلون دقت بدست آمده:

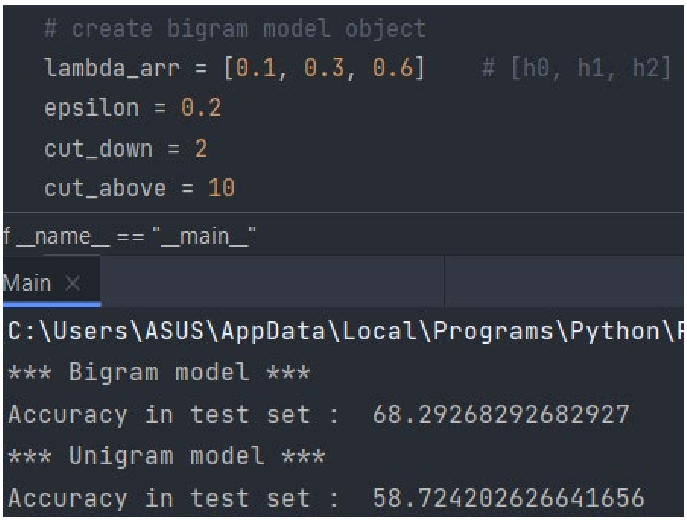
- Epsilon: 0.2
  Lambda: [L1:0.33 , L2:0.33 , L3:0.33]



```
# create bigram model object
lambda_arr = [0.33, 0.33, 0.33]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\Pyt
*** Bigram model ***
Accuracy in test set :  64.9155722326454
*** Unigram model ***
Accuracy in test set :  57.223264540337716
```

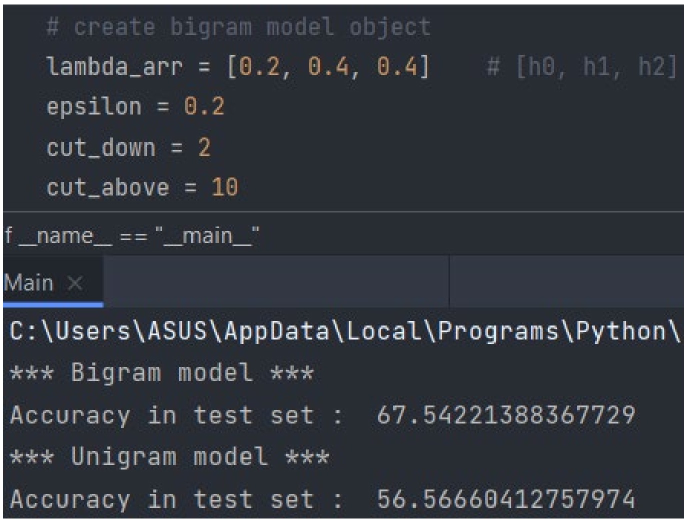
- Epsilon: 0.2
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]



```
# create bigram model object
lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\F
*** Bigram model ***
Accuracy in test set :  68.29268292682927
*** Unigram model ***
Accuracy in test set :  58.724202626641656
```

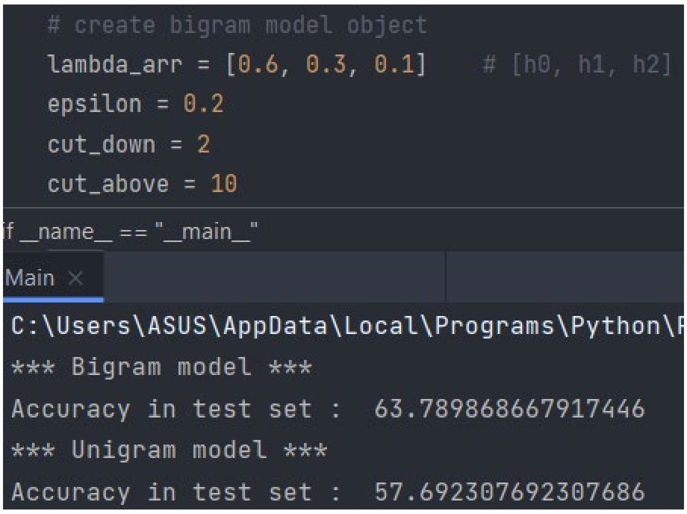
- Epsilon: 0.2
  Lambda: [L1:0.2 , L2:0.4 , L3:0.4]



```
# create bigram model object
lambda_arr = [0.2, 0.4, 0.4]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  67.54221388367729
*** Unigram model ***
Accuracy in test set :  56.56660412757974
```

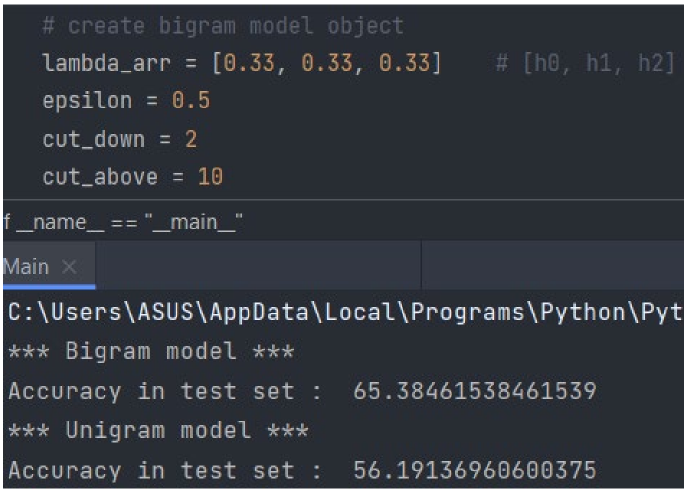
- Epsilon: 0.2
  Lambda: [L1:0.6 , L2:0.3 , L3:0.1]



```
# create bigram model object
lambda_arr = [0.6, 0.3, 0.1]    # [h0, h1, h2]
epsilon = 0.2
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\P
*** Bigram model ***
Accuracy in test set :  63.789868667917446
*** Unigram model ***
Accuracy in test set :  57.692307692307686
```

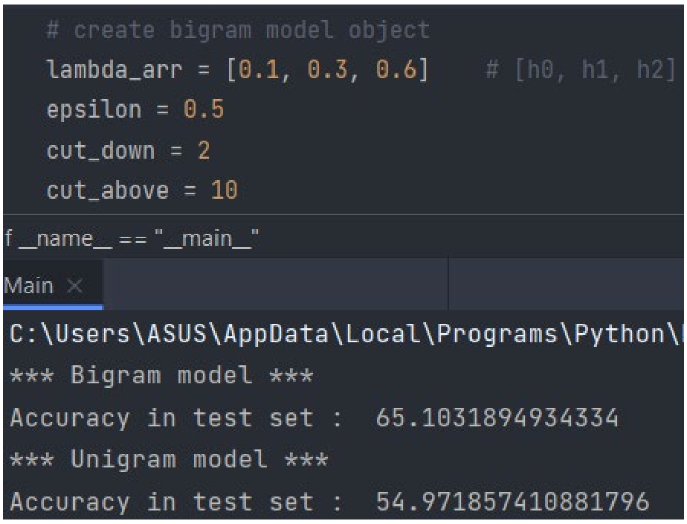
- Epsilon: 0.5
  Lambda: [L1:0.33 , L2:0.33 , L3:0.33]



```
# create bigram model object
lambda_arr = [0.33, 0.33, 0.33]    # [h0, h1, h2]
epsilon = 0.5
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\Pyt
*** Bigram model ***
Accuracy in test set :   65.38461538461539
*** Unigram model ***
Accuracy in test set :   56.19136960600375
```

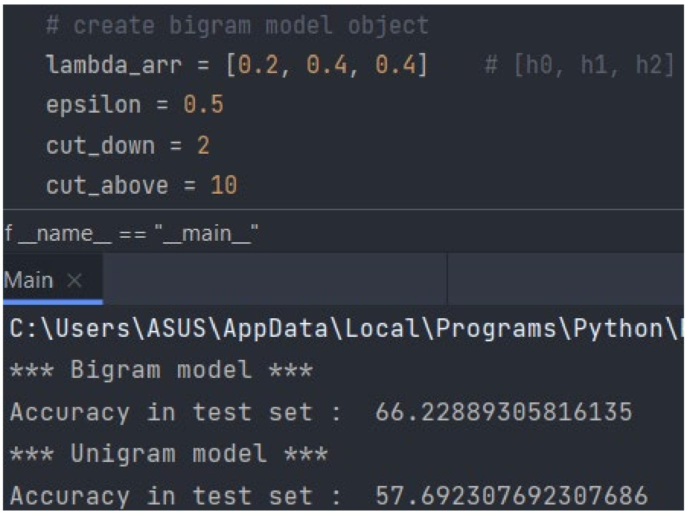
- Epsilon: 0.5
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]



```
# create bigram model object
lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
epsilon = 0.5
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :   65.1031894934334
*** Unigram model ***
Accuracy in test set :   54.971857410881796
```

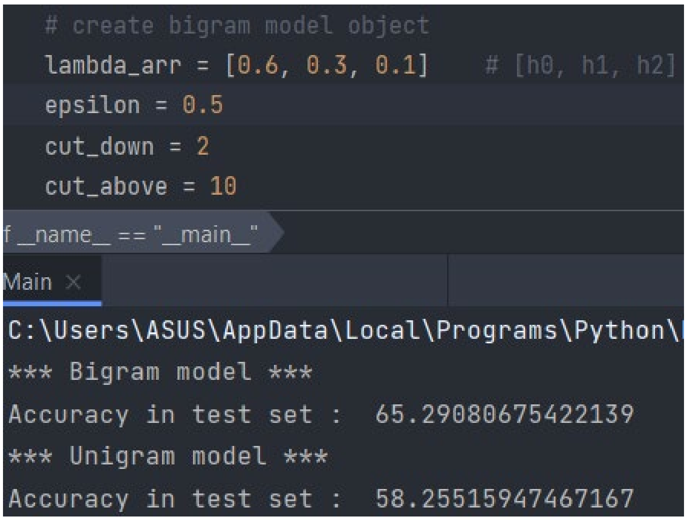
- Epsilon: 0.5
  Lambda: [L1:0.2 , L2:0.4 , L3:0.4]



```
    # create bigram model object
    lambda_arr = [0.2, 0.4, 0.4]    # [h0, h1, h2]
    epsilon = 0.5
    cut_down = 2
    cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  66.22889305816135
*** Unigram model ***
Accuracy in test set :  57.692307692307686
```

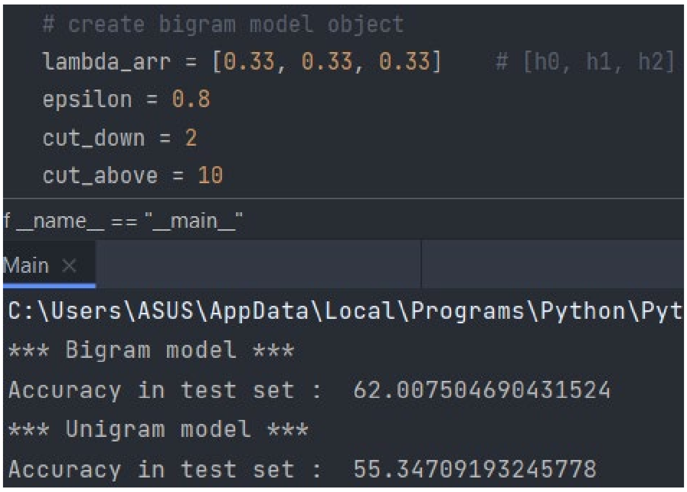
- Epsilon: 0.5
  Lambda: [L1:0.6 , L2:0.3 , L3:0.1]



```
    # create bigram model object
    lambda_arr = [0.6, 0.3, 0.1]    # [h0, h1, h2]
    epsilon = 0.5
    cut_down = 2
    cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  65.29080675422139
*** Unigram model ***
Accuracy in test set :  58.25515947467167
```

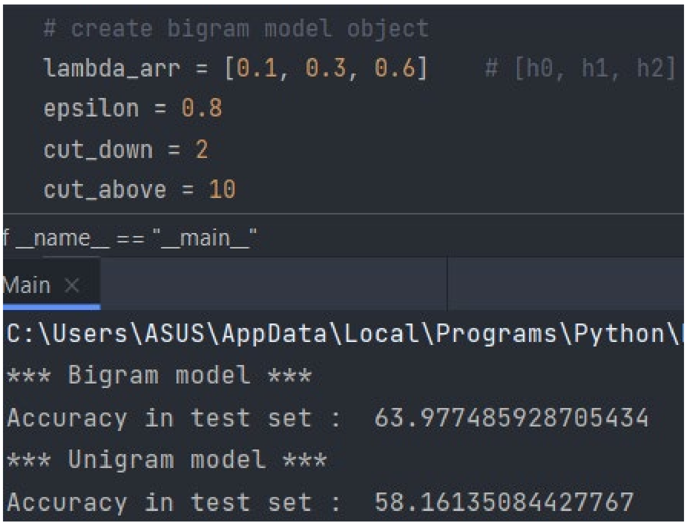
- Epsilon: 0.8
  Lambda: [L1:0.33 , L2:0.33 , L3:0.33]



```
# create bigram model object
lambda_arr = [0.33, 0.33, 0.33]    # [h0, h1, h2]
epsilon = 0.8
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\Pyt
*** Bigram model ***
Accuracy in test set :  62.007504690431524
*** Unigram model ***
Accuracy in test set :  55.34709193245778
```

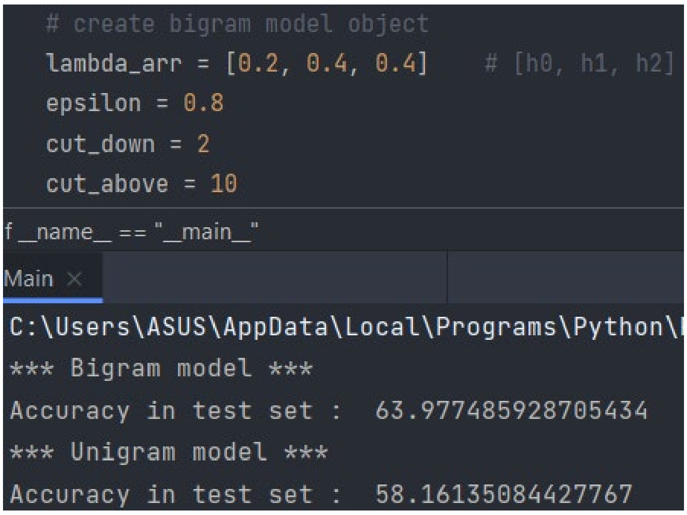
- Epsilon: 0.8
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]



```
# create bigram model object
lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
epsilon = 0.8
cut_down = 2
cut_above = 10
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  63.977485928705434
*** Unigram model ***
Accuracy in test set :  58.16135084427767
```

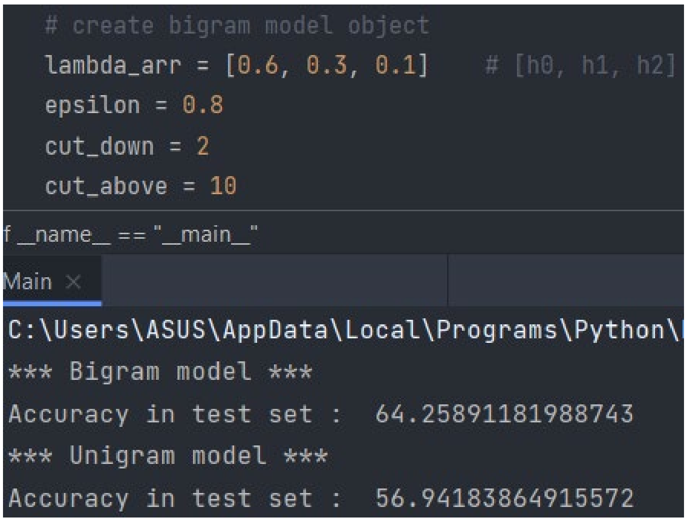
- Epsilon: 0.8
  Lambda: [L1:0.2 , L2:0.4 , L3:0.4]



```
    # create bigram model object
    lambda_arr = [0.2, 0.4, 0.4]    # [h0, h1, h2]
    epsilon = 0.8
    cut_down = 2
    cut_above = 10
f __name__ == "__main__"
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  63.977485928705434
*** Unigram model ***
Accuracy in test set :  58.16135084427767
```

- Epsilon: 0.8
  Lambda: [L1:0.6 , L2:0.3 , L3:0.1]



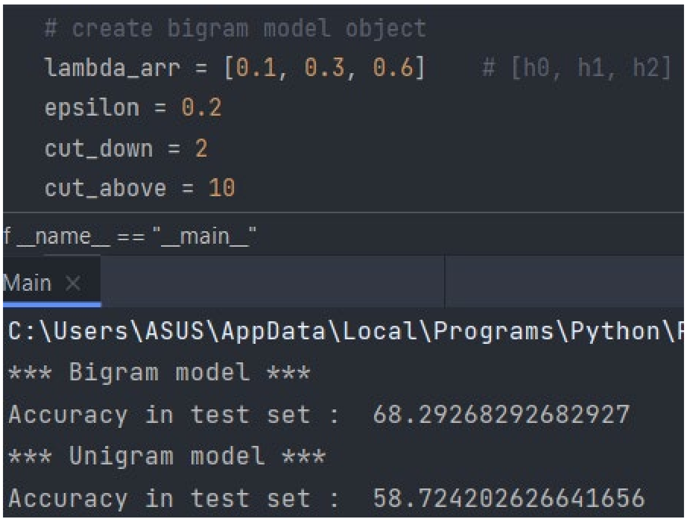
```
    # create bigram model object
    lambda_arr = [0.6, 0.3, 0.1]    # [h0, h1, h2]
    epsilon = 0.8
    cut_down = 2
    cut_above = 10
f __name__ == "__main__"
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\
*** Bigram model ***
Accuracy in test set :  64.25891181988743
*** Unigram model ***
Accuracy in test set :  56.94183864915572
```

این قسمت را به ۳ بخش تقسیم کردیم (اپسیلون 0.2, 0.5, 0.8) و سپس در هر بخش ۴ آرایه‌ی لاندای متفاوت را تست کردیم که یکی با مقادیر صعودی، دیگری با مقادیر نزولی و دیگری با مقادیر مساوی و یکی با مقادیر [0.4, 0.4, 0.2] و دیدیم که در حالت صعودی (حالتی که بایگرام وزن بیشتری دارد) و اپسیلون مقدار کم تری (0.2) دارد بهترین نتیجه را گرفتیم.

بهترین دقت دست‌یافته و پارامترها در آن:

- Epsilon: 0.2
  Lambda: [L1:0.1 , L2:0.3 , L3:0.6]
  Cut_down = 2
  Cut_above = 10



```
    # create bigram model object
    lambda_arr = [0.1, 0.3, 0.6]    # [h0, h1, h2]
    epsilon = 0.2
    cut_down = 2
    cut_above = 10
f __name__ == "__main__"
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\F
*** Bigram model ***
Accuracy in test set :  68.29268292682927
*** Unigram model ***
Accuracy in test set :  58.724202626641656
```