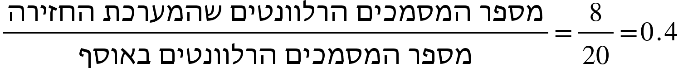
תרגיל בית 2 – Result Wizards

תז: 316350768, 208608448, 208983189, 318185956.

<https://github.com/ndvp39/crawler.git>

שאלה 1:

Precision מחושב באופן הבא: {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mfrac><mrow><mi>&#x5D4;&#x5D7;&#x5D6;&#x5D9;&#x5E8;&#x5D4;</mi><mo>&#xA0;</mo><mi>&#x5E9;&#x5D4;&#x5DE;&#x5E2;&#x5E8;&#x5DB;&#x5EA;</mi><mo>&#xA0;</mo><mi>&#x5D4;&#x5E8;&#x5DC;&#x5D5;&#x5D5;&#x5E0;&#x5D8;&#x5D9;&#x5DD;</mi><mo>&#xA0;</mo><mi>&#x5D4;&#x5DE;&#x5E1;&#x5DE;&#x5DB;&#x5D9;&#x5DD;</mi><mo>&#xA0;</mo><mi>&#x5DE;&#x5E1;&#x5E4;&#x5E8;</mi></mrow><mrow><mi>&#x5D4;&#x5D7;&#x5D6;&#x5D9;&#x5E8;&#x5D4;</mi><mo>&#xA0;</mo><mi>&#x5E9;&#x5D4;&#x5DE;&#x5E2;&#x5E8;&#x5DB;&#x5EA;</mi><mo>&#xA0;</mo><mi>&#x5D4;&#x5DE;&#x5E1;&#x5DE;&#x5DB;&#x5D9;&#x5DD;</mi><mo>&#xA0;</mo><mi>&#x5DE;&#x5E1;&#x5E4;&#x5E8;</mi></mrow></mfrac><mo>=</mo><mfrac><mn>8</mn><mrow><mn>10</mn><mo>+</mo><mn>8</mn></mrow></mfrac><mo>=</mo><mfrac><mn>8</mn><mn>18</mn></mfrac><mo>=</mo><mn>0</mn><mo>.</mo><mn>444</mn></mstyle></math>","origin":"MathType for Microsoft Add-in"}

Recall מחושב באופן הבא: 

שאלה 2:

טבלה המציגה את מונה המסמכים הרלוונטים שחזרו בכל החזרה של מנוע חיפוש (נגזר מהטבלה הנתונה בשאלה):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| מספר החזרה | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Engine 1 | 1 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 5 | 6 |
| Engine 2 | 0 | 1 | 2 | 3 | 3 | 4 | 5 | 5 | 5 | 5 |

לפי נוסחאות ה-precision, ה-recall והטבלה הנ"ל, נחשב באקסל לכל החזרה.

(סך המסמכים הרלוונטים בקורפוס הינו 10).

**טבלה עם recall ו-precision לאחר כל החזרה:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | מספר החזרה | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Precision | Engine 1 | 1/1=1 | 1/2=0.5 | 2/3=0.667 | 3/4=0.75 | 3/5=0.6 | 3/6=0.5 | 4/7=0.571 | 4/8=0.5 | 5/9=0.556 | 6/10=0.6 |
|  | Engine 2 | 0/1=0 | 1/2=0.5 | 2/3=0.667 | 3/4=0.75 | 3/5=0.6 | 4/6=0.667 | 5/7=0.714 | 5/8=0.625 | 5/9=0.556 | 5/10=0.5 |
| Recall | Engine 1 | 1/10=0.1 | 1/10=0.1 | 2/10=0.2 | 3/10=0.3 | 3/10=0.3 | 3/10=0.3 | 4/10=0.4 | 4/10=0.4 | 5/10=0.5 | 6/10=0.6 |
|  | Engine 2 | 0/10=0 | 1/10=0.1 | 2/10=0.2 | 3/10=0.3 | 3/10=0.3 | 4/10=0.4 | 5/10=0.5 | 5/10=0.5 | 5/10=0.5 | 5/10=0.5 |

אינטרפולציה ב-11 נקודות מחלקת את תחום הrecall מ-0 עד ל-1 ל-11 נקודות קבועות (0.0 – 1.0).

בכל אחת מהנקודות האלו, לוקחים את הערך המקסימלי של ה-precision שנמצא עבור ערך recall השווה או גדול לנקודה זו.

**טבלה וגרף של 11 נקודות אינטרפולציה:**

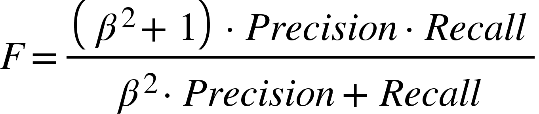
**Engine 1:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recall | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| Interpolated Precision | 1 | 1 | 0.75 | 0.75 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 |

**Engine 2:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recall | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| Interpolated Precision | 0.75 | 0.75 | 0.75 | 0.75 | 0.714 | 0.714 | 0.714 | 0.714 | 0.714 | 0.714 | 0.714 |

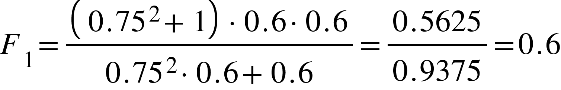
**חישוב f-measure מתבצע באופן הבא (נתון {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>&#x3B2;</mi><mo>=</mo><mn>0</mn><mo>.</mo><mn>75</mn></mstyle></math>","origin":"MathType for Microsoft Add-in"}):**



ניקח את הprecision- וה-recall בהחזרה 10 (מה שהשאילתה החזירה בסוף) בכל מנוע.

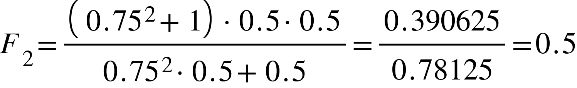
f-measure עבור מנוע 1:

Precision = 0.6), (Recall = 0.6



f-measure עבור מנוע 2:

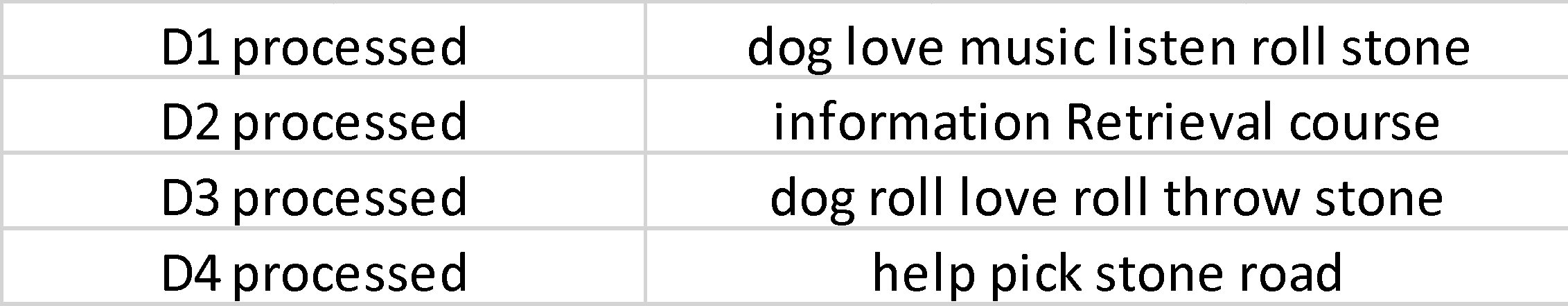
Precision = 0.5), (Recall = 0.5

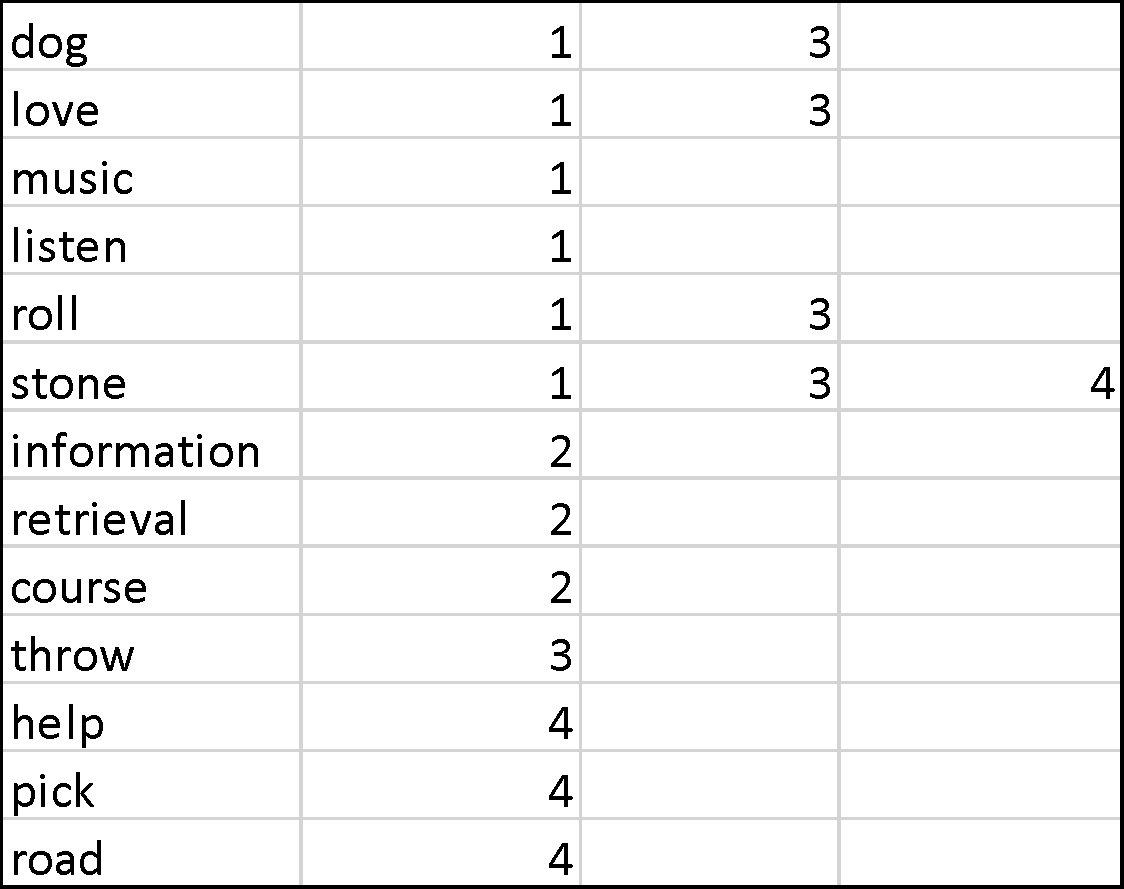


**לכן, קיבלנו שמנוע חיפוש 1 טוב יותר ממנוע חיפוש 2.** ({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>F</mi><mn>1</mn></msub><mo>=</mo><mn>0</mn><mo>.</mo><mn>6</mn><mo>&gt;</mo><msub><mi>F</mi><mn>2</mn></msub><mo>=</mo><mn>0</mn><mo>.</mo><mn>5</mn><mspace linebreak=\"newline\"/></mstyle></math>","origin":"MathType for Microsoft Add-in"} ).

שאלה 3:

א+ב:





ג. הגרף התפלגות מתנהל בקירוב ע"פ חוק zipf למרות מרחב המדגם הקטן. מילים בדרגת נפוצות 3 מופיעות פי 3 פחות ממילה בדרגת נפוצות 1 ופי 2-3 פחות ממילים בדרגת נפוצות 2.

תמונה שמכילה טקסט, צילום מסך, עלילה, מספר

התיאור נוצר באופן אוטומטי

זחלן:

**חלק א'**

השאילתות שלנו:

1. Which movies have the best directors?

בשאילתה זו נרצה למצוא את הבמאים הטובים ביותר באתר rotten tomatoes, להחזיר את הבמאים ואת הסרטים שלהם.

נמצא את אותם במאים על ידי מעבר על כל הסרטים לצפייה.

כאשר נחפש בדף המידע על הסרט, נתמקד בשם הבמאי שנמצא ב class = "caregory -wrap" ובדירוגים של הסרט שנמצאים ב- data-json="reviewData". נבצע חישוב המכליל את כמות הסרטים של הבמאי ואת הדירוגים של כל הסרטים השייכים לו. לבסוף, נחזיר את כל הבמאים עם הניקוד הגבוה ביותר שחושב ואת דפי הסרטים השייכים לכל במאי.

1. Which movie has the longest run time in every genre?

בשאילתה זו נרצה למצוא את הסרטים הארוכים ביותר בכל ג'אנר באתר.

נבצע זאת על ידי מעבר על כל הסרטים לצפייה באתר ונמצא את אותם סרטים בכל ז'אנר על ידי שימוש בתגית class = "caregory -wrap" שם נמצא הruntime של כל סרט. לבסוף, נחזיר את דף הסרט הארוך ביותר מכל ז'אנר.

1. Which movie has the highest box office in every genre?

בשאילתה זו נחפש את הסרט עם הכי הרבה הכנסות בכל ז'אנר.

גם בשאילתה זו נעבור על כל הסרטים ונשתמש בתגית class = "caregory -wrap" שבה נמצא סכום ההכנסות של הסרט. נחזיר את דף הסרט עם ההכנסה הגדולה ביותר מכל ז'אנר.

**כמו כן, לחישוב האינדקס נשתמש בספריית nltk לstop words ולהורדת סופיות במילים כאשר נתמקד בדף הביקורות של כל סרט (ביקורות של מומחים וביקורות של משתמשים) – רלוונטי לשאילתות שלנו.**

**חלק ב'**

**קישור לדרייב:** <https://drive.google.com/drive/folders/1Nd3ITyDM8f03vOzj511P4MyTjWTMnkl1?usp=sharing>

לאחר הרצת הזחלן, קיבלנו שלושה קבצי אקסל הנמצאים בתיקייה queries\_result, כך שבכל קובץ אקסל, התוצאות של שאילתה.

לאחר בניית האינדקס על כל המסמכים שחזרו מהשאילתות, לקחנו את 15 המילים הנפוצות ביותר ועשינו להם inverted index, לכל מילה לקחנו את 20 המסמכים הראשונים שבהם היא מופיעה אם היא מופיע ביותר מ20 מסמכים (כמתבקש).

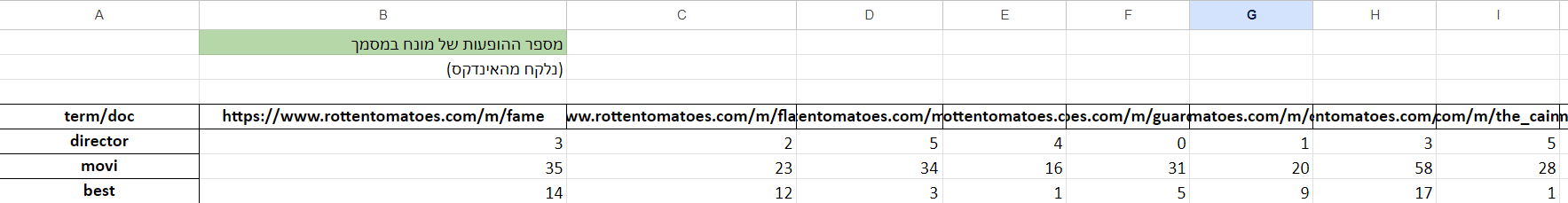
(הערה: האינדקס נמצא בדרייב בתקייה index, כך שהקובץ index\_combined מכליל את כל האינדקס).

ניקח את השאילתה:

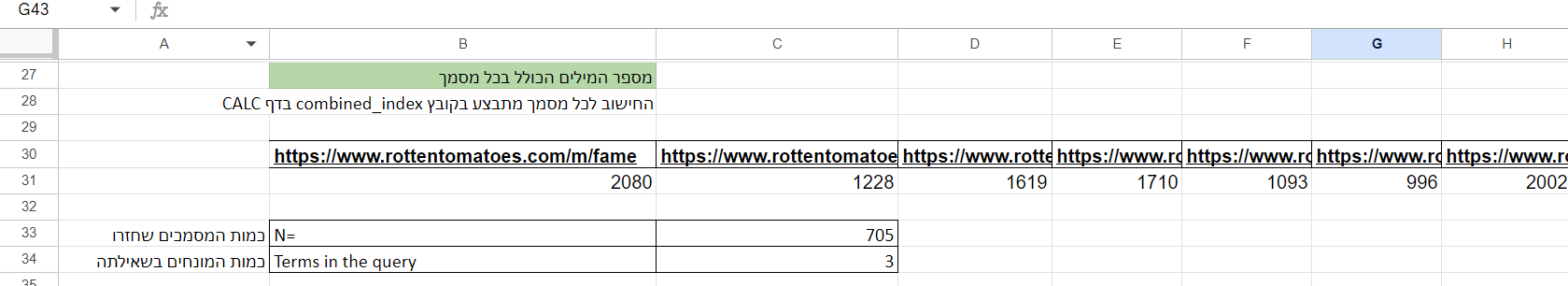
Which movies have the best directors?

ובעזרת האינדקס, נבצע חישוב tf-df של המושגים בשאילתה על פני כל המסמכים שחזרו. את החישובים נבצע בקובץ calculates excel בדף CALCS.

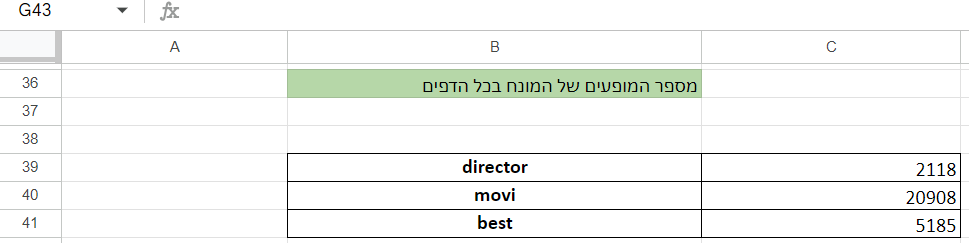
תחילה, נארגן את הנתונים כך שלכל מושג בשאילתה (לאחר הורדת stop words וסופיות), יופיעו כמות ההופעות שלו בכל מסמך (נלקח מהאינדקס).



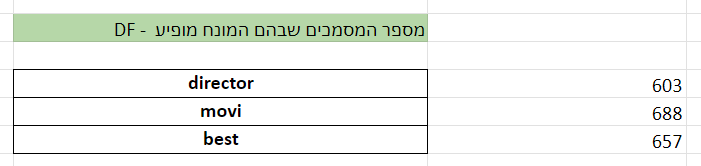
לאחר מכן, נמצא את מספר המילים הכולל בכל מסמך, את כמות המסמכים שהוחזרו, וכמות המושגים בשאילתה.

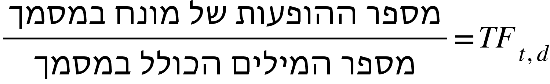


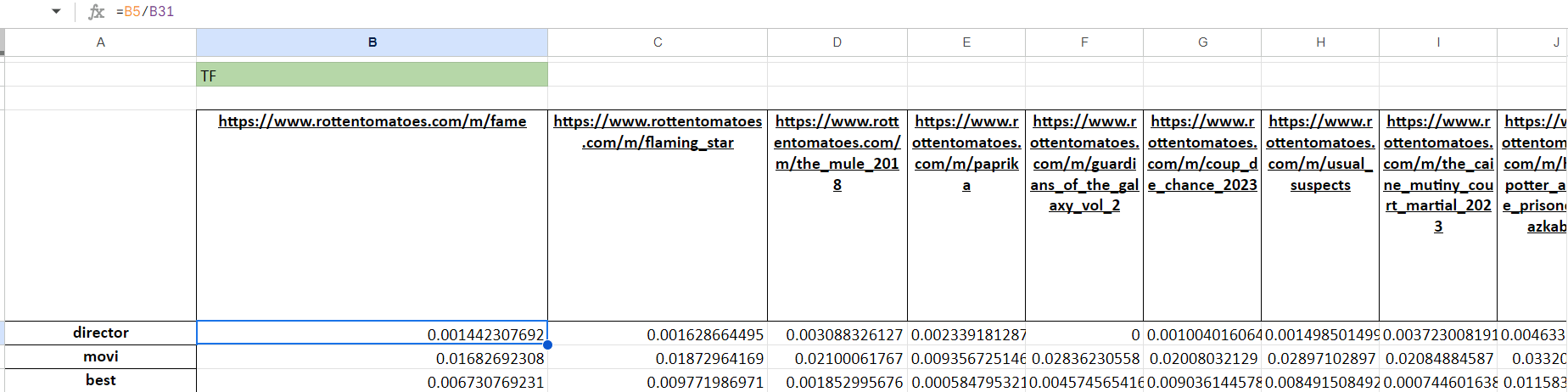
נבנה טבלה שבה יש את מספר המופעים של כל מושג בשאילתה בכל הדפים:

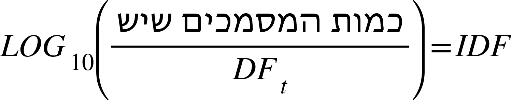


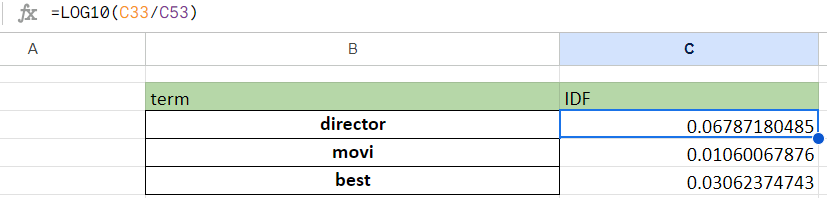
נחשב DF - מספר המסמכים שבהם מופיע כל מונח (באקסל בעזרת COUNTIF):



נחשב TF על ידי הנוסחא: 



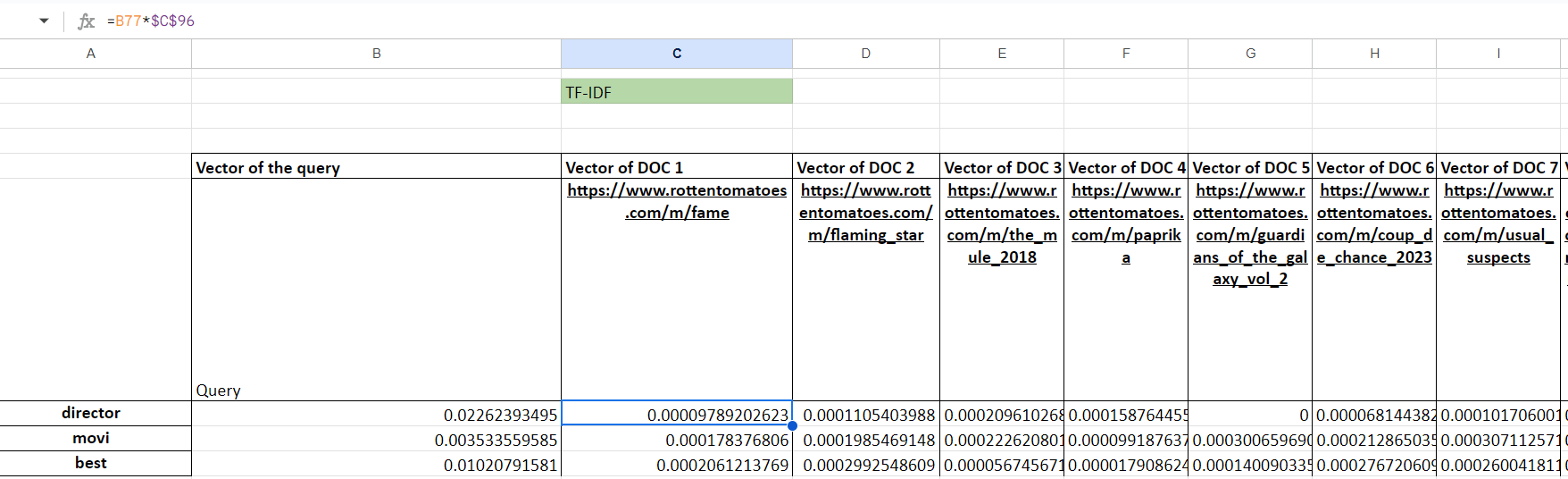
נחשב IDF על ידי הנוסחא: 



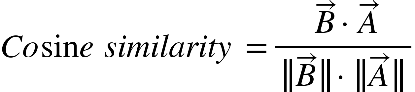
נחשב את הtf-idf- של המושגים בשאילתה על פני על כל המסמכים:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>T</mi><mi>F</mi><mo>-</mo><mi>I</mi><mi>D</mi><msub><mi>F</mi><mrow><mi>t</mi><mo>,</mo><mo>&#xA0;</mo><mi>d</mi></mrow></msub><mo>=</mo><mi>T</mi><msub><mi>F</mi><mrow><mi>t</mi><mo>,</mo><mi>d</mi></mrow></msub><mo>&#xB7;</mo><mi>I</mi><mi>D</mi><msub><mi>F</mi><mi>t</mi></msub><mspace linebreak=\"newline\"/><mspace linebreak=\"newline\"/></mstyle></math>","origin":"MathType for Microsoft Add-in"} כך נחשב לכל מסמך ודף ובנוסף נחשב tf-idf של השאילתה עצמה.

נקבל:

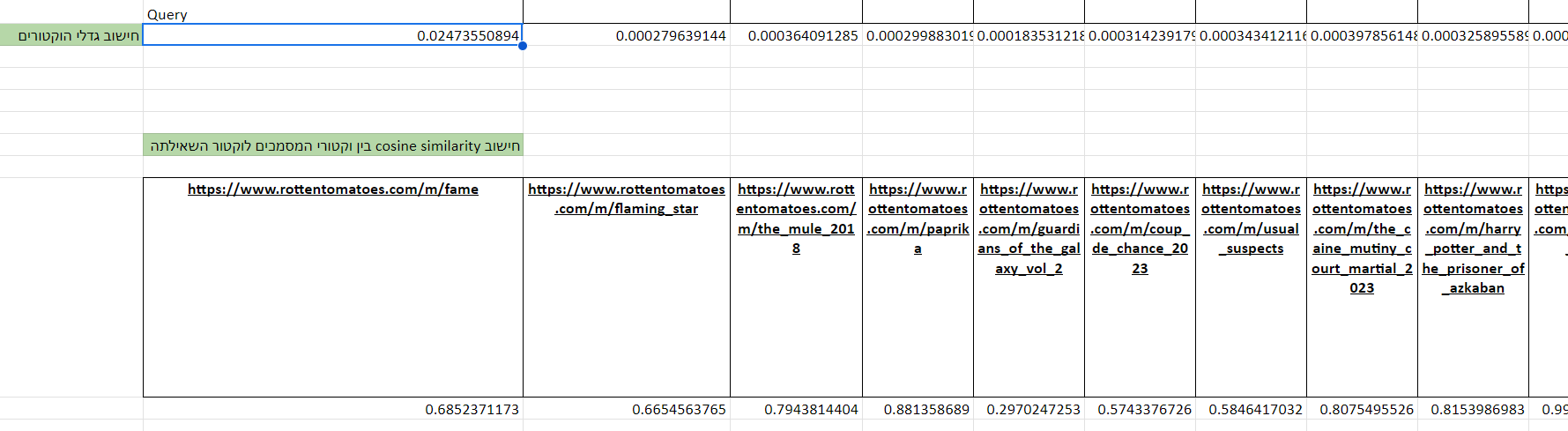


ניתן לראות בעזרת הטבלה שקיבלנו של tf-idf, כי **וקטורי המסמכים הם העמודות**.  
ולכן, על מנת לחשב ולהבין אילו מסמכים הם הקרובים ביותר לשאילתה. נחשב בעזרת Cosine similarity, את הדמיון בין וקטור השאילתה לבין כל אחד מוקטורי המסמכים, כך שלמעשה נקבל ערכים גבוהים עבור דמיון גבוה (כיווני וקטורים דומים יותר).



כאשר נחשב אותו עבור כל וקטור של מסמך. וקטור A – וקטור השאילתה, וקטור B – וקטור של מסמך.

נקבל:



 דוגמא לחישוב גדלי הוקטורים.

דוגמא לחישוב cosine similarity.

3. שאלנו שני אנשים האם 10 התוצאות הראשונות רלוונטיות לשאלה:

1. סטודנט להנדסת חשמל במכללת אפקה בעל ידע נרחב בסרטים.
2. אמא של עודד אשר מכירה במאים ותיקים רבים.

להלן התוצאות:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
| NR | R | NR | NR | R | R | R | R | R | R | P1 |
| R | R | R | NR | R | NR | R | NR | R | R | P2 |

נחשב את ה-Recall וה-Precision עבור כל אחד מהנשאלים.

מכיוון שלא ניתן לדעת באופן אובייקטיבי את מספר התוצאות הרלוונטיות, נשתמש בהיוריסטיקה:

ישנם כ-75 במאים זוכי פרס אוסקר, והמספר הממוצע של סרטים לכל במאי זוכה הוא כ-25. לכן, נניח כי ישנם כ-1875 סרטים העונים על הדרישה.

באתר Rotten Tomatoes ישנם כ-150,000 סרטים, מתוכם סקרנו רק 10,000. לכן, נניח שסקרנו בערך 125 סרטים העונים על הקריטריונים של סרט השייך לבמאי מוצלח.

|  |  |  |
| --- | --- | --- |
| Recall | Precision |  |
| 7/125 | 7/10 | P1 |
| 7/125 | 7/10 | P2 |