

گزارش کار

کد تنها یک کلاس به نام bigram دارد که در ابتدا با استفاده از تابع read دو دیکشنری bidict, unidict از پیکره های داده شده می سازد.

برای محاسبه ی backoff model به احتمال bigram و unigram هر کلمه نیاز داریم که به صورت دو تابع تعریف شده اند و در آخر با تابع back off model طبق فرمول به محاسبه ی احتمال back off می پردازیم.

```
back_off_word_probability = (self.bi_word(preWord, word) * landa3) + (self.uni_word(word) * landa2) + (
    landa1 * epsilon)
back_off_sentence_probability *= back_off_word_probability
```

سپس در کل برای بیت شعر پیکره ی تست احتمال هر شاعر را حساب کرده و بیشترین را با جواب درست مقایسه می کنیم تا تعداد جواب های درست به دست آید و precision نیز نسبت جواب های درست به کل پیکره ی تست است.

دقت برنامه ارتباط زیادی با پارامترهای لاندای و اِپسیلون دارد:

1.

```
landa1 = 0.2
landa2 = 0.3
landa3 = 0.5
epsilon = 0.3
```

```
1866 Were Correct Answers
0.6785454545454546 precision
```

2.

```
landa1 = 0.2
landa2 = 0.3
landa3 = 0.5
epsilon = 0.1
```

```
1972 Were Correct Answers
0.7170909090909091 precision
```

3.

```
landa1 = 0.1
landa2 = 0.2
landa3 = 0.7
epsilon = 0.1
```

```
2065 Were Correct Answers
0.7509090909090909 precision
```

.4

```
landa1 = 0.01  
landa2 = 0.09  
landa3 = 0.9  
epsilon = 0.1
```

```
2203 Were Correct Answers  
0.8010909090909091 precision
```

.5

```
landa1 = 0.01  
landa2 = 0.09  
landa3 = 0.9  
epsilon = 0.01
```

```
2299 Were Correct Answers  
0.836 precision
```

همانطور که ملاحظه می شود با افزایش وزن مدل بایگرام (زیاد کردن landa3 و کم کردن landa1 landa2 epsilon) دقت برنامه بیشتر می شود. علت آن می تواند این باشد که در شعر ترکیبات دوکلمه ای هر بیت با شانس زیادی منجر به فهمیدن شاعر خواهد شد و تعداد تکرار هر کلمه به تنهایی اعتبار زیادی ندارد. همچنین اگر کلمه ای در شعرهای پیکره ی آموزشی شاعری نیامده، احتمال کمی دارد که آن شاعر در شعرهای دیگرش که در پیکره ی ما نیست استفاده کرده باشد.