Analysis Assignment-3

Q1 Static Score

As we could see on different values of r that how much to send to high list.Output was behaving in different ways.

It was seen that best outcome came when r is near 20 or 25.As the mean of length of posting list of all terms was near about 20.Considering some False positive we could also run on r=25.

As can be seen in given query.

```
querylist=[]
lemmatizer=WordNetLemmatizer()
for word in result:
    if word.isdecimal()==True:
        word=num2words(word)
    querylist.append(lemmatizer.lemmatize(word))
print(querylist)
```

```
I am looking for publically accessible sources of data depicting brain and neuron functions.  Especially interesting would b
e volumetric data from brain scans, electromicographs, and so forth.  Please email me and let me know if you know of such.
Number of documents to retrive
10
['looking', 'publically', 'accessible', 'source', 'data', 'depicting', 'brain', 'neuron', 'function', 'especially', 'interes
ting', 'would', 'volumetric', 'data', 'brain', 'scan', 'electromicographs', 'forth', 'please', 'email', 'let', 'know', 'kno
w']
```

```
rank(k)
```

```
1008 : 1.0738742724290513
13011 : 1.0669519141006238
13277 : 0.7920841363967563
13012 : 0.5571023136207442
13075 : 0.4853533073379789
651 : 0.41799917574875545
14591 : 0.3754413187145786
1009 : 0.3734288442329891
4533 : 0.3689030243559645
1692 : 0.36439134129337863
```

Here when kept r=20 .Mean length We are getting better result than r=15.Hence this is analysed to be better

Q2

res is original list of url.

X is new sorted list of urls.

```
In [5]:
X=sorted(res,reverse=True)
```

```
In [6]:
print(res) #res:list of list
```

```
[['0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:999 12:0 13:110 14:5 15:1114 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.972391 21:20 22:0 23:5 24:0 25:25 26:1 27:0 28:0 29:0 30:1 31:12 32:0 33:4 34:0 35:16 36:6.666667 37:0 38:1.666667 39:0 40:8.333333 41:20.222222 42:0 43:2.888889 44:0 45:37.555556 46:0.02002 47:0 48:0.045455 49:0 50:0.022442 51:0.001001 52:0 53:0 54:0 55:0.000898 56:0.012012 57:0 58:0.036364 59:0 60:0.014363 61:0.006673 62:0 63:0.015152 64:0 65:0.007481 66:0.00002 67:0 68:0.000239 69:0 70:0.00003 71:77.577533 72:0 73:30.667985 74:0 75:90.53171 76:5.52713 77:0 78:0 79:0 80:5.526745 81:57.882066 82:0 83:18.750101 84:0 85:66.125373 86:25.859178 87:0 88:10.222662 89:0 90:30.177237 91:525.177766 92:0 93:60.031269 94:0 95:675.850674 96:1 97:0 98:0 99:0 100:1 101:0.875901 102:0 103:0.66135 104:0 105:0.864571 106:28.756809 107:0 108:3.274639 109:0 110:28.985515 111:-17.640291 112:-29.251906 113:-20.596041 114:-31.107208 115:-17.519629 116:-19.440921 117:-31.580405 118:-24.146168 119:-33.960286 120:-19.161514 121:-16.596977 122:-31.750477 123:-21.267965 124:-33.908554 125:-16.503638 126:2 127:27 128:2 129:9 130:124 131:4678 132:54 133:74 134:0 135:0 136:0 ', ''], ['0 qid:4 1:3 2:0 3:3 4:0 5:3 6:1 7:0 8:1 9:0 10:1 11:1561 12:2 13:34 14:10 15:1607 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.972391 21:116 22:0 23:5 24:0 25:121 26:25 27:0 28:1 29:0 30:27 31:46 32:0 33:2 34:0 35:47
```

Max dcg is calculated as follows

```
In [8]:
import math
def Cal_Max_dcg(res):
    dcg=0
    for i in range(len(res)):
        #print(i)
        rel=int(res[i][0][0])
        #print(rel)
        dcg+=(pow(2,rel)-1)/(math.log(i+2,2))
    #print(dcg)
    return dcg

val=Cal_Max_dcg(X)
print("Max dcg.....",val)
```

```
Max dcg..... 28.98846753873482
```

All permutation for number of files possible

```
In [29]:
c=0
for i in range(c0+1):
    c=c+permutation(c0,i)
print("Total number of files with max dcg:",math.factorial(c1)*math.factorial(c2)*math.factorial(c3)*c)
```

```
Total number of files with max dcg: 5.407613242151098e+121
```

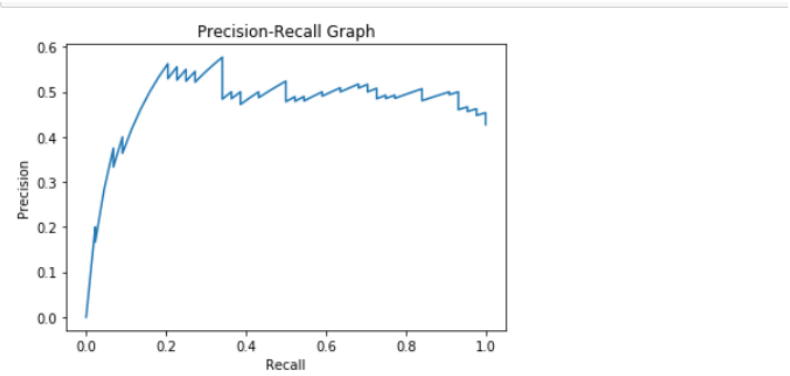
Ndcg calculated at 50 and wholedataset

```
ndcg 0.35612494416255847
NDCG at 50 ..... 0.35612494416255847
ndcg 0.5784691984582591
NDCG at wholedata..... 0.5784691984582591
```

After sorting on basis of 75th column that is sum of tf idf and assuming non zero is relevant.We have total rel docs as follows.

```
total rel docs 44
```

Precision Recall Graph is built.By calculating precision and recall as the document is retrieved.



As we can infer from graph recall is always increasing with time or its non decreasing in nature and finally reaches 1.

Precision is initially increases and then jitters can be seen .

Hence the relation between precision and recall is analysed from graph above.

Q3

a) Explain the relationship between ROC curve and PR curve.

Relation between Roc curve and PR curve is such that for a given dataset of positive and negative samples , there exist a one to one correspondence b/w ROC curve & PR curve , if Recall > 0.

As point in a ROC defines unique matrix as TPR and FPR consist of all TP,FP,TN and FN for a fixed dataset .In PR we ignore TN ,this might seem like we may have different confusion matrix .But with given other three enteries for fixed data TN is also uniquely identified.This is true only when TPR is not 0.If recall is 0 we cant measure FN.Hence no uniqueness left.

Hence Confusion matrix is in one to one relation with ROC and PR curve.This impliesROC and FP are in one to one relation with each other.

b) Prove that a curve dominates in ROC space if and only if it dominates in PR space.

We will prove the above in two claims stated below:

Consider two curves curve 1 and curve 2 with point B on curve 1 and point A on curve 2.
Claim 1:If a curve dominates in ROC space then it dominates in PR space
Proof by Contradiction
Let us assume Curve 1 dominate in ROC Space and do not dominate in PR space.

Since curve 1 does not dominate in PR curve then there will be one point A on curve 2 whose precision is greater than one point B on curve 1 point on same Recall.
i.e PRECISION(A) > PRECISION(B) at TPR(A) = TPR(B).----eq1
FPR(B) = False Positive(B) / Total Negatives
FPR(A)= Fasle Positive(A)/ Total Negatives
Now curve 1 dominates curve 2 in ROC therefore FP(A) ≥ FP(B) because Total Negatives are constant.
PRECISION(A) = TP/(FPA + TP) PRECISION(B) = TP/(FPB + TP)
Now here we can see that PRECISION(A)<=PRECISION(B). but this contradicts our assumption(eq1)
Therefore Claim 1 is true.

Claim 2 : If a curve dominates in PR space then it dominates in ROC space.
Let us consider that curve 1 dominates in PR space but does not dominate in ROC space.
Proof by Contradiction
AS curve I dominates in PR space we know that PRECISION(A) ≤ PRECISION(B)->(I) , but RECALL(A)=RECALL(B).
PRECISION(A) = TP /(TP + FP(A))
PRECISION(B) = TP/(TP + FP(B))
By (I) we can say that FP(A)>=FP(B)
Now
FPR(A) = FPA/ Total Negatives
FPR(B) = FPB/ Total Negatives
From above equation we can say that FPR(A) ≥ FPR(B) and this contradicts our original assumption that FPR(A) < FPR(B). Hence it also dominate in roc.
Hece proved roc dominate if and only if pr dominates.

Q3:]

It is incorrect to interpolate between points in PR space. When and why does this happen? How will you tackle this problem?

It is incorrect to interpolate between points in precision recall space because as FP is considered in Precision and FN is considered in Recall and hence in different levels recall varies but it Precision does not necessarily change. In order to tackle this problem, it is better to translate the points to ROC curve(convex hull in ROC space) and again from ROC curve to PR space. Now, this curve PR space excludes exactly those points which are beneath the convex hull in ROC space.

To interpolate between two points A and B, we must interpolate between their counts TP(A), TP(B), FP(A) and FP(B).
For this we find local skew defined as:
FP(B)-FP(B)/TP(A)-TP(B)
Now, for all x where 1<=x<=TP(A)-TP(B), we create new points TP(A)+x and calculate corresponding FP by increasing it linearly by local skew. The intermediate points are:
(TP/Total positives, (TP(A)+x)/ ((TP(A)+x+ FP(A)+local skew)

Hence the problem can be solved in this way by newly constructed PR space curve suitable for interpolation.