אחזור מידע

מטלה מספר 2

מגישים

אלעד פישר - 318882800

חן חנצין - 208636761

גלעד סגל - 314865841

ניצן ממן – 205798515

קישור לגיט: <https://github.com/gilseg10/Information-recovery-class>

חלק א – שאלות מחומר ההרצאות

פתרון שאלה 1:

ראשית, נחשב את ה-precision: Precision הוא יחס המסמכים הרלוונטיים שהמערכת החזירה מתוך כלל המסמכים שהמערכת החזירה:

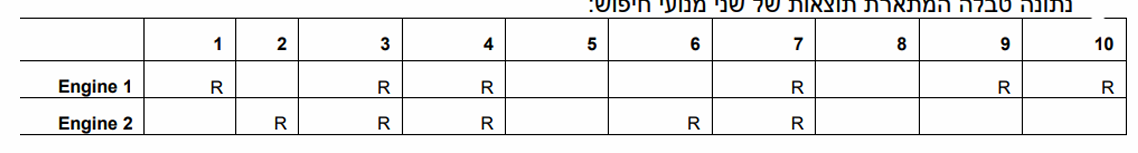
כעת נחשב את ה-recall: Recall הוא יחס המסמכים הרלוונטיים שהמערכת החזירה מתוך כלל המסמכים הרלוונטיים באוסף.

לסיכום:

Precision = 0.444

Recall = 0.4

פתרון שאלה 2:



מההנחה שלנו : הייצוג R אומר כי המסמך שחזר באותו מנוע הוא רלוונטי.

עבור חישוב Precision אנו נחשב לפי הנוסחה עבור Precision@K , כלומר, נחלץ באחוזים את כמות הרלוונטיים שהוחזרו בהחזר K.

עבור חישוב recall, נחשב לפי הנוסחה שהוצגה לנו: Recall = TP / (TP + FN) עבור כל החזרה K.

|  |  |  |
| --- | --- | --- |
| Engine 2 | Engine 1 | Rank |
| P = 0/1 = 0.0  R = 0/10 = 0.0 | P = 1/1 = 1.0  R = 1/10 = 0.1 | **1** |
| P = 1/2 = 0.5  R = 1/10 = 0.1 | P = 1/2 = 0.5  R = 1/10 = 0.1 | **2** |
| P = 2/3 = 0.666  R = 2/10 = 0.2 | P = 2/3 = 0.666  R = 2/10 = 0.2 | **3** |
| P = 3/4 = 0.75  R = 3/10 = 0.3 | P = 3/4 = 0.75  R = 3/10 = 0.3 | **4** |
| P = 3/5 = 0.6  R = 3/10 = 0.3 | P = 3/5 = 0.6  R = 3/10 = 0.3 | **5** |
| P = 4/6 = 0.66  R = 4/10 = 0.4 | P = 3/6 = 0.5  R = 3/10 = 0.3 | **6** |
| P = 5/7 = 0.71  R = 5/10 = 0.5 | P = 4/7 = 0.57  R = 4/10 = 0.4 | **7** |
| P = 5/8 = 0.625  R = 5/10 = 0.5 | P = 4/8 = 0.5  R = 4/10 = 0.4 | **8** |
| P = 5/9 = 0.55  R = 5/10 = 0.5 | P = 5/9 = 0.55  R = 5/10 = 0.5 | **9** |
| P = 5/10 = 0.5  R = 5/10 = 0.5 | P = 6/10 = 0.6  R = 6/10 = 0.6 | **10** |

כעת נציג טבלה וגרף של 11 נקודות האינטרפולציה:  
עבור כל מנוע נעבוד כך: עבור כל רמה של Recall (בין 0.0 ל 1.0), אנו נמצא את הערך Precision הגבוה ביותר שמשויך לRecall שגדול/שווה לאותה רמה.

מנוע 1:

|  |  |
| --- | --- |
| Interpolated Precision | Recall Level |
| 1.0 | 0.0 |
| 1.0 | 0.1 |
| 0.75 | 0.2 |
| 0.75 | 0.3 |
| 0.6 | 0.4 |
| 0.6 | 0.5 |
| 0.6 | 0.6 |
| 0.0 | 0.7 |
| 0.0 | 0.8 |
| 0.0 | 0.9 |
| 0.0 | 1.0 |

נציג כעת את הגרף של מנוע 1:

מנוע 2:

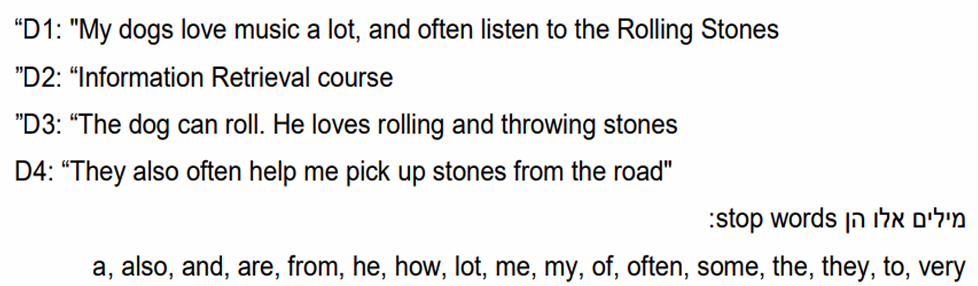
|  |  |
| --- | --- |
| Interpolated Precision | Recall Level |
| 0.75 | 0.0 |
| 0.75 | 0.1 |
| 0.75 | 0.2 |
| 0.75 | 0.3 |
| 0.71 | 0.4 |
| 0.71 | 0.5 |
| 0.0 | 0.6 |
| 0.0 | 0.7 |
| 0.0 | 0.8 |
| 0.0 | 0.9 |
| 0.0 | 1.0 |

נציג כעת את הגרף של מנוע 2:

כעת נחשב את f-measure עם :

ניתן לראות כי לפי חישובי F-measure, הערכים כמעט זהים, אך בסוף מנוע 1 עדיף על פני מנוע 2

פתרון שאלה 3:



א+ב. בניית טבלה של inverted index:

|  |  |
| --- | --- |
| Documents | Term |
| 1, 3 | dog |
| 1, 3 | love |
| 1 | music |
| 1 | listen |
| 1, 3 | roll |
| 1, 3, 4 | stone |
| 2 | information |
| 2 | retrieval |
| 2 | course |
| 3 | can |
| 3 | throw |
| 4 | help |
| 4 | pick |
| 4 | up |
| 4 | road |

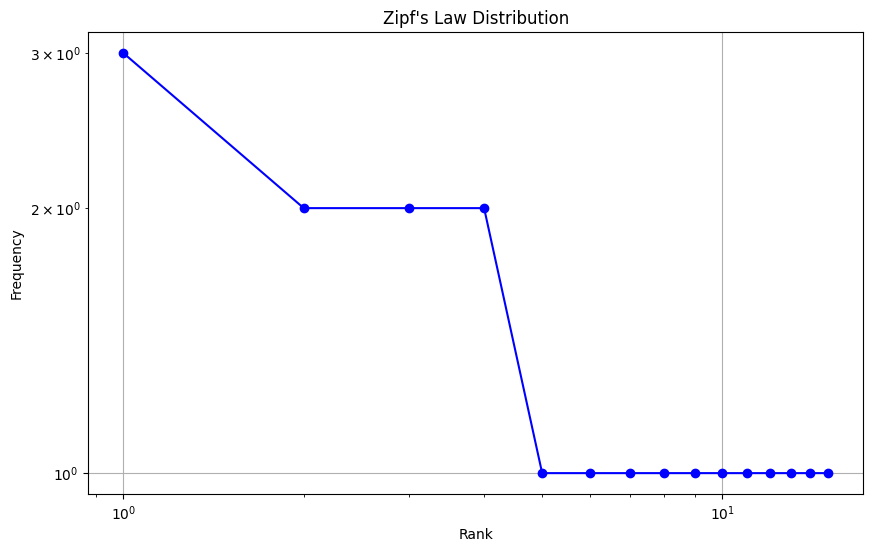
ג. בדיקת חוק Zipf:

ראשית, ניצור טבלת תדירות בהתבסס על הinverted index:

|  |  |
| --- | --- |
| Frequency | Term |
| 2 | dog |
| 2 | love |
| 1 | music |
| 1 | listen |
| 2 | roll |
| 3 | stone |
| 1 | information |
| 1 | retrieval |
| 1 | course |
| 1 | can |
| 1 | throw |
| 1 | help |
| 1 | pick |
| 1 | up |
| 1 | road |

נמיין ונקבל את הטבלה הבאה:

|  |  |
| --- | --- |
| Frequency | Term |
| 3 | stone |
| 2 | dog |
| 2 | love |
| 2 | roll |
| 1 | listen |
| 1 | music |
| 1 | information |
| 1 | retrieval |
| 1 | course |
| 1 | can |
| 1 | throw |
| 1 | help |
| 1 | pick |
| 1 | up |
| 1 | road |



ניתן לראות כי הנקודות יוצרות קו ישר פחות או יותר בסקלה לוגריתמית, ועובדה זו מרמזת כי **אכן הנתונים עוקבים אחרי חוק Zipf**.

שאלה 3 – בניית זחלן (קדם פרויקט)

אנו מתמקדים בבניית זחלן לאתר האינטרנטי **Steam**, האתר המוביל בעולם למכירה והפצת משחקי מחשב בפורמט דיגיטלי. כל אדם שנכנס יוצר לעצמו משתמש, ומסכים לתיעוד המידע עליו לאורך השימוש באתר/אפליקציה של החברה. בהתבסס מידע זה אנו נעבוד על השאילתות שלנו.

לצרוך חילוץ מידע, אנו משתמשים ב [Steam Web API](https://developer.valvesoftware.com/wiki/Steam_Web_API), ולאחר מחקר מעמיק על היכולות והמידע שהAPI יכול לחלץ, נאלצנו לשנות את השאילתות שלנו, שאותן נציג כעת.

**חלק א:**

כעת נגיד במדויק את השאילתות אשר עליהן נעבוד ונחקור:

1. שאילתה: מהו ממוצע זמן המשחק ביחס לכלל השחקנים שרשומים אליו.  
   הסבר: מתוך ה-API, אנו משתמשים ב EndPoint הבא: <http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/>  
   שמאפשר החזרה של רשימת משחקים שנרכשו (בתשלום/חינם) עבור כל משתמש, וגם זמן משחק כולל של כל משתמש על כל אחד מהמשחקים.   
   התגיות הרלוונטיות בהקשר שאילתה זו הן:  
   **playtime\_forever** - זמן המשחק הכולל של משחק מסוים (בדקות).  
   השאילתה מחזירה: רשימה של משחקים, כאשר עבור כל משחק: מתעדים את **כמות המשתמשים** וגם כן את **הזמן המשחק של כל משתמש**, ובנוסף גם **הזמן הממוצע ששיחקו במשחק**.
2. שאילתה: חילוץ המילים הבולטות ביותר בהישגים של משחק. **(Achievement או Badges).**הסבר: לכל משחק שרשום בSteam, יש שלל הישגיים (Badges) שמשתמש יכול להשיג לאורך זמן המשחק שלו. לכל הישג שכזה יש תיאור ושם שמקושר למשחק, וגם אחוז גלובלי שמציג כמה מכלל המשתמשים בעולם שרכשו את המשחק השיגו את ההישג הזה.  
   מידע זה ניתן לחלץ מה-API דרך הEndPoint הבא:   
   <http://api.steampowered.com/ISteamUserStats/GetGlobalAchievementPercentagesForApp/v0002>   
   התגיות הרלוונטיות בהקשר לשאילתה זו הן:  
   **gameid** – אמצעי זיהוי ייחודי עבור כל משחק.  
   השאילתה מחזירה: רשימה של כלל ההישגים (שמות ואחוז הישג גלובלי) למשחק מסוים. דרכו, אנו נחלץ את המילים הבולטות ביותר מכלל שמות ההישגים הקיימים למשחק.
3. שאילתה: חילוץ המילים הנפוצות ביותר בתיאור של כל משחק.  
   הסבר: לכל משחק שמופץ ב Steam קיים עמוד משלו ובתוך העמוד ישנו תיאור המשחק תחת הכותרת "ABOUT THIS GAME". אנו נחפש את המילים הנפוצות ביותר בכל תיאור משחק אשר לרשותנו דרך שאיבת עמוד ה-HTML של המשחק וחילוץ הפסקאות המתארות את המשחק. ניתן לחלץ את עמוד ה-HTML של כל משחק דרך נקודת הגישה הבאה: https://store.steampowered.com/app/{appid}  
   התגיות הרלוונטיות בהקשר לשאילתה הן:  
   appid – אמצעי הזיהוי של משחק מסוים לחקירה.  
   game\_area\_description – ה-id של ה-div המכיל את הפסקאות של תיאור המשחק.  
   השאילתה מחזירה: רשימה של המילים הנפוצות ביותר בתיאור משחקים ממוינות בסדר לא עולה.

**חלק ב:**

התוצאות עבור כל אחת מהשאילתות שתיארנו נמצאות בקובץ האקסל שמצורף בהגשה.

נציין את הפרטים הבאים:

1. אנו מצרפים קובץ אקסל של steam ids שמשמש את הזחלן שלנו במגוון שאילתות שונות. בכל שאילתא למעשה הזחלן סורק את קובץ האקסל עם המזהים הייחודיים של משתמשים של אתר Steam ועל ידי כך מוציא את פרטי המידע הרלוונטיים של המשתמשים והמשחקים שלהם.
2. אנו מצרפים קבצי זחלן (פייתון) בתיקיית ההגשה (תחת תיקייה "crawlerFiles") ובנוסף מצרפים קישור לקולאב עם אותם קטעי קוד.

[קישור לקולאב](https://colab.research.google.com/drive/1LmzJBuwgCWVR04HK4Z4LMw_uQz0PS0VS?usp=sharing)

1. אנו מצרפים קובץ אקסל (בשם: "מטלה 2 – תוצאות חישובים") שבו מופיעים החישובים של כל שאילתה: בקובץ ישנם תגיות/tabs עם השם המתאים: Query 2, Query 3.

בנוסף, ניתן לראות את הקובץ עצמו בדרייב:  
[גוגל אקסל לחישובי שאילתות](https://docs.google.com/spreadsheets/d/1t9Fy0EGIe-I3TNBNYDyamwmd013W4i9urrRoGL4iTzw/edit?usp=sharing)

1. התשובות עבור שאילתה 1 הינם ערכים מספריים של ממוצע משחק.  
   אנו הגדרנו בהתאם לממוצעים מאגר קטגוריות עבור כל משחק:  
    Casual, Intense, Competitive.  
   עבור כל משחק שקיבלנו בשאילתה שייכנו קטגוריה בהתאמה.  
   קובץ זה מצורף גם כן בהגשה, ואיתו אנו נמשיך למטלה הבאה לפיתוח.
2. עבור שאילתה 2, אנו מתייחסים לכל מסמך/document כסך כלל המילים שמופיעות בסך כל שמות ההישגים (badges/achievements).
3. חישוב tf-idf ביצענו עבור שאילתה מספר 3.