به نام خدا

پروژه سوم

این شعر از کیست؟

دریا زارع مهذبیه

۹۷۳۱۰۸۶

فهرست

مراحل الگریتم:

ابتدا با توجه به داده train وردی Unygram و bygram را برای هر یک از شاعد ها میشازیم(که یم دیکشتری با کلید"لغت" و مقدار"تعداد تکرار" می‌باشد.

باید داده های unigram را تمیز کنیم بنابراین لغاتی که کمتر از ۲ باز تکرار شده اند را از لغتنامه حذف میکنیم.

سپس با توجه به الگریتم backoff احتمال وقوع هر کلمه را برای داده تست حساب میکنیم و در نهایت شاعر با احتمال بیشتر که یک شاعر شعری را سروده به عنوان نتیجه برمیگردانیم.

توابع:

```
def backoff(w1,w2,unygram,bygram,lambda1,lambda2,lambda3,beta):
    resault=0
    ww=w1+' '+w2

    if ww in bygram and w1 in unygram:

        resault = lambda1*bygram.get(ww)/unygram.get(w1)
    if w1 in unygram:
        resault+=lambda2*unygram.get(w1)/len(unygram.keys())
    resault+=lambda3*beta
    return resault
```

تابع ای که برای هر کلمه محاسبه میکند احتمال وقوعش را در unygramوbygram ورودی.

```
def howMuchIn(poem,unigram,bygram,lambda1,lambda2,lambda3,beta):
    poem=poem.split()
    prob=0

    for i in range(len(poem)):
        if i==len(poem)-1:
            prob+=backoff(poem[i],'<>',unigram,bygram,lambda1,lambda2,lambda3,bet
a)
        else:
            prob+=backoff(poem[i],poem[i+1],unigram,bygram,lambda1,lambda2,lambda
3,beta)

    return prob
```

مجاسبه میکند احتمالی که یک شعر برای شاعری که unygramو bygram آن را ورودی گرفته.

```
def whichOne(poem,lambda1,lambda2,lambda3,beta):
    probF = howMuchIn(poem,ferdowsi,ferdowsi2,lambda1[0],lambda2[0],lambda3[0],be
ta[0])
    probH = howMuchIn(poem,hafez,hafez2,lambda1[1],lambda2[1],lambda3[1],beta[1])
    probM = howMuchIn(poem,molavi,molavi2,lambda1[2],lambda2[2],lambda3[2],beta[2
])
    m=max(probF,probH,probM)
    #print(probF,probH,probM)
    if probF==m:
        return '1'
    if probH==m:
        return '2'
    return '3'
```

با توه به احتمالات حساب شده تصمیم میگیرد کدام شاعد شعر را سروده است.

```
def maxLambda(lambda1,lambda2,lambda3,beta):
    val=[[0,0,0],[0,0,0],[0,0,0]]
    for i in range(len(poems)):
        temp = whichOne(poems[i],lambda1,lambda2,lambda3,beta)

        if poets[i] == temp:
            val[1][int(temp)-1]+=1
        else:
            val[2][int(temp)-1]+=1
        val[0][int(poets[i])-1]+=1

    return val
```

این تابع برای پیدا کردن لامبدا مناسب بکار می‌رود.

به این صورت که برای لامبدا ورودی محاسبه میکند از هر شاعر چند تا را درست تشخیص داده و چند تا را غلط در نتیجه میتوان دقت هر سری لاندا را برای هر شاعد مشاهدا کرد و مقادیر را بهبود داد.

تحلیل:

با توجه به اندازه لغت نامه ها داده ها بالانس نیستند و دیتا حافظ بیشتر از بقیه است و کمترین را فردوسی دارد. درنتیجه بتا را باید برای حافظ مقدار خیلی کمی درنظر بگیریم و برای فردوسی بیشتر باشد.

درکل مقدار لاندا ای که در بتا ضرب میشود(در صورت پروژه لاندا۳) مقدار خیلی کمی باید باشد زیرا دقت را کاهش میدهد. اما اگر تعداد غلط هایی که یک شاعر داشت زیاد بود باید ضریب بایگرام افزایش یابد.

ادامه تحلیل ها در کد.