**אחזור מידע – תרגיל בית 2 – אביב התשפ"ד**

**מגישים:**

**מור שמואל 315040980**

**מעיין אביטן 313550907**

**אופק בן אברהם 313310559**

**רותם פורת 206607731**

[**https://github.com/avtn96/Information\_Retrieval**](https://github.com/avtn96/Information_Retrieval)

שאלה 1: מודלים לאחזור מידע

שאלה 1:

נתונים:

Total relevant documents (R): 100

Retrieved documents (P): 18 (8 relevant + 10 non-relevant)

Relevant documents retrieved (A): 8

נחשב את הprecision ואת הrecall:

Precision = (relevant documents retrieved)/(retrieved documents) = (8/18) = 0.444 = 44.4%

Recall = (relevant documents retrieved)/(total relevant documents) = (8/100) = 0.08=8%

שאלה 2:

לפי הנתונים אפשר לראות כי עבור מנוע חיפוש 1 התוצאות הרלוונטיות הם במיקומים: 1,3,4,7,9,10

ועבור מנוע חיפוש 2 התוצאות הרלוונטיות הם במיקומים: 2,3,4,6,7

בטבלה נתון כי עבור כל מנוע חיפוש יש 10 תוצאות ולכן כל שורה בטבלה מייצגת חזרה אחת.

נציג את התוצאות של percision&recall עבור כל מנוע חיפוש:

Engine 1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Relevant Retrieved | Retrieved documents | |  | | --- | | Precision |  |  | | --- | |  | | | Recall | | --- |  |  | | --- | |  | |
| 1 | 1 | 1 | 1/1 = 1 | 1/10 = 0.10 |
| 2 | 1 | 2 | 1/2 = 0.5 | 1/10 = 0.10 |
| 3 | 2 | 3 | 2/3 = 0.67 | 2/10 = 0.2 |
| 4 | 3 | 4 | 3/4 = 0.75 | 3/10 = 0.3 |
| 5 | 3 | 5 | 3/5 = 0.6 | 3/10 = 0.3 |
| 6 | 3 | 6 | 3/6 = 0.5 | 3/10 = 0.3 |
| 7 | 4 | 7 | 4/7 = 0.57 | 4/10 = 0.4 |
| 8 | 4 | 8 | 4/8 = 0.5 | 4/10 = 0.4 |
| 9 | 5 | 9 | 5/9 = 0.56 | 5/10 = 0.5 |
| 10 | 6 | 10 | 6/10 = 0.6 | 6/10 = 0.6 |

Engine 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Relevant Retrieved | Retrieved documents | |  | | --- | | Precision | | | Recall | | --- | |
| 1 | 0 | 1 | 0/1 = 0 | 0/10 = 0 |
| 2 | 1 | 2 | 1/2 = 0.5 | 1/10 = 0.1 |
| 3 | 2 | 3 | 2/3 = 0.67 | 2/10 = 0.2 |
| 4 | 3 | 4 | 3/4 = 0.75 | 3/10 = 0.3 |
| 5 | 3 | 5 | 3/5 = 0.6 | 3/10 = 0.3 |
| 6 | 4 | 6 | 4/6 = 0.67 | 4/10 = 0.4 |
| 7 | 5 | 7 | 5/7 = 0.71 | 5/10 = 0.5 |
| 8 | 5 | 8 | 5/8 = 0.63 | 5/10 = 0.5 |
| 9 | 5 | 9 | 5/9 = 0.56 | 5/10 = 0.5 |
| 10 | 5 | 10 | 5/10 = 0.5 | 5/10 = 0.5 |

כדי לחשב את הinterpolated precision:

נגדיר 11 רמות Recall שהן: 0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0

נחשב את ה- הinterpolated precision עבור כל רמת Recall.

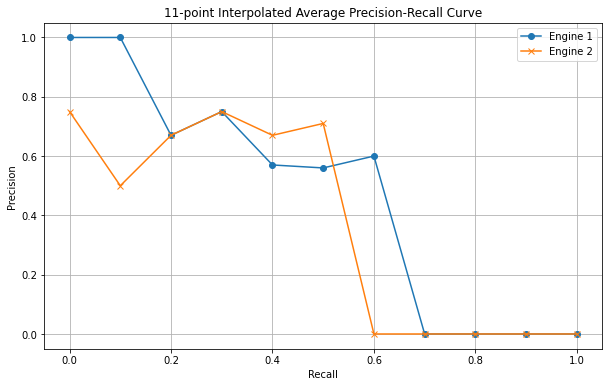
Engine1:

|  |  |  |  |
| --- | --- | --- | --- |
| Interpolated Percision | | Recall | | --- |  |  | | --- | |  | |
| 1 | 0 |
| 1 | 0.1 |
| 0.67 | 0.2 |
| 0.75 | 0.3 |
| 0.57 | 0.4 |
| 0.56 | 0.5 |
| 0.6 | 0.6 |
| 0 | 0.7 |
| 0 | 0.8 |
| 0 | 0.9 |
| 0 | 1.0 |

Engine2:

|  |  |  |  |
| --- | --- | --- | --- |
| Interpolated Percision | | Recall | | --- |  |  | | --- | |  | |
| 0.75 | 0 |
| 0.5 | 0.1 |
| 0.67 | 0.2 |
| 0.75 | 0.3 |
| 0.67 | 0.4 |
| 0.71 | 0.5 |
| 0 | 0.6 |
| 0 | 0.7 |
| 0 | 0.8 |
| 0 | 0.9 |
| 0 | 1.0 |

וזה הגרף שקיבלנו:



נחשב את הf-measure :

Engine1:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Relevant Retrieved | Retrieved documents | |  | | --- | | Precision |  |  | | --- | |  | | | Recall | | --- |  |  | | --- | |  | | f-measure |
| 1 | 1 | 1 | 1/1 = 1 | 1/10 = 0.10 | 0.2358 |
| 2 | 1 | 2 | 1/2 = 0.5 | 1/10 = 0.10 | 0.2049 |
| 3 | 2 | 3 | 2/3 = 0.67 | 2/10 = 0.2 | 0.3629 |
| 4 | 3 | 4 | 3/4 = 0.75 | 3/10 = 0.3 | 0.4870 |
| 5 | 3 | 5 | 3/5 = 0.6 | 3/10 = 0.3 | 0.4412 |
| 6 | 3 | 6 | 3/6 = 0.5 | 3/10 = 0.3 | 0.4032 |
| 7 | 4 | 7 | 4/7 = 0.57 | 4/10 = 0.4 | 0.4944 |
| 8 | 4 | 8 | 4/8 = 0.5 | 4/10 = 0.4 | 0.4587 |
| 9 | 5 | 9 | 5/9 = 0.56 | 5/10 = 0.5 | 0.5368 |
| 10 | 6 | 10 | 6/10 = 0.6 | 6/10 = 0.6 | 0.6 |

Engine2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Relevant Retrieved | Retrieved documents | |  | | --- | | Precision | | | Recall | | --- | | f-measure |
| 1 | 0 | 1 | 0/1 = 0 | 0/10 = 0 | NaN |
| 2 | 1 | 2 | 1/2 = 0.5 | 1/10 = 0.10 | 0.2049 |
| 3 | 2 | 3 | 2/3 = 0.67 | 2/10 = 0.2 | 0.3629 |
| 4 | 3 | 4 | 3/4 = 0.75 | 3/10 = 0.3 | 0.4870 |
| 5 | 3 | 5 | 3/5 = 0.6 | 3/10 = 0.3 | 0.4412 |
| 6 | 4 | 6 | 4/6 = 0.67 | 4/10 = 0.4 | 0.5390 |
| 7 | 5 | 7 | 5/7 = 0.71 | 5/10 = 0.5 | 0.6167 |
| 8 | 5 | 8 | 5/8 = 0.63 | 5/10 = 0.5 | 0.5761 |
| 9 | 5 | 9 | 5/9 = 0.56 | 5/10 = 0.5 | 0.5368 |
| 10 | 5 | 10 | 5/10 = 0.5 | 5/10 = 0.5 | 0.5 |

הממוצע במנוע חיפוש 1 של f-measure הוא 0.42249

הממוצע במנוע חיפוש 2 של f-measure הוא 0.47384

לפי זה ניתן לראות כי מנוע 2 טוב יותר ממנוע 1.

שאלה 2:  
  
א. לשם בניית Inverted index נרצה קודם להסיר את הstop words מהמילים במסמכים הנתונים:

“D1: “dogs love music, listen Rolling Stones

“D2: “Information Retrieval course

“D3: “dog can roll. He loves rolling throwing stones

“D4: “ help pick up stones from road

כעת נסיר סיומות s, ing, ed, ly:

“D1: “dog love music, listen Roll Stone

“D2: “Information Retrieval course

“D3: “dog can roll. He love roll throw stone

“D4: “ help pick up stone from road

ונעבור לבניית Inverted index:

dog: [1, 3]

love: [1, 3]

music: [1]

listen: [1]

roll: [1, 3]

stone: [1, 3, 4]

information: [2]

retrieval: [2]

course: [2]

can: [3]

throw: [3]

help: [4]

pick: [4]

up: [4]

road: [4]

ב. מכיוון שכאן הדרישה היא לבנות Inverted index לכל term נעשה את אותו התהליך רק ללא הסרת הסיומות s, ing, ed, ly:

dogs: [1]

love: [1]

music: [1]

listen: [1]

rolling: [1, 3]

stones: [1, 3, 4]

information: [2]

retrieval: [2]

course: [2]

dog: [3]

can: [3]

roll: [3]

loves: [3]

throwing: [3]

help: [4]

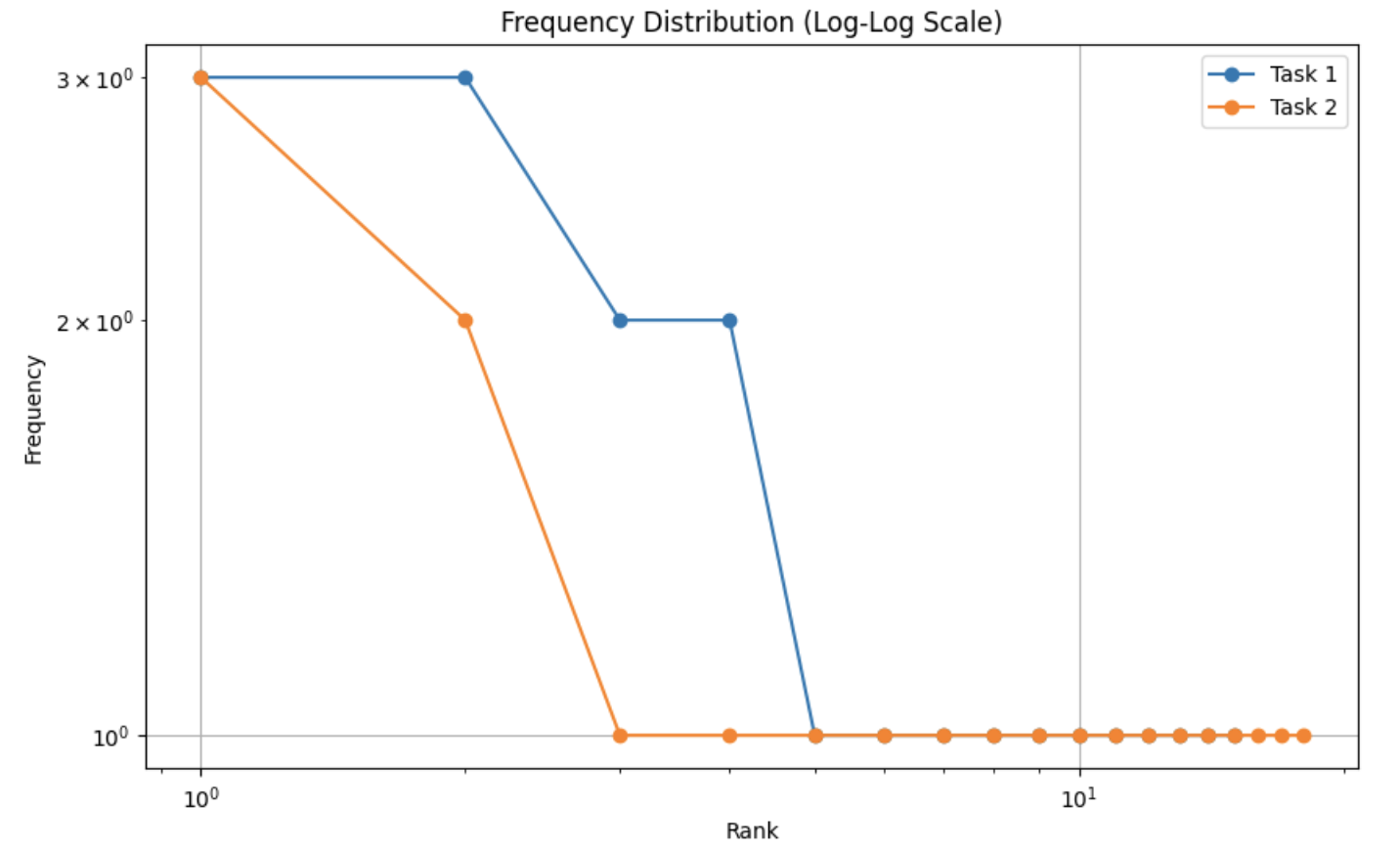
pick: [4]

up: [4]

road: [4]

ג.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| משימה 1 - תדירות | משימה 1 - מילים | משימה 2 - תדירות | משימה 2 - מילים | Rank |
| 3 | roll | 3 | stones | 1 |
| 3 | stone | 2 | rolling | 2 |
| 2 | dog | 1 | dogs | 3 |
| 2 | love | 1 | love | 4 |
| 1 | music | 1 | music | 5 |
| 1 | listen | 1 | listen | 6 |
| 1 | information | 1 | information | 7 |
| 1 | retrieval | 1 | retrieval | 8 |
| 1 | course | 1 | course | 9 |
| 1 | can | 1 | dog | 10 |
|  | throw | 1 | can | 11 |
|  | help | 1 | roll | 12 |
|  | pick | 1 | loves | 13 |
|  | up | 1 | throwing | 14 |
|  | road | 1 | help | 15 |
|  |  | 1 | pick | 16 |
|  |  | 1 | up | 17 |
|  |  | 1 | road | 18 |



לא, הם אינם עומדים בחוק Zipf מכיוון שהגרף אינו מציג קו ישר על סקאלת לוג-לוג. ההתפלגות מראה ירידות חדות ולא אחידות בתדירות המילים.

שאלה 3: קדם פרויקט –בניית זחלן



**חלק א**

התגיות שנחזיר: title, duration, gener, description, critic rating, audience rating, releas date, type, netflix, max-us, fandango, vudu, disney+, paramount+, peacock, apple tv, apple tv+, hulu, amazon prime, consensus text, URL

אנו נרצה להחזיר נתונים שקשורים לסרטים או סדרות שהם מפולטרים(לדוגמה סינון החל משנה מסוימת, ז׳אנר מסוים, דירוג מסוים), בנוסף נרצה להחזיר גם נתוני השוואות הקשורים לנתוני הסרטים והסדרות(לדוגמה השוואות נתוני סרטים מול סדרות)  
**השאילתא הראשונה:**

**movies 2023 with duration > 1.5 h**

השאילתא תחזיר את כל הלינקים של הסרטים ששוחררו בשנת 2023 שאורך הסרט הוא מעל שעה וחצי. בחרנו בשאילתא זאת כיוון שהאתר לא נותן מענה עליה וכן כי היא עונה על התגיות אותם אנו מעוניינים לחקור.

**השאילתא השנייה:**

**average of critics of movies released in 2023 compared to tv shows released in 2023**

השאילתא תחזיר השוואה בין הממוצע של מבקרי הסרטים לסרטים ששוחררו בשנת 2023 אל מול מבקרי הסדרות ששוחררו בשנת 2023. בחרנו בשאילתא זאת כיוון שהאתר לא נותן מענה עליה וכן כי היא עונה על התגיות אותם אנו מעוניינים לחקור ומחזירה תוצאה מעניינת יותר ביחס למנוע חיפוש רגיל(תוצאה שמזכירה שימוש בכלי AI)

**השאילתא השלישית:**

**tv shows with release year > 2011**

השאילתא תחזיר את כל הלינקים של סדרות טלוויזיה ששוחררו החל משנת 2011, גם כאן אנו בחרנו בשאילתא זו כדי להשתמש בתגיות אותם אנו חוקרים.

**חלק ב**

1. קובץ inverted index נמצא ביחד עם קבצי הקוד בגיט.  
   קישור לקוד שמבצע זאת: <https://github.com/avtn96/Information_Retrieval/blob/main/inverted_index.py>
2. השאילתה שבחרנו: "movies 2023 with duration > 1.5 h":

**הסבר תהליך חישוב TF-IDF:**

TF - (Term Frequency)

מדד המתאר את מספר הפעמים שהמונח מופיע במסמך מסוים יחסית לאורך המסמך. זה נותן הערכה לגבי חשיבות המונח בתוך המסמך עצמו.

IDF - (Inverse Document Frequency)

מדד המתאר כמה המונח נדיר באוסף המסמכים כולו. מונחים המופיעים בהרבה מסמכים יקבלו ערך IDF נמוך, בעוד שמונחים נדירים יקבלו ערך גבוה יותר.

TF-IDF

זהו מכפלת ה TF וה.IDF השילוב של שני הערכים מאפשר להעריך את החשיבות היחסית של מונח מסוים במסמך ספציפי בהשוואה לכלל המסמכים. זה שימושי במיוחד בניתוח טקסטים כדי לזהות מונחים שמשמעותיים לתוכן הספציפי.

**פירוט שלבי החישוב:**

1. פונקציה : fetch\_and\_process\_text(url)

תיאור:

פונקציה זו אחראית לטעון ולעבד את הטקסט מ-URL נתון. היא מבצעת בקשה לכתובת האינטרנט, מחלצת את הטקסט מהדף עם BeautifulSoup, ואז מבצעת תיקון והסרת מילים שאינן רלוונטיות (כמו (stop words .

- בקשה ל-URL עם ספריית requests .

- שימוש ב-BeautifulSoup לחלץ טקסט נקי מהHTML .

- תיקון הטקסט: חיתוך למילים, הורדת תווים שאינם אלפביתיים והסרת מילות עצירה.

- החזרת רשימת המילים שעברו עיבוד.

2. פונקציה: process\_multiple\_urls(urls, words)

תיאור:

פונקציה זו מעבדת רשימת URLs ומחשבת את ה-TF-IDF לכל מילה בכל אחד מה-URLs היא משתמשת בפונקציה fetch\_and\_process\_text לקבלת המילים מכל URL ומחשבת את תדירות המילה בכל מסמך ובמאגר המסמכים.

שלבים:

- ריצה על כל URL וקבלת רשימת מילים מעובדות.

- חישוב תדירות הופעת כל מילה בכל מסמך (TF).

- חישוב מספר המסמכים שבהם המילה מופיעה לפחות פעם אחת.

- חישוב ה-IDF עבור כל מילה.

- חישוב ה-TF-IDF על ידי כפל ה-TF ב IDF .

3. הדפסות בתוך לולאה

תיאור:

בכל איטרציה של עיבוד URL הקוד מדפיס את ה-TF-IDF לכל מילה.

שלבים:

- חישוב והדפסת ה-TF-IDF לכל מילה.

**התוצאה:**

TF-IDF for 'movie': 0.01924117728284252

TF-IDF for 'duration': 0

TF-IDF for '2023': 0.003947097333711943

TF-IDF for '1.5': 0.0003071314321727182

TF-IDF for 'h': 0.0003597428038138616

קישור לקוד שמבצע זאת: <https://github.com/avtn96/Information_Retrieval/blob/main/tf_idf.py>

1. השאילתא שהורצה: movies 2023 with duration > 1.5 h

Ron:  
515 <https://www.rottentomatoes.com/m/asteroid_city>



1035 <https://www.rottentomatoes.com/m/napoleon_2023>



1036 <https://www.rottentomatoes.com/m/saw_x>



13 <https://www.rottentomatoes.com/m/pain_hustlers>



1037 <https://www.rottentomatoes.com/m/sisu_2022>



475 <https://www.rottentomatoes.com/m/memory_2023>



988 <https://www.rottentomatoes.com/m/saltburn>



997 <https://www.rottentomatoes.com/m/ferrari_2023>

488 <https://www.rottentomatoes.com/m/ant_man_and_the_wasp_quantumania>



1013 <https://www.rottentomatoes.com/m/no_one_will_save_you>



Precision: 10/10 = 1

Recall: 10/76 = 0.14

Amit:

515 <https://www.rottentomatoes.com/m/asteroid_city>



1035 <https://www.rottentomatoes.com/m/napoleon_2023>



1036 <https://www.rottentomatoes.com/m/saw_x>



13 <https://www.rottentomatoes.com/m/pain_hustlers>



1037 <https://www.rottentomatoes.com/m/sisu_2022>



475 <https://www.rottentomatoes.com/m/memory_2023>



988 <https://www.rottentomatoes.com/m/saltburn>



997 <https://www.rottentomatoes.com/m/ferrari_2023>

488 <https://www.rottentomatoes.com/m/ant_man_and_the_wasp_quantumania>



1013 <https://www.rottentomatoes.com/m/no_one_will_save_you>



Precision: 8/10 = 0.8

Recall: 8/76 = 0.105