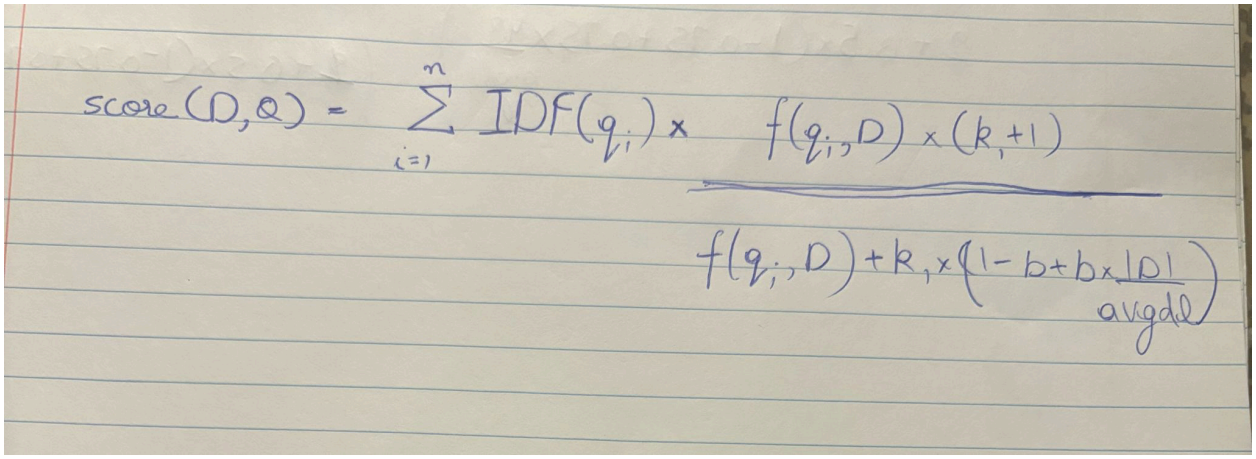
Aaryan Bhagat abhag017 862468325

1

I have assumed constant b to be is x and 0.75 and for k1, since its a short collection, k1 is on the lower side which is 0.5.

First the formula



For query 1

$k_1 = 0.5$

$b = 0.75$

Q_1 = university riverside (Assuming Case Insensitive)

q_1 q_2

$IDF(q_1) = \ln\left(\frac{10^4 - 500 + 0.5}{500 + 0.5} + 1\right)$ $f(q_1, D) = 2$

$IDF(q_2) = \ln\left(\frac{10^4 - 100 + 0.5}{100 + 0.5} + 1\right)$ $f(q_2, D) = 1$

$|D| = 48$ [Calculated by tokenizing]

$$IDF(q_1) \times \frac{2 \times (0.5+1)}{2 + 0.5 \times \left(1 - 0.75 + 0.75 \times \frac{48}{\text{avgdl}}\right)} + IDF(q_2) \times \frac{1 \times (0.5+1)}{1 + 0.5 \times \left(1 - 0.75 + 0.75 \times \frac{48}{\text{avgdl}}\right)}$$

Q_2: diverse university

q_1 q_2

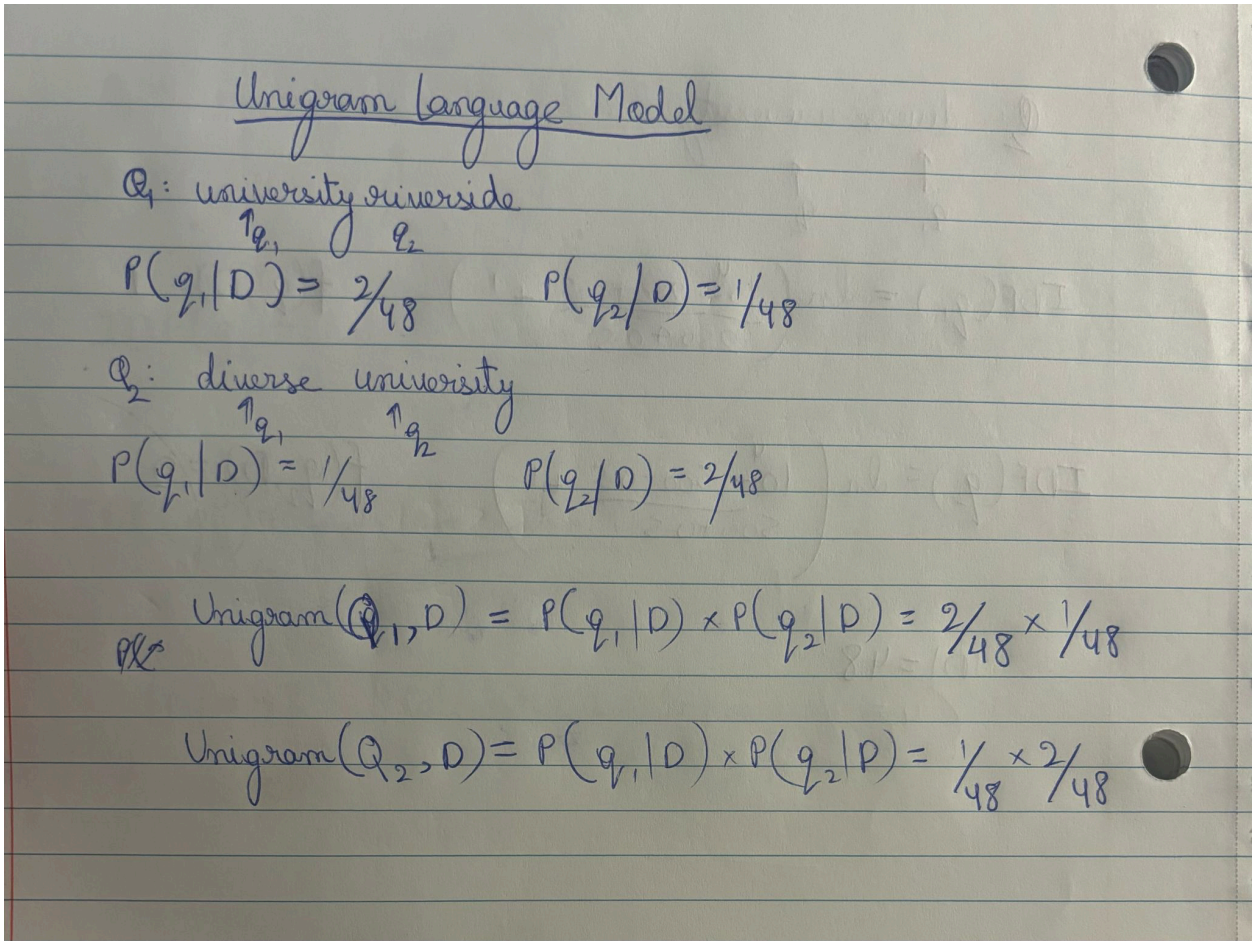
$IDF(q_1) = \ln\left(\frac{10^4 - 250 + 0.5}{250 + 0.5} + 1\right)$ $f(q_1, D) = 1$

$IDF(q_2) = \ln\left(\frac{10^4 - 500 + 0.5}{500 + 0.5} + 1\right)$ $f(q_2, D) = 2$

$|D| = 48$

$$IDF(q_1) \times \frac{1 \times (0.5+1)}{1 + 0.5 \times \left(1 - 0.75 + 0.75 \times \frac{48}{\text{avgdl}}\right)} + IPF(q_2) \times \frac{1 \times (0.5+1)}{2 + 0.5 \times \left(1 - 0.75 + 0.75 \times \frac{48}{\text{avgdl}}\right)}$$

For Unigram it is very straightforward to compute



2

Program code for pagerank

```
import numpy as np



def pageRank(M, num_iter: int = 100, d: float = 0.85):
    N = M.shape[1]
    v = np.ones(N) / N
    M_hat = (d * M + (1 - d) / N)
    count = 0
    while True:
        count = count + 1
        v_new = M_hat @ v
        print("Iteration ", count)
        print(v)
```

```
        if(np.linalg.norm(v_new - v) <= 0.01):
            break
        v = np.copy(v_new)
    return v



M = np.array([[0, 1, 0, 0, 0],
              [1/3, 0, 1/2, 0, 1],
              [0, 0, 0, 0, 0],
              [1/3, 0, 0, 0, 0],
              [1/3, 0, 1/2, 0, 0]])
v = pageRank(M, 100, 0.85)
# print(v)
```

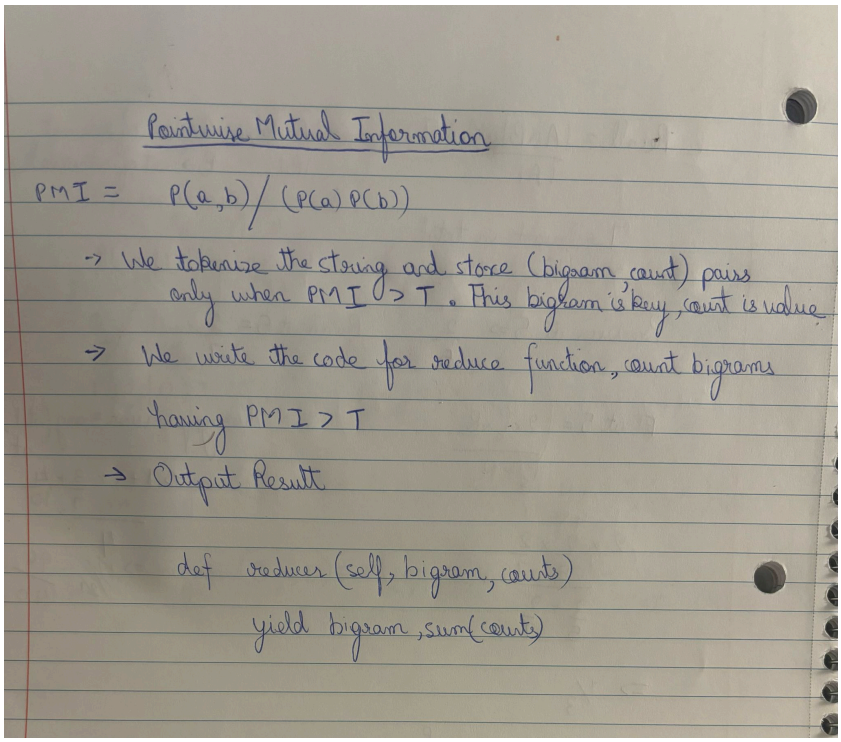
Output

```
Iteration  1
[0.2 0.2 0.2 0.2 0.2]
Iteration  2
[0.2       0.34166667 0.03       0.08666667 0.17166667]
Iteration  3
[0.31531667 0.24023333 0.0249     0.08156667 0.09431667]
Iteration  4
[0.22688833 0.20278139 0.02269    0.11202972 0.12261222]
Iteration  5
[0.19297423 0.19875872 0.02061005 0.08489508 0.09453833]
Iteration  6
[0.1866982  0.16154617 0.01775329 0.07242932 0.0811886 ]
Iteration  7
[0.15290272 0.14504175 0.01558847 0.06848629 0.07603144]
Iteration  8
[0.137027   0.12831578 0.01374152 0.05706396 0.06368905]
Iteration  9
[0.12106353 0.11079528 0.01199512 0.05081944 0.05665958]
Iteration  10
[0.10471598 0.09809989 0.01053999 0.04484132 0.04993925]
Iteration  11
[0.092629   0.08584148 0.00924409 0.03891362 0.04339311]
Iteration  12
[0.08106589 0.07515841 0.00810064 0.03434552 0.03827426]
Iteration  13
[0.07099299 0.06605291 0.00710834 0.03007701 0.03351978]
Iteration  14
[0.0623775  0.05786007 0.00623253 0.02634721 0.02936826]
Iteration  15
[0.05464663 0.05075104 0.00546557 0.02313919 0.02578802]
Iteration  16
[0.04793209 0.04451961 0.00479371 0.02027692 0.02259979]
```

So converging after 16 iterations

3

For map reduce this is the steps of how we will devise an algorithm to solve a bigram having a condition of pointwise mutual information



If we use Combiner here, it can optimize the reduce function by doing certain data preprocessing in different shards where data is stored.

4

All the formulas are applied below, the example mentioned for high recall is google map which will certainly give the right answers as well as other answers.

For high precision it is the normal google search

Recall = $\frac{|A \cap B|}{|A|}$ Precision = $\frac{|A \cap B|}{|B|}$ A → set of relevant
B → set of retrieved

7 relevant in total

r r x x x x r x x r

Precision at 5 ⇒ $\frac{2}{5}$ Recall at 5 = $\frac{2}{7}$

F1 at 5 = $\frac{2 \times P_5 \times R_5}{P_5 + R_5}$

$\frac{2 \times \frac{2}{5} \times \frac{2}{7}}{\frac{2}{5} + \frac{2}{7}}$

⇒ $1/3$

Average precision = avg (at 1, at 2, at 7, at 8)

⇒ $\frac{1 + 1 + \frac{3}{7} + \frac{4}{10}}{4}$

⇒ $\frac{99}{140}$

$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$

p = 5

$1 + \sum_{i=2}^{5} \frac{rel_i}{\log_2 i} \Rightarrow 1 + \frac{1}{\log_2 2} + 0 + 0 + 0$

⇒ 2

Higher Recall when we need all ~~correct~~ relevant pages to be present
Example is google maps which will show all correct + more
Higher Precision is when we have no non relevant in output
Example is google search engine