Task 1. Compare Genia Corpus with AGAC Corpus (TTR)

Github link: https://github.com/Allen-ZKW/NLP_HZAU/tree/Task1

Date: Kewei Zhao
Date: 2021-3-29

Abstract

In this task, we calculate TTR of different Corpuses. By comparing their TTR and other parameters, we try to discover and analyze the difference of these two corpuses.

Principle

$$a \leftarrow count \quad of \quad unique \quad words \quad in \quad text$$
$$b \leftarrow count \quad of \quad all \quad words \quad in \quad text$$
$$TTR = \frac{a}{b}$$

This parameter will reflect the information density corpus.

Measure

1. Download Corpus

AGAC corpus: http://pubannotation.org/collections/AGAC
Genia corpus: http://www.nactem.ac.uk/GENIA/current/GENIA-corpus/Part-of-speech/GENIAcorpus3.02p.tgz

2. Clean the Data(Linux)

We try to just save number and English words, so other kinds of characters may play act as a divider between the two words. Capital letters would not change the meaning of words, so we change all letters to lowercase. After that, we find '[' and ']' still exit in the text file, so we try to delete the punctuations.

```
 cat file | tr -cs "[:alnum:]" "\n" | tr "[:upper:]" "[lower]" | tr -d "
[:punct:]"
```

3. Analyze the Data(Python)

1) Import Essential Modules

```
from random import shuffle
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

2) Import Data

```
def importdata(dirpath):

    genia_raw = []
    with open (dirpath + '/genia_pure.txt') as f1:
        for le in f1:
            l = le.strip()
            genia_raw.append(l)

    AGAC_raw = []
    with open (dirpath + '/train_pure.txt') as f2:
        for le in f2:
            l = le.strip()
            AGAC_raw.append(l)
    return genia_raw,AGAC_raw
```

3) Sampling

After Shuffling the list of words, we use the elements in the head of list as a sample.

size of AGAC/size of GENIA ≈ 1/10 → size of sample = 1/10

```
def sampling(genia_raw,AGAC_raw):

    shuffle(genia_raw)
    genia_sample = genia_raw[0:30000]

    shuffle(AGAC_raw)
    AGAC_sample = AGAC_raw[0:3000]

    return genia_sample,AGAC_sample
```

4) Calculate TTR

Change the type sample to set in order to get the unique words, then, calculate the TTR of different sample.

```
def TTR(genia_sample,AGAC_sample):
    genia_TTR = len(set(genia_sample))/30000
    AGAC_TTR = len(set(AGAC_sample))/3000

    return genia_TTR,AGAC_TTR
```

5) Loop Call These Functions

Call these functions for 1000 times to get data of different random samples.

```
dirpath = 'D:/junior_n/NLP/task_1'
genia_raw,AGAC_raw = importdata(dirpath)
genia_TTR = [0]*1000
AGAC_TTR = [0]*1000
for i in range(1000):
    genia_sample,AGAC_sample = sampling(genia_raw,AGAC_raw)
    genia_TTR[i],AGAC_TTR[i] = TTR(genia_sample,AGAC_sample)
```

5) Normal Distribution Test

```
stats.normaltest (genia_TTR, axis=0)
stats.normaltest (AGAC_TTR, axis=0)
```

6) Variance Homogeneity Test

Because Levene-test does not require data with normal distribution, so we can choose to do normal distribution test or not in the last step.

```
statistic, pvalue = stats.levene(genia_TTR,AGAC_TTR)
```

7) T-test

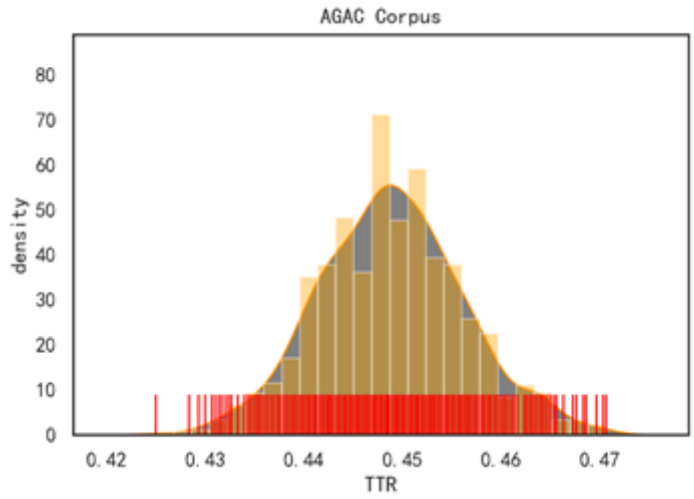
```
if pvalue < 0.05:
    stats.ttest_ind(genia_TTR, AGAC_TTR, equal_var=False)
else:
    stats.ttest_ind(genia_TTR, AGAC_TTR, equal_var=True)
```

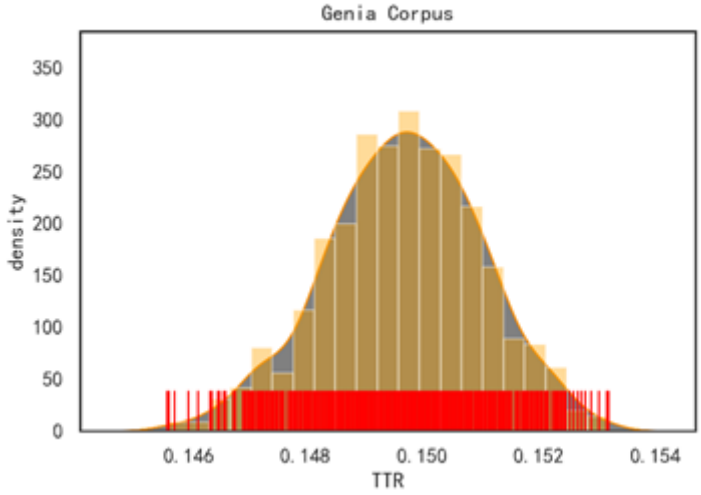
8) Visualization

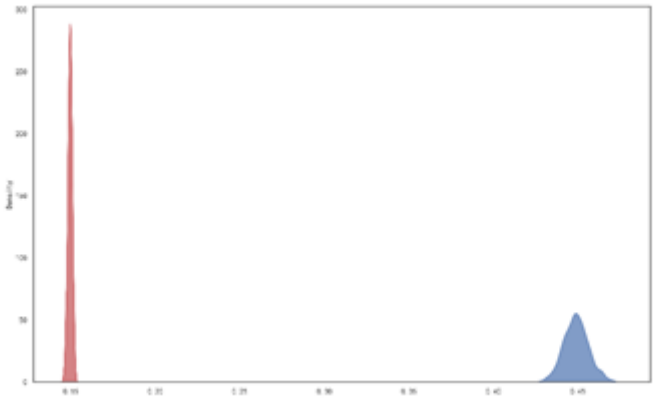
```
def KdePlot(x,name):

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False

    plt.figure()
    sns.set(style='white',
            font = 'SimHei')
    sns.distplot(x,
                 color='orange',
                 kde=True,
                 hist=True,
                 rug=True,
                 kde_kws = {"shade": True,
                            "color": 'darkorange',
                            # 'linewidth': 1.0,
                            'facecolor': 'gray'},
                 rug_kws = {'color': 'red',
                            'height': 0.1})
                 # vertical = True)

    plt.title(name)
    plt.xlabel('TTR')
    plt.ylabel('density')
    plt.savefig(name+'.png', dpi=300)
    plt.show()

def boxplot(genia_TTR,AGAC_TTR):
    plt.boxplot((genia_TTR,AGAC_TTR),labels=('genia_TTR','AGAC_TTR'))
    plt.savefig('box.png', dpi=300)

KdePlot(genia_TTR,'Genia Corpus')
KdePlot(AGAC_TTR,'AGAC Corpus')
boxplot(genia_TTR,AGAC_TTR)
```

Result

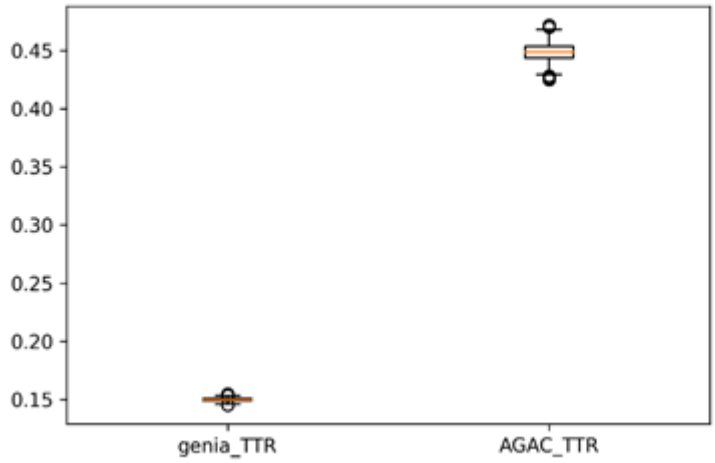
Density Curve







Box Plot



Discussion

According to t-test, AGAC-TTR is remarkably higher than Genia's. Which may partly illustrate that the information in AGAC corpus is more abundant. However, we also count the sum of words in these two corpuses, the scale of Genia is nearly 10 times larger than AGAC. This shows that although AGAC corpus' density of information is higher than Genia corpus', Genia corpus does better than AGAC corpus in aspect of sum of information.

Genia corpus, a semantically annotated corpus of biological literature, is being compiled and annotated in the scope of GENIA project. It is aiming at providing high quality reference materials to let NLP techniques work for informatics and at providing the gold standard for the evaluation of text mining systems.

AGAC corpus is a customized corpus for mining functions caused by mutations. "Annotation of Genes with Alteration-Centric function changes."

We think different function, aims and size of field of these two corpuses lead to the difference in TTR and size of the text file: Genia corpus try to cover biological literature in all aspects, while AGAC corpus focus on biological literature which is about mining functions caused by mutations.

So, AGAC corpus may do a better job in mining functions caused by mutations. And Genia corpus can play a part in all aspects of biology NLP problems.