**מטלה 2 - 404 Brain Not Found**

**שאלה 1:**

We are given that we retrieved 15 documents, 5 of which are relevant and 10 are irrelevant. We know that there are additional 20 relevant documents in the collection. Therefore, we deduce the following:

* True Positives (TP): 5
* False Positives (FP): 10
* False Negatives (FN): 20

Using these values:

1. Recall = TP / (TP + FN) = 5/25= 0.2 = 20%
2. Precision = TP / (TP + FP) = 5/15 =1/3 = 33.33%

**שאלה 2 סעיף א:**

תזכורת לחישוב Recall ו- Precision:

Recall - כמות המסמכים הרלוונטיים שהוחזרו​ / סך כל המסמכים הרלוונטיים.

Precision - כמות המסמכים הרלוונטיים שהוחזרו​ / סך המסמכים שהוחזרו.

נבנה את הטבלה עבור :**Engine 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Precision** | **Recall** | **R/N** | **Document** |
| 1/1 = 1 = 100% | 1/10 = 0.1 = 10% | R | 1 |
| 1/2 = 0.5 = 50% | 1/10 = 0.1 = 10% | N | 2 |
| 2/3 = 0.67 = 67% | 2/10 = 0.2 = 20% | R | 3 |
| 3/4 = 0.75 = 75% | 3/10 = 0.3 = 30% | R | 4 |
| 3/5 = 0.6 = 60% | 3/10 = 0.3 = 30% | N | 5 |
| 3/6 = 0.5 = 50% | 3/10 = 0.3 = 30% | N | 6 |
| 4/7 = 0.57 = 57% | 4/10 = 0.4 = 40% | R | 7 |
| 4/8 = 0.5 = 50% | 4/10 = 0.4 = 40% | N | 8 |
| 5/9 = 0.55 = 55% | 5/10 = 0.5 = 50% | R | 9 |
| 6/10 = 0.6 = 60% | 6/10 = 0.6 = 60% | R | 10 |

נבנה את הטבלה עבור :**Engine 2**

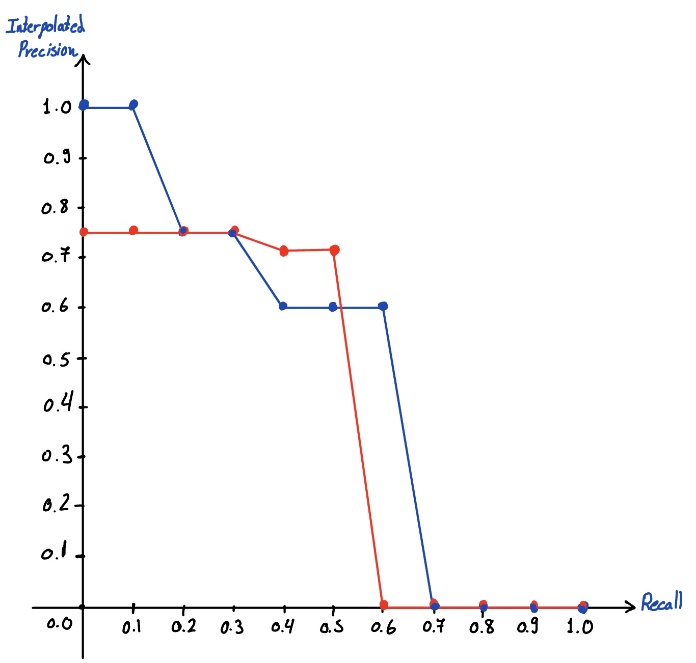
|  |  |  |  |
| --- | --- | --- | --- |
| **Precision** | **Recall** | **R/N** | **Document** |
| 0/1 = 0 = 0% | 0/10 = 0 = 0% | N | 1 |
| 1/2 = 1 = 50% | 1/10 = 0.1 = 10% | R | 2 |
| 2/3 = 0.67 = 67% | 2/10 = 0.2 = 20% | R | 3 |
| 3/4 = 0.75 = 75% | 3/10 = 0.3 = 30% | R | 4 |
| 3/5 = 0.6 = 60% | 3/10 = 0.3 = 30% | N | 5 |
| 4/6 = 0.67 = 67% | 4/10 = 0.4 = 40% | R | 6 |
| 5/7 = 0.714 = 71.4% | 5/10 = 0.5 = 50% | R | 7 |
| 5/8 = 0.625 = 62.5% | 5/10 = 0.5 = 50% | N | 8 |
| 5/9 = 0.555 = 55.5% | 5/10 = 0.5 = 50% | N | 9 |
| 5/10 = 0.5 = 50% | 5/10 = 0.5 = 50% | N | 10 |

נציג טבלה של 11 נקודות אינטרפולציה:

כדי לחשב את הInterpolated Precision -: עבור כל נקודת Recall נבדוק את כל ערכי ה- Precision בנקודה זו ומעבר לה (גדול או שווה) ונשמור את הערך הגבוה ביותר של Precision שנמצא. (ניקח את 11 הנקודות מ- 0.0 עד 1.0).

|  |  |  |
| --- | --- | --- |
| **Engine 2** | **Engine 1** | **Recall** |
| 75% | 100% | 0.0 |
| 75% | 100% | 0.1 |
| 75% | 75% | 0.2 |
| 75% | 75% | 0.3 |
| 71.4% | 60% | 0.4 |
| 71.4% | 60% | 0.5 |
| 0% | 60% | 0.6 |
| 0% | 0% | 0.7 |
| 0% | 0% | 0.8 |
| 0% | 0% | 0.9 |
| 0% | 0% | 1.0 |

נציג הגרף של 11 נקודות אינטרפולציה:



נחשב F-Measure עם 0.75 = β:

F-Measure = (1 + β²) \* (P \* R) / (β² \* P + R)

**For Engine 1:**

F-Measure = (1 + 0.75^2) \* (0.6 \* 0.6) / (0.75^2 \* 0.6 + 0.6)

F- Measure = **60%**

**For Engine 2:**

F- Measure = (1 + 0.75^2) \* (0.5 \* 0.5) / (0.75^2 \* 0.5 + 0.5)

F- Measure = **50%**

🡨המנוע 1 יותר טוב מהמנוע 2 כי יש לו F-Measure יותר טובה.

**שאלה 2 סעיף ב:**  
**א.** אחרי מחיקת ה- stop words והשמטת (ly, ed, ing, s) נבנה טבלת ה- inverted index:

|  |  |
| --- | --- |
| **After removing the stop words, (ly, ed, ing, s)** | **Document** |
| dog, love, music, listen, Roll, Stone | D1 |
| Information, Retrieval, course | D2 |
| dog, can, roll, love, throw, Stone | D3 |
| help, pick, up, Stone, road | D4 |

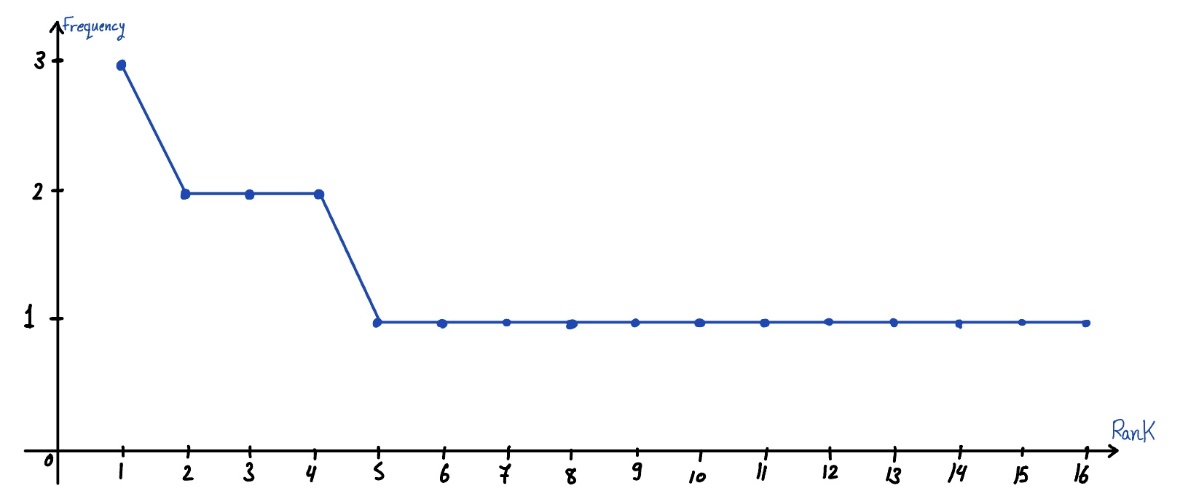
|  |  |
| --- | --- |
| D1, D3 | love |
| D1 | music |
| D1 | listen |
| D1, D3 | Roll |
| D1, D3, D4 | Stone |
| D2 | Information |
| D2 | Retrieval |
| D2 | course |
| D1, D3 | dog |
| D3 | can |
| D3 | throw |
| D4 | help |
| D4 | pick |
| D4 | up |
| D4 | road |

**ב.** נבנה את ה- inverted index עבור כל term:

**ג.** נבנה את הטבלה:

|  |  |  |
| --- | --- | --- |
| **Rank** | **TF** | **Word** |
| 1 | 3 | stone |
| 2 | 2 | dog |
| 3 | 2 | love |
| 4 | 2 | roll |
| 5 | 1 | music |
| 6 | 1 | listen |
| 7 | 1 | Information |
| 8 | 1 | Retrieval |
| 9 | 1 | course |
| 10 | 1 | can |
| 11 | 1 | throw |
| 12 | 1 | help |
| 13 | 1 | pick |
| 14 | 1 | up |
| 15 | 1 | road |

TF – נעבור על כל המילים מכל המסמכים ונספור את מספר הפעמים שכל מילה מופיעה.

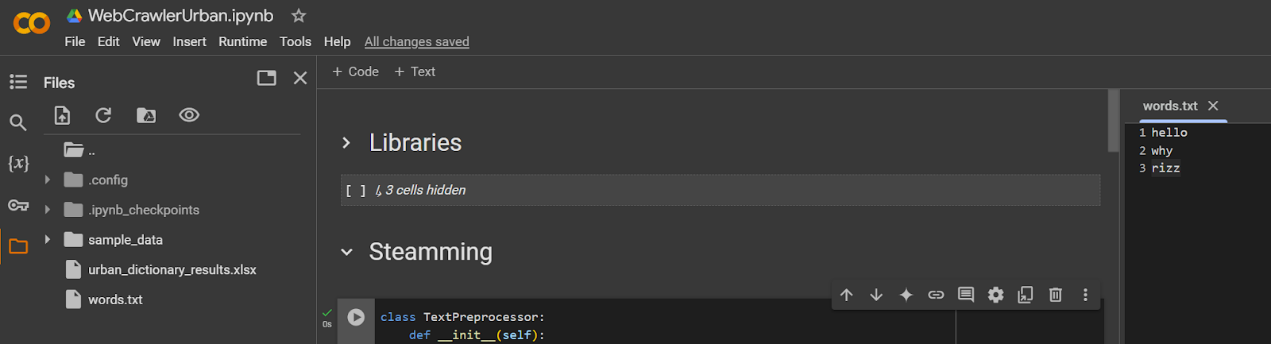
נבנה את הגרף:

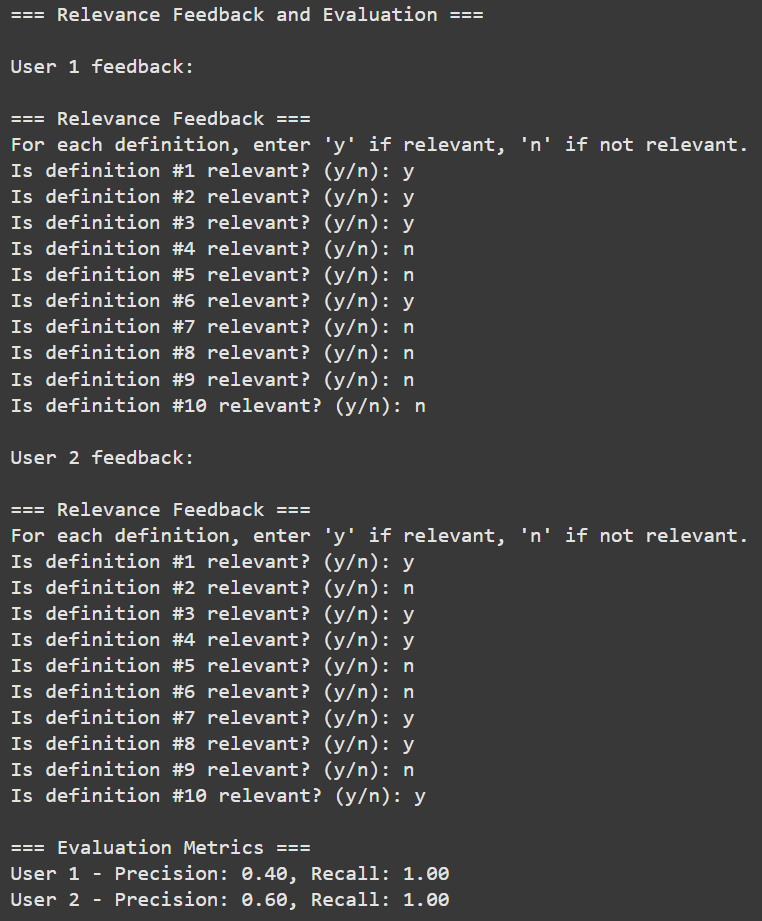
🡨 אז גילינו שהם מתנהגים לפי חוק Zipf. (במיוחד בגלל הירידה החדה בתדירות המילים בין הדרגות הראשונות לשאר המילים.).

**שאלה 3:**

[**https://colab.research.google.com/drive/1SR1gdzyNuD2SWqASyV6qAFuJN7tz\_flt#scrollTo=vamleASf5qZ6**](https://colab.research.google.com/drive/1SR1gdzyNuD2SWqASyV6qAFuJN7tz_flt#scrollTo=vamleASf5qZ6)

הכל מבחינת קוד נמצא בלינק הזה, בשביל להפעיל את הקוד צריך להוסיף קובץ בשם "words.txt" לסביבת עבודה ולרשום מילה שרוצים לחפש בצורה הבאה (קובץ XLSX נוצר מהקוד):





we can't calculate Recall because we use an API that returns all relevant definitions, so we return all definitions but we only show the top 10 definitions using our scoring system, even though we show the top 10 we have essentially returned all the definitions which is why it is 100%, we are also not able to suggest a modified query for the fact that our website is a dictionary website, so we need to assume that the person inputted the correct query and we can't mess with it.

The calculations are done using these 2 equations:

Precision = relevant\_retrieved / retrieved\_doc

Recall = relevant\_retrieved / relevant\_doc

We already explained that with Recall the relevant\_retrieved = relevant\_doc so we always get 100%, with precision we count relevant docs as ones marked with 'y' when getting the feedback from the user -as shown in the picture- and the retrieved\_doc are what we got from our API as a result, so that is why we get the results that are shown above.