**אחזור מידע – פרויקט להגשה – אביב התשפ"ד**

**מגישים:**

**מור שמואל 315040980**

**מעיין אביטן 313550907**

**אופק בן אברהם 313310559**

**רותם פורת 206607731**

<https://github.com/avtn96/Information_Retrieval.git>  
<https://project-information-gamma.vercel.app/final-project>

מהו תחום העיסוק המרכזי של האתר? מהו המידע הזמין למשתמשי האתר? ענו בפסקה אחת. צרפו את הקישור לאתר.

האתר שבחרנו הוא rottentomatoes:

<https://www.rottentomatoes.com>

האתר מתמחה בדירוג והערכה של סרטים ותוכניות טלוויזיה. הוא מספק למשתמשים סקירות ביקורתיות, ציון ממוצע לפי ביקורות של מבקרים וקהל, ותמונות ומידע נוסף על סרטים ותכנים טלוויזיוניים. האתר כולל גם רשימות מומלצות ודרוגים פופולריים, ומאפשר למשתמשים לבדוק ולהשוות בין תכנים שונים.

רשמו שלוש שאילתות מעניינות שהייתם רוצים לקבל עליהן תשובה באתר, והאתר אינו **עונה עליהן** כעת. עבור כל שאילתא כזו, ציינו מהם פרטי המידע הנדרשים לצורך מענה על השאילתא.

שלוש שאילתות מעניינות שהיינו רוצים לקבל עליהן תשובה באתר והאתר אינו עונה עליהן כעת:

**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.

בנו זחלן המחזיר את התוצאות לשאילתות שהגדרתם. מומלץ (אך לא חובה) לבנות את הזחלן בפייתון. הציגו את קוד הזחלן.

קובץ הזחלן נמצא ביחד עם קבצי הקוד בגיט.  
קישור לקוד שמבצע זאת: <https://github.com/avtn96/Information_Retrieval/blob/main/scraper.py>

רשמו טכנולוגיות שונות מעניינות שהשתמשתם בהם בפרויקט.

**asyncio:** ספריית Python לניהול אסינכרוניות, שאפשרה לנו להריץ מספר פעולות במקביל בצורה יעילה יותר, במיוחד כאשר מדובר בביצוע שאילתות רשת.  
**pyppeteer:** כלי זה משמש כאוטומציה לדפדפן (מבוסס על Puppeteer ל-Node.js), ומאפשר לנו להפעיל את דפדפן Chrome מתוך Python ולבצע פעולות כגון טעינת עמודים ולקיחת תוכן HTML בצורה תוכניתית.  
**BeautifulSoup:** ספרייה לעיבוד וניתוח של תוכן HTML/XML, אשר שימשה אותנו לשליפת מידע מתוך מבנה ה-HTML של הדפים שסרקנו.  
**pandas:** ספריית Python לניתוח ועיבוד נתונים, שאפשרה לנו לבנות DataFrame מכל המידע שנאסף, לבצע פעולות עיבוד נתונים, ולשמור את הנתונים לקובץ CSV. טכנולוגיות אלו אפשרו לנו לבצע סקריפט סריקה מתקדם, לעבד את הנתונים שנאספו, ולהפיק תובנות מפורטות מהאתר Rotten Tomatoes.

כמה זמן רצו השאילתות שלכם?במה זה תלוי? האם לדעתכם ניתן לשפר זמן זה?

השאילתות רצות בערך כשנייה בגלל שהם ממוקדות והאינדקס מוגדר היטב, ריצת האינדקס תלויה בקובץ csv, ככל שהוא גדול יותר כך יידרש זמן לסרוק ולהחזיר את התוצאות המתאימות, בנוסף זה תלוי בתכנון האינדוקס ובשאילתות, כלומר אם השאילתות מכילות תנאים מורכבים(סינונים למינהם) כך ידרש יותר זמן לסינון התוצאות.   
אנו בטוחים שיש טכנולוגיות מתקדמות יותר שעושות אופטימיזציה לקוד שלנו, שמירה בזכרון מטמון, שימוש במבני נתונים יעילים, חלוקה לתהליכים שיפעלו במקביל, לכן ניתן לשפר זמן זה.

בנו inverted index ל- 15 המילים הנפוצות שחזרו (מספיק להתיחס ל -20 הדפים הראשונים שחזרו)

קובץ inverted index נמצא ביחד עם קבצי הקוד בגיט.  
קישור לקוד שמבצע זאת: <https://github.com/avtn96/Information_Retrieval/blob/main/inverted_index.py>

בחרו את אחת השאילתות, וחשבו tfIdf של המושגים בשאילתא.

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>

האם בדפים שהוחזרו קיימים hubs? Authorities? נמקו.

בדפים שהוחזרו קיימים Authorities וגם hubs משום שהדפים מקושרים להרבה דפים חשובים אחרים וגם את הדפים מקשרים הרבה דפים אחרים חשובים ולכן זה גם Authorities וגם hubs.

דוגמה: הדף - <https://www.rottentomatoes.com/tv/tracker_2023/s01>

מוביל להרבה דפים אחרים בתוך rottentomatoes ומצד שני יש כמה דפים שמזכירים את הקישור לדף כמו- <https://www.rottentomatoes.com/browse/tv_series_browse/sort:popular>

בחרו 10 דפים שהזחלן החזיר, ואשר יש ביניהם קישורים .חשבו pageRank לכל דף, מומלץ להעזר בסקריפט מוכן. הציגו את החישובים ואת הדירוג הסופי.

**השאילתא:** “movies from 2023 with duration > 1.5 h”

<https://www.rottentomatoes.com/m/asteroid_city>: 0.028942919002214944

<https://www.rottentomatoes.com/m/a_man_called_otto>: 0.05354440042450051

<https://www.rottentomatoes.com/m/the_contestant>: 0.028942919002214944

<https://www.rottentomatoes.com/m/the_hatchet_wielding_hitchhiker>: 0.237280124048726

<https://www.rottentomatoes.com/m/to_kill_a_tiger>: 0.041243659713357736

<https://www.rottentomatoes.com/m/this_much_we_know>: 0.23063032926193258

<https://www.rottentomatoes.com/m/boston_strangler_2023>: 0.14224714535737545

<https://www.rottentomatoes.com/m/killers_of_the_flower_moon>: 0.09867594704077756

<https://www.rottentomatoes.com/m/reptile_2023>: 0.06924627807445118

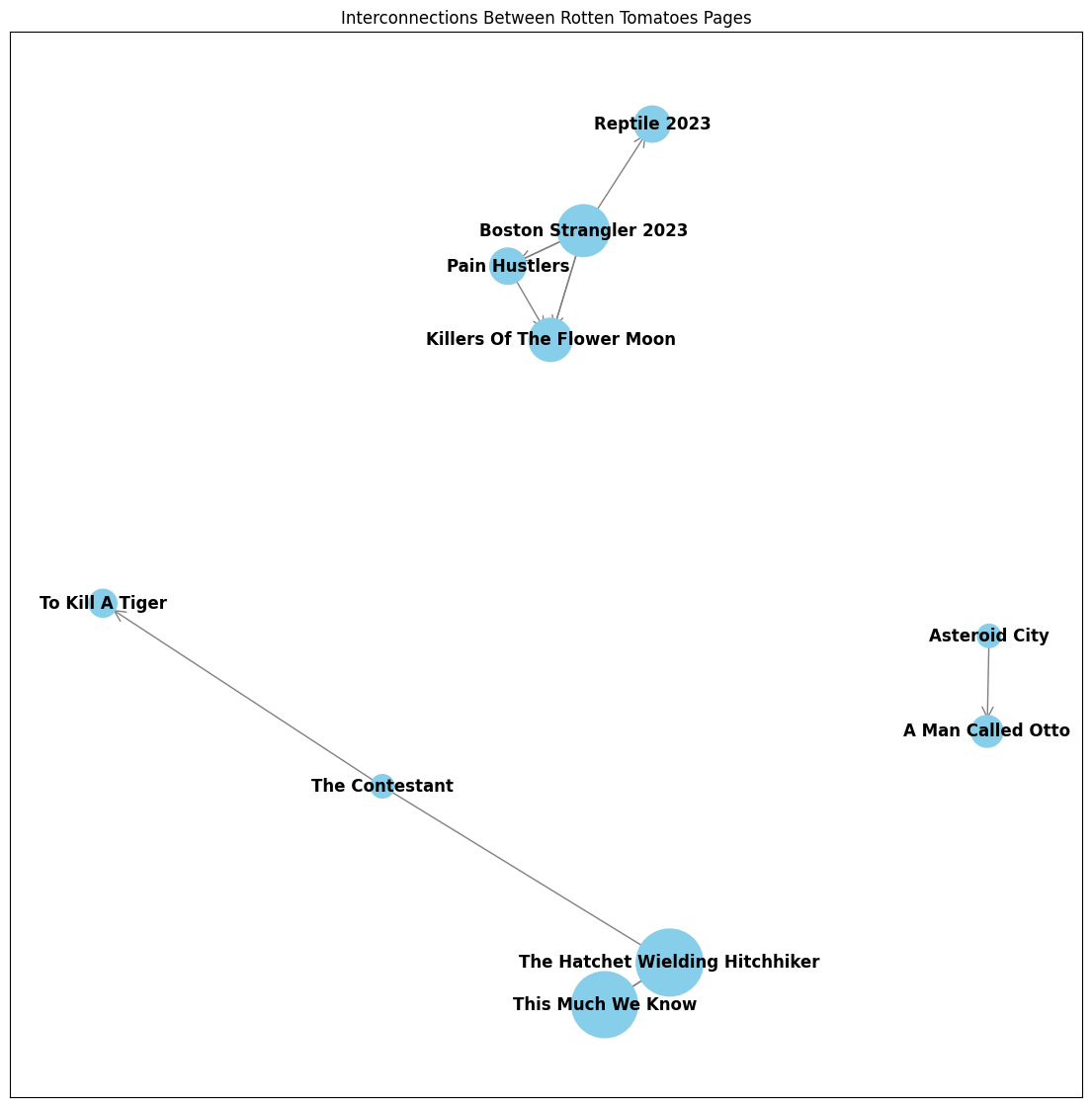
<https://www.rottentomatoes.com/m/pain_hustlers>: 0.06924627807445118

**שלבי החישוב של PageRank:**

יצירת מטריצת סמיכויות (Adjacency Matrix) : הקוד מייצר מטריצה שמייצגת את הקישורים בין הדפים. כל תא במטריצה מציין אם יש קישור מדף מסוים לדף אחר.

חישוב PageRank : משתמשים במטריצת סמיכות מנורמלת לחישוב ערכי PageRank לכל דף. כל דף מקבל ערך שמייצג את מידת החשיבות שלו כאשר ערך זה מחושב לפי הנוסחאות שלמדנו בכיתה, כאשר דפים עם יותר קישורים פנימיים מקבלים ערכים גבוהים יותר. האלגוריתם חוזר על עצמו מספר פעמים (עד למקסימום איטרציות שאצלנו מוגדר כ100) עד שהערכים מתכנסים לשינוי קטן מאוד מוגדר על ידי tol=1e-6.

הצגת תוצאות PageRank : התוצאות מודפסות על המסך כך שכל URL מוצג עם ערך הPageRank שלו.

****הדמיית הקשרים בין הדפים: הקוד מייצר גרף שמציג את הקשרים בין הדפים, כאשר גודל הצמתים משקף את ערך ה-PageRank של כל דף.

**התמונה להלן מדגימה את הקישוריות בין הדפים כאשר גודל כל צומת הוא לפי הדירוג.**קישור לקוד של ה pageRank : <https://github.com/avtn96/Information_Retrieval/blob/main/page_rank.py>

הראו לשני משתמשים שונים את הדירוג מהסעיף הקודם, בקשו מהם לסמן relevance feedback

**Ron:**

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

<https://www.rottentomatoes.com/m/a_man_called_otto> R

<https://www.rottentomatoes.com/m/the_contestant> NR

<https://www.rottentomatoes.com/m/the_hatchet_wielding_hitchhiker> R

<https://www.rottentomatoes.com/m/to_kill_a_tiger> R

<https://www.rottentomatoes.com/m/this_much_we_know> NR

<https://www.rottentomatoes.com/m/boston_strangler_2023> R

<https://www.rottentomatoes.com/m/killers_of_the_flower_moon> R

<https://www.rottentomatoes.com/m/reptile_2023> R

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

**Amit:**

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

<https://www.rottentomatoes.com/m/a_man_called_otto> R

<https://www.rottentomatoes.com/m/the_contestant> NR

<https://www.rottentomatoes.com/m/the_hatchet_wielding_hitchhiker> R

<https://www.rottentomatoes.com/m/to_kill_a_tiger> R

<https://www.rottentomatoes.com/m/this_much_we_know> R

<https://www.rottentomatoes.com/m/boston_strangler_2023> NR

<https://www.rottentomatoes.com/m/killers_of_the_flower_moon> R

<https://www.rottentomatoes.com/m/reptile_2023> R

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

**ניתן להציע שאילתא מותאמת על מנת לשפר תוצאות אלו.**

הצעות לשיפור השאילתה:

* אופטימיזציה לפי פידבק:

ניתן להגדיר שאילתה מותאמת אישית שמביאה בחשבון את הפידבק של המשתמשים, למשל מתן עדיפות לדפים שסומנו כרלוונטיים על ידי שני המשתמשים.

* הוספת מילות מפתח:

הוספת מילות מפתח המבוססות על התכנים שמעניינים את המשתמשים.

* אינטראקציה מתמשכת עם המשתמש:

ניתן להמשיך ולשאול את המשתמשים על תוצאות אחרות כדי לשפר את איכות השאילתה.

**Ron:**

Precision= 7/10 = 0.7

Recall = 7/76 = 0.0921

**Amit:**

Precision = 6/10 = 0.6

Recall = 6/76 = 0.0789

הציגו את כל התוצרים של סעיפים א-ו בדף HTML יחיד. הדף יכול להיות דינמי או סטטי (כלומר מציג צילומי מסך של התשובות לשאלות). העלו את הדף לgit pages. צרפו קישור לדף שהעליתם.

**קישור לאתר:** <https://project-information-gamma.vercel.app/final-project>

בתאריך 12.8 תציגו את הדף מסעיף 11, הכולל את סעיפים 1-10. תקבלו משוב מחבריכם. ענו בטבלה:

|  |  |
| --- | --- |
| **איזה שינוי הוצע?** | **האם לדעתכם יש מקום לשיפור זה?** |
| סינון על פי ז׳אנר ולא לפי שנת יציאה | סינון על פי ז'אנר יאפשר למשתמשים למצוא תכנים שמתאימים להעדפותיהם האישיות בצורה טובה יותר. ניתן להוסיף פילטר נוסף המקטלג את הסדרות/סרטים לפי ז'אנר לצד שנת היציאה. |
| שימוש בספריית Pandas לצורך עיבוד נתונים מהיר יותר | אכן נעשה שימוש בספרייה זאת בקוד. |
| שילוב אלמנטים ויזואליים כמו תמונות להצגת הנתונים בצורה ברורה יותר | אלמנטים ויזואליים יכולים להפוך את המידע לנגיש ומובן יותר למשתמשים. הוספת גרפים, תמונות או דיאגרמות יכולה לשפר את חוויית המשתמש ולהבהיר את תוצאות החיפוש ואכן יש מקום לשיפור זה. |
| שיפור אלגוריתמים של חיפוש: פיתוח אלגוריתמים חכמים יותר שמבינים את כוונת המשתמש ואת ההקשר של השאילתה, תוך התבססות על למידת מכונה ובינה מלאכותית | שיפור האלגוריתמים באמצעות למידת מכונה ובינה מלאכותית יכול להוסיף ערך רב בכך שהמערכת תוכל להבין ולהתאים את החיפושים בצורה חכמה יותר לצורכי המשתמש. יש מקום לשיפור אך כפרויקט נפרד ולא בזמן הקצוב של הקורס. |
| לפלטר לפי מה הם מעדיפים שיהיה ממשק ממש טוב | שיפור הממשק בהתאם להעדפות המשתמשים יוכל להגדיל את השימושיות והאטרקטיביות של המערכת ולכן יש מקום לשיפור זה. |

מהו היו האתגרים בקורס/ פרויקט כצוות?

**אתגרים טכניים:**  התמודדות עם תקלות טכניות, מציאת אתר שלא חוסם אותך, בעיות בקוד, ולמידת טכנולוגיות חדשות.

**ניהול זמן בסמסטר קצר:**  תכנון קפדני של הזמן כדי לעמוד בלוחות הזמנים ולסיים את כל המשימות בזמן, הוספה של יותר באפרים עקב מילואים, פרויקטים וכו׳.

**קושי במציאת אתר מתאים :** אתגר נוסף היה במציאת אתר שיכיל מספיק מידע רלוונטי ויאפשר ביצוע חיפושים וניתוחים מורכבים.

**מציאת שאילתות שהאתר לא עונה עליהן:**  יצירת שאילתות ייחודיות שמנצלות את המידע הקיים באתר, אך חושפות פערים שהאתר לא נותן עליהן מענה ישיר.