**HW3 – Data Diggers**

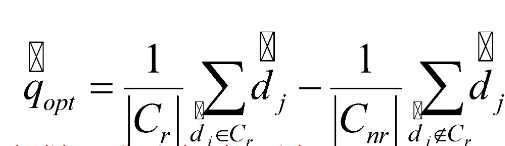
יובל זוהר - 318159373  
אבישי הרשקוביץ - 209460443  
אביחי חדד - 209286665

**1.א.**

תנו דוגמא למקרה שבו מסווג Rocchioמשייך תגית לא נכונה לדוגמת למידה. הוכיחו את טענתכם. רמז: בידקו מקרה חד מימדי

ניקח לדוגמא 6 מסמכים המחולקים ל2 מחלקות כלשהם, כגון ספורט וטכנולוגיה, הפרושים כנקודות בחד-מימד.  
מסמכים הרלוונטיים לטכנולוגיה נמצאים בנקודות: 1,3,5  
מסמכים הרלוונטיים לספורט נמצאים בנקודות: 9,11,13  
כמובן שמרכזי המסמכים ה- Centroid נמצא ב11,3 לטכנולוגיה ולספורט בהתאמה.

כעת נניח ונוסף מסמך חדש הקשור לטכנולוגיה לבישה בספורט (צמיד חכם לדוג') בנקודה 6 וההקשר הרלוונטי הוא **לספורט** יותר מאשר לטכנולוגיה, נראה כיצד המסווג מסווג אותו כקשור לטכנולוגיה.

כידוע:  


כעת שקיבלנו את המסמך הנמצא בנק' 6 ניתן לראות כי הוא יותר קרוב למרכז של המסמכים הרלוונטים לטכנולוגיה (6-3=**3** < 11-6=5) לכן הוא יתוייג כמסמך המתאים לטכנולוגיה, בשונה מהתיוג האמיתי שלו - ספורט.

האלגוריתם מניח שהנתונים מסודרים על קו כלשהו בצורה הגיונית ופשוטה וישנה הפרדה ברורה בין מסמכים רלוונטים ולא רלוונטים. אך יש מקרים שמחלקות יצטרכו ייצוג יותר מורכב מאשר פרישת נקודות על חד-מימד ומציאת המרכז שלהם, ייתכנו חפיפות בין מחלקות שונות וכו'.  
דוגמא זו ממחישה את הבעיתיות במבנה בו האלגוריתם מסדר את המסמכים והאופן בו הוא מתייג אותם.

**2.א.**

א.תנו דוגמא לגרף מכוון, הכולל לפחות 4 צמתים ו-2 קשתות, כך שלכל צומת שלו מתקיים: ציון המומחיות (score authority (של הצומת הוא דרגת הכניסה שלו )degree-in )וציון הריכוזיות שלו )score hub )הוא דרגת היציאה שלו.

A diagram of a diagram

Description automatically generated

Initial Scores (כמו שנעשה בכיתה):

h(A)=h(B)=h(C)=h(D)=a(A)=a(B)=a(C)=a(D)=1

נתבונן בגרף זה ונחשב:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hub** | **authority** | **Degree Out** | **Degree In** | **Node** |
| 1(A->C) | 0 | 1 | 0 | **A** |
| 1(B->D) | 0 | 1 | 0 | **B** |
| 0 | 1(A->C) | 0 | 1 | **C** |
| 0 | 1(B->D) | 0 | 1 | **D** |

באמת ניתן לראות שלכל צומת מתקיים כי h(x)=Degree-Out(x) וגם כי a(X)=Degree-In(x) .

2.ב

ג.בהנתן הרשת הבאה:

A diagram of a diagram of a diagram

Description automatically generated

מהם ציוני ה hub וה- authority של כל דף? חשבו עד להתכנסות.נרמלו לפי שיטה שהוצעה בכיתה, ציינו במפורש מה הנרמול

\*\* מהתבוננות באינטרנט (וגם מהגיון) ניתן להסיק כי אין משמעות ללולאות עצמיות כמו שיש לצומת A. אם לדף כלשהו יש חץ לעצמו, הדבר לא מעיד על יתרון שיש לעמוד זה על פני עמוד אחר מפני שהוא זה שמפנה לעצמו . לכן בחישובים לא נתייחס אל קשת זו (A->A).

\*\* ננרמל כפי שהוצע בכיתה – פשוט ניקח את התוצאות (Authority/Hub) נסכום אותם והתוצאה תהיה הגורם נרמול, כאשר ניקח כל תוצאה של כל צומת ונחלק בגורם זה.

Initial scores (all set to 1):

h(A)=h(B)=h(C) = 1

a(A)=a(B)=a(C) = 1

**Iteration 1:**

**Authority update:**

   a(A) = h(C) = 1

   a(B) = h(A) + h(C) = 2

   a(C) = h(A) + h(B) = 2

   Normalize: Sum = 1 + 2 +2 = 5

   a(A) = 1/5 = 0.2, a(B) = 2/5 = 0.4, a(C) = 2/5 = 0.4

**Hub update:**

   h(A) = a(B) + a(C) = 2/5 + 2/5 = 4/5

   h(B) = a(C) = 2/5

   h(C) = a(A) + a(B) =1/5 + 2/5 = 3/5

   Normalize: Sum = 4/5 + 2/5 + 3/5 = 9/5

   h(A) =, h(B) =, h(C) =

**Iteration 2:**

**Authority update:**

   a(A) = h(C) = 3/9

   a(B) = h(A) + h(C) = 7/9

   a(C) = h(A) + h(B) = 6/9

   Normalize: Sum = 3/9 + 7/9 + 6/9 = 16/9

   a(A) = , a(B) = , a(C) =

**Hub update:**

   h(A) = a(B) + a(C) = 7/16 + 6/16 = 13/16

   h(B) = a(C) = 6/16

   h(C) = a(A) + a(B) = 3/16 + 7/16 = 10/16

   Normalize: Sum = 13/16 + 6/16 + 10/16 = 29/16

   h(A) =, h(B) =, h(C) =

**Iteration 3:**

**Authority update:**

   a(A) = h(C) = 10/29

   a(B) = h(A) + h(C) = 23/29

   a(C) = h(A) + h(B) = 19/29

   Normalize: Sum = 10/29 + 23/29 + 19/29 = 52/29

   a(A) = , a(B) = , a(C) =

**Hub update:**

   h(A) = a(B) + a(C) = 23/52 + 19/52 = 42/52

   h(B) = a(C) = 19/52

   h(C) = a(A) + a(B) = 10/52 + 23/52 = 33/52

   Normalize: Sum = 42/52 + 19/52 + 33/52 = 94/52

   h(A) =, h(B) =, h(C) =

כעת נשים לב כי בין התוצאות של איטרציה 2 ו-3 היו שינויים קטנים מאוד, ולכן נסיק כי בשלב זה התכנסנו.

התוצאות שקיבלנו לבסוף הן:

**h(A) =, h(B) =, h(C) =**

**a(A) = , a(B) = , a(C) =**

**3.א.**

במהלך בניית הזחלן השתמשנו בשני טכנולוגיות מעניינות – BeautifulSoup, Pandas.

**BeautifulSoup –**

BeautifulSoup היא ספריית Python המשמשת לניתוח מסמכי HTML ו-XML. ספרייה זו מאפשרת ביצוע של תהליך של web scraping בכך שהיא מאפשרת לחלץ ולתפעל בקלות נתונים מדפי אינטרנט (HTML). בזחלן שבנינו לשליפת נתונים מאתר Audible, הספרייה אפשרה לנו "לחפש" ספרים ספציפיים ולאסוף עליהם מידע כמו כותרות של הספרים, שמות הסופרים, דירוגים הספרים ותיאוריהם על ידי איתור רכיבי HTML ספציפיים והתוכן שלהם.

**Pandas –**

Pandas היא ספריית Python שמאפשרת מניפולציות שונות וניתוח נתונים בכמויות גדולות. ספרייה זו מספקת מבני נתונים כמו DataFrames המקלים על טיפול וניתוח מאגרי נתונים גדולים. בזחלן שאנו בנינו, Pandas משמשת להמרת רשימות של נתוני ספרים ל-DataFrames, מה שמאפשר לנו לפשט פעולות כמו סינון, מיון ואיסוף של הנתונים. בנוסף, ספרייה זו מאפשרת בייצוא נתונים לקבצי Excel בצורה קלה.

**ב.**



כאשר ביצענו מדידה של הזמנים, הזמן שלקח לכלל השאילתות לרוץ כולל יבוא של הנתונים הוא 7.74 שניות.

כעת נציג במה תלוי זמן הריצה וכיצד ניתן לשפר (אם ניתן) את זמן הריצה לכל גורם:

1. **הפעולות שמבצעים על הנתונים** – כמובן שזמן הריצה מושפע ישירות מהפעולות אותם מבצעים על הנתונים שאותם ייבאנו. במידה ועושים פעולות מסובכות בכמויות גדולות, כמובן שהדבר עלול להשפיע על זמני הריצה משום שחישובים אלו עלולים לקחת זמן.

במקרה שלנו ביצענו פעולות כמו סינון, מיון ואיסוף של הנתונים.

לפעולות אלו דאגנו להשתמש בספריית Pandas אשר ידוע שזו ספרייה יעילה מבחינת ביצועי זמן ריצה המבצעת את הפעולות הנדרשות ביעילות מרבית.

1. **ביצועי הרשת** – כמובן שזמן הריצה מושפע ישירות מהביצועים של הרשת בין המחשב לשרת שממנו אנו מביאים את הנתונים (מהירות הרשת וזמן התגובה של השרת). במקרים כאלו יותר קשה לשפר את הביצועים, אמנם ניתן לחסוך ככל הניתן בבקשות לשרת שממנו מביאים את הנתונים על מנת לחסוך את הזמן שכרוך בהבאתם.
2. **כמויות הנתונים –** עיבוד כמות גדולה של נתונים תגדיל את זמן הייבוא שלהם וזמן הניתוח שמבצעים עליהם. בשאילתות שביצענו, כמות הנתונים תלויה באופן ישיר משם הספר אותו אנו מחפשים. עבור ספרים מסוימים, כמות התוצאות שהוחזרו גדולה משמעותית מספרים אחרים ובכך יש השפעה ישירה על זמן ביצוע השאילתה.

יתר על כן, שימוש במבני נתונים יעילים (כמו DataFrames) ואלגוריתמים יעילים מקצרים את זמן ביצוע השאילתה.

1. **משאבי המערכת –** בדומה ל-"ביצועי הרשת", זמן הניתוח והעיבוד על הנתונים אותם ייבאנו תלוי באופן ישיר ממשאבי המערכת כגון: סוג המעבד, כמות הזיכרון וכו' .

אנו הרצנו את קוד הזחלן בGoogle Colab אשר משתמש בשרתים של גוגל על מנת להריץ את הקוד. כמובן שניתן להשתמש ב"מחשבים" חזקים יותר בעלי משאבים איכותיים יותר על מנת לשפר את זמן ביצוע השאילתה.

ג. בכלל התוצאות של השאילתות, אנו מבצעים בקשה מהאתר Audible, כאשר האתר כשלעצמו מהווה Hub לספרים משום שהוא מאגד ומקשר למספר רב של ספרים, כולל ספרים ממקורות שונים ומספק נתונים ומידע עליהם. כאשר אנו מקבלים מספר ספרים בתור תוצאה לשאילתה, כל ספר בפרט הוא Authority משום שהוא מהווה מקור חשוב ומוסמך בתחום שלו, עם דירוגים, ביקורות ומידע איכותי שמצביע על הרלוונטיות והסמכותיות שלו בתחום הנושא שבו עוסק הספר.

ד.

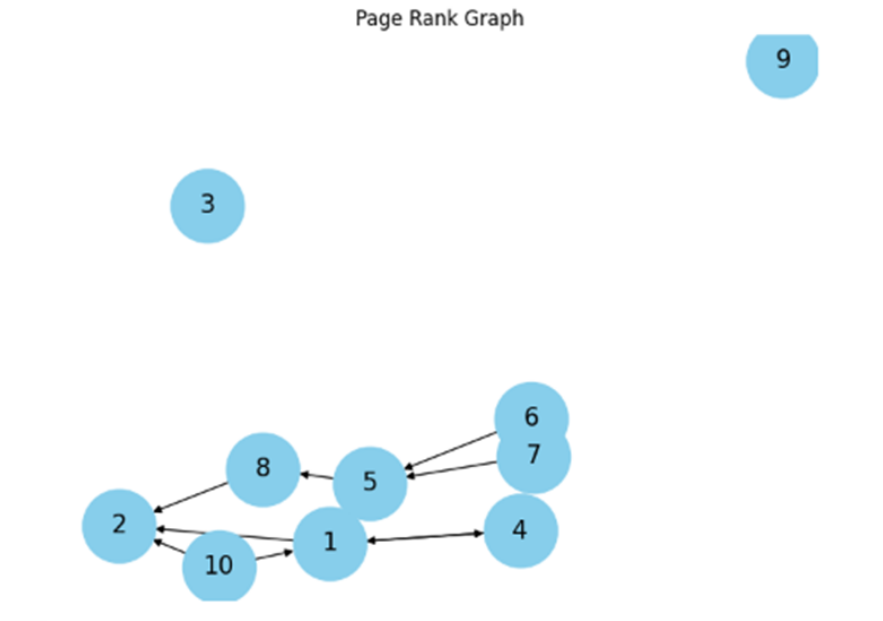
**Query: descriptions of fully valid books related to 'It Ends with Us'**

**10 ספרים אשר חזרו לנו לשאילתה כאשר חיפשנו את הספר "It ends with us" הם:**

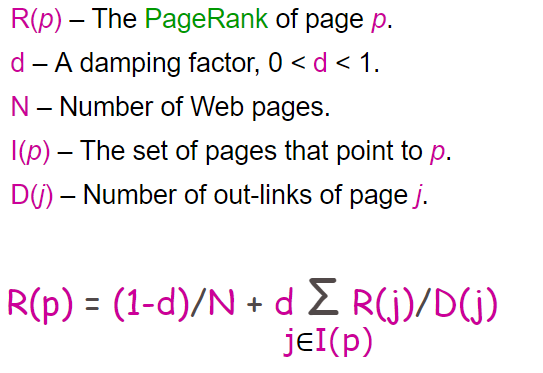
|  |
| --- |
| **Book name** |
| A Stolen Life |
| Bad Mormon |
| Healing the Shame That Binds You |
| Unf--k Your Brain |
| It Happened One Summer |
| George Orwell’s 1984 |
| It Ends with Violence |
| A Court of Thorns and Roses |
| James Patterson, David Ellis |
| Think Twice |

כעת, עבור כל ספר נגיד כמה קישורים יוצאים ממנו וכמה ספרים (מאלו שצוינו למעלה) מצביעים עליו.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Book Number** | **Book name** | **Total Number of out-links** | **Number of points to me** | **Books number I link to** |
| 1 | A Stolen Life | 2 | 3 | [2, 4] |
| 2 | Bad Mormon | 0 | 3 | [] |
| 3 | Healing the Shame That Binds You | 0 | 0 | [] |
| 4 | Unf--k Your Brain | 1 | 1 | [1] |
| 5 | It Happened One Summer | 2 | 2 | [1, 8] |
| 6 | George Orwell’s 1984 | 2 | 1 | [7, 5] |
| 7 | It Ends with Violence | 2 | 1 | [6, 5] |
| 8 | A Court of Thorns and Roses | 1 | 1 | [2] |
| 9 | James Patterson, David Ellis | 0 | 0 | [] |
| 10 | Think Twice | 2 | 0 | [1, 2] |



על מנת לחשב השתמשנו בסקריפט קיים (סקריפט שרץ בGoogle Colab, בשם WebScraperAudible.ipynb בתחתית הקובץ – הגשנו את קובץ זה).



A close up of black text

Description automatically generatedכפי שראינו בהרצאה מקובל להשתמש בd = 0.85 לכן נגדיר ערך זה.

N = 10 (מספר הדפים)

R(j) של איטרציה מספר 1 (לכל j) = 1/10

על מנת לקבוע את כמות האיטרציות הנדרשות ביצענו לכתחילה 100 איטרציות אך לא ראינו שינוי בין איטרציה 50 ל100, לכן לאחר בדיקה משוערת, במקרה שלנו, מספיק לגרף שלנו 30 איטרציות.

נראה את התוצאות כפי שיצאו מהסקריפט:

Page ranks for iteration 1:

['0.248322', '0.248322', '0.020134', '0.077181', '0.134228', '0.077181', '0.077181', '0.077181', '0.020134', '0.020134']

Page ranks for iteration 30:

['0.274901', '0.258965', '0.020113', '0.176878', '0.073422', '0.046768', '0.046768', '0.061958', '0.020113', '0.020113']

**Final PageRank after 30 iterations:**

Page 1: 0.274901

Page 2: 0.258965

Page 3: 0.020113

Page 4: 0.176878

Page 5: 0.073422

Page 6: 0.046768

Page 7: 0.046768

Page 8: 0.061958

Page 9: 0.020113

Page 10: 0.020113

נראה דוגמא לחישוב עבור איטרציה 1 עבור ספר מס' 1:

נחשב את התרומה של כל ספר שמצביע על ספר מס' 1 לדירוג של 1:

ספרים 4,5,10 מצביעים לספר 1:

ספר 4:

ספר 5:

ספר 10:

לכן בסה"כ:

ולאחר חישוב כלל הדירוג של הספרים והנרמול נקבל:

**ה.**

על מנת לענות על סעיף זה, בחרנו 2 משתמשים אשר נתנו להם 10 מסמכים והם התבקשו לבחור אילו מסמכים רלוונטיים עבורם ואילו לא רלוונטיים עבורם.

המסמכים אשר ניתנו להם הם:

|  |
| --- |
| A Stolen Life |
| Bad Mormon |
| Healing the Shame That Binds You |
| Unf--k Your Brain |
| It Happened One Summer |
| George Orwell’s 1984 |
| It Ends with Violence |
| A Court of Thorns and Roses |
| James Patterson, David Ellis |
| Think Twice |

המשתמש הראשון בחר בדפים הרלוונטיים : 1,2 – סה"כ 2 מסמכים רלוונטיים.

המשתמש השני בחר בדפים הרלוונטיים :1,4,7 – סה"כ 3 מסמכים רלוונטיים .

**Precision & Recall:**

על מנת לחשב את הPrecision אנו נצטרך לקחת את כמות הדפים הרלוונטיים (של כל משתמש) ולחלק בכמות הספרים שניתנו למשתמש (10).

על מנת לחשב את הRecall אנו נבדוק כמה ספרים חוזרים לשאילתת החיפוש עבור הספר Its end with us וככה נוכל לבצע את החישוב. ניקח את התוצאה ונחלק בסך כמות הספרים שחוזרים לחיפוש ספר זה (43).

|  |  |  |
| --- | --- | --- |
| **User 2** | **User 1** |  |
| 3/10 | 2/10 | **Precision** |
| 3/43 | 2/43 | **Recall** |

על מנת לשפר את התוצאות נבחן בנפרד Precision וRecall.

על מנת לשפר את הPrecision, כאשר מספר הספרים שנתן כל פעם למשתמש הוא 10, מספיק פשוט לחפש בשאילתא ספר "שיותר קרוב" לקטגוריה ולסוג הספרים שאותם המשתמש מעדיף ומעוניין למצוא באתר. כך ה"מונה" יגדל, והמכנה נשאר קבוע (10) ובכך הPrecision יגדל.

לעומת זאת, על מנת לשפר את הRecall, נרצה גם להגדיל מונה וגם להקטין מכנה.

כלומר במקרה כזה, נרצה כי באמת נחפש ספר אשר יחזיר מספר תוצאות רב שרלוונטיות למשתמש (ספר שכנראה מאותה הקטגוריה שהוא מחפש) אך באותה המידה גם נרצה להקטין את המכנה, דבר שדורש שעבור ספר זה לא יתקבלו הרבה תוצאות, לשם כך רלוונטי אולי לעשות חיפוש יותר ממוקד הקולע לשם הספר או קרוב לכך, ולא חיפוש של מילים כלליות מתוך שם הספר (לדוגמה לא לעשות חיפוש עבור המילה “Love”).

ו.

קישור לתיקיית גיט : <https://github.com/avishayhersh18/DataDiggers-HW3-3>

קישור ל-git pages: <https://avishayhersh18.github.io/DataDiggers-HW3-3/>