פרויקט מנוע חיפוש

דו"ח חלק ב'

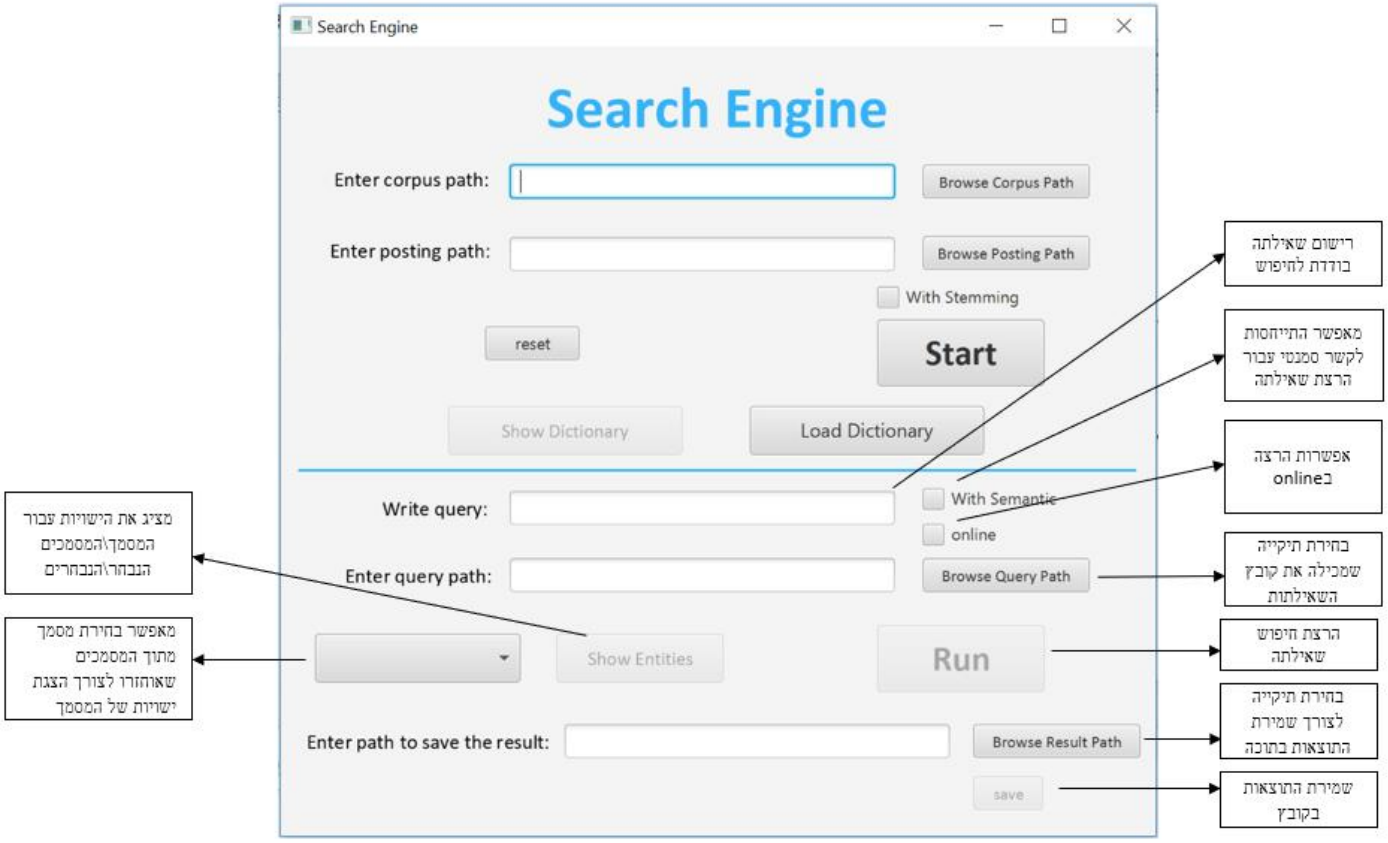
מגישות:

פז יונה 312204340

מיטל יניב 307938969

1. אופן פעולת המנוע

אנו מתייחסות אך ורק לשינויים מחלק א'



עבור הוראות הפעלה יש לקרוא את קובץ README המצורף.

מחלקות נוספות שיצרנו בmodel package:

א) המחלקה Searcher

מטרת המחלקה הזו היא לחפש שאילתה בודדת או קובץ שאילתות מתוך הpostings והמילון שנוצרו בשלב הפירסור והאינדוקס. לצורך כך המחלקה נעזרת במחלקת הRenker ובטעינת כל המידע הרלוונטי מהאינדקסר הכללי וה postings. המחלקה שולחת את השאילתא לפרסור ומחזירה את המסמכים הרלוונטיים ביותר לשאילתא באופן מדורג. (לא יותר מ-50 מסמכים עבור כל שאילתא). בנוסף, המחלקה משתמשת באלגוריתם לטיפול סמנטי שמטרתו להבין את הקשר הסמנטי בין השאילתא למסמכים.

הפונקציות במחלקה:

1) search – פונקציה זו תקבל את השאילתא והשדות המסומנים עבורה, תאתחל את הרשימות ששמרנו בשדות, ולאחר מכן שולחת למחלקת הparse לפונקציית הפירסור את השאילתה ובמידת הצורך, אם המשתמש בחר להיעזר בקשר סמנטי, השאילתה תישלח לפירסור עם המילים שקשורות סמנטית למילות השאילתה (בעזרת שירשור הterms שיחזרו מפונקציית הuseSemantic). נשמור את המילים שפורסרו ברשימה- parseTerms. נשלח רשימה זו לפונקציה createListOfTerms

(מפורטת בהמשך), ונשלח את הרשימות שנוצרו למחלקה Ranker שתדאג לדרג לנו את השאילתא עבור כל מסמך שנמצא רלוונטי. את התוצאות נשמור ברשימה docByRank אותה נמיין ונשמור את 50 המסמכים שדורגו ראשונים.

2) semanticSynonymTerm- פונקציה זו מאפשרת את השימוש בקשר סמנטי. הפונקציה תקבל משנה בוליאני online, כך שאם אנחנו בonline: בעזרת בקשת http נשתמש בapi וממנו נחלץ את המילים הנרדפות לכל אחת ממילות השאילתה. לבסוף נחזיר את המילים הנרדפות הללו. בחרנו להגביל את כמות המילים הנרדפות עבור כל שאילתה לצורך ייעול זמנים.
אם אנחנו בoffline: נחלץ את המילים מקובץ טקסט שיצרנו ונוסיפם למילים הקיימות של השאילתא.

3) sortFiftyDocs- הפונקציה ממיינת את רשימת המסמכים הרלוונטים על פי הדירוג שלהם- מהגדול לקטן, לאחר מכן מוסיפה לרשימה חדשה fiftySortedDocs את 50 המסמכים הראשונים (שקיבלו את הדירוג הגבוה ביותר). במידה ומספר המסמכים שאוחזרו קטן מ50, הפונקציה תחזיר את הרשימה כפי שהיא.

4)createListOfTerm - פונקציה זו מחלצת את שורת הposting המתאימה בעזרת הפוינטר שנמצא במילון הראשי עבור כל term שנמצא בשאילתא. לאחר הבאת השורה מקובץ הposting לזיכרון, "נשחזר" את הterm מהמילון הראשי, לצורך נוחות בשליפת הנתונים כלומר ניצור term ובו פרטי המידע בהם אנו משתמשות לצורכי הדירוג- המסמכים בהם הterm מופיע ומספר החזרות של כל term במסמך. איך נבצע זאת? תחילה נחפש את המילה במילון האינדקסר, משם נחלץ את הנתיב לקובץ הposting המתאים ואת הpointer לשורה הרלוונטית בposting. לאחר קריאת הקובץ, וקריאת השורה הרצויה- ניצור אובייקט term. עבור על term שיצרנו ניצור את רשימת מסמכים בהם המילה נמצאת. עבור כל מסמך נשמור את שם המסמך וכמות המופעים של המילה במסמך. בדרך זו נחסוך את הגישה לדיסק בכל פעם לצורך קריאת שורה מהposting וכל המידע הנחוץ יישמר בזיכרון.

5) readQueries- מטרת פונקציה זו היא לייצג את מסמך השאילתות ולטפלו בהתאם. פונקציה זו פותחת את קובץ השאילתות, ושולחת כל שאילתה בנפרד ל Searcher לצורך חיפושה ומתן דירוג מתאים עבורה. לשאילתה עצמה אנו משרשרות את הdescription של כל שאילתה לצורך שיפור תוצאות האיחזור. את תוצאות הדירוג נשרשר לשדה listOfAllQueryResults המייצג רשימה של string לפי הפורמט הרצוי ע"מ שנוכל לכתוב את התוצאות לקובץ עבור כל השאילתות בצורה נוחה ופשוטה יותר. בנוסף, על מנת להציג את כלל המסמכים הרלוונטים נשמור רשימה של HashSet הנקראת allDocForQueriesFile אליה נוסיף את המסמכים הרלוונטים של כל שאילתא תוך טיפול בכפילויות.

6)writeToQueryResultsFile - פונקציה זו תקבל את הרשימה ששמרנו הreadQueries ונתיב לתיקיה בה נשמור מסמך עם התוצאות של הרצת קובץ השאילתות במלואו, שם המסמך בו ישמרו התוצאות נקרא "results.txt".

בנוסף, קיימים Setters&Geterrs שמסייעים בשליפה ועדכון המידע הנחוץ.

ב) המחלקה Ranker

מטרת המחלקה הזו היא לייצר את דירוג השאילתה לכל מסמך על פי קריטריונים שונים עליהם נפרט בהמשך.

<u>הפונקציות במחלקה:</u>

1) rankByDate- פונקציה זה נותנת דירוג למסמך ע"פ תאריכו. כלומר, נרצה שמסמך שפורסם בשנת 1994 יאוחזר בחשיבות נמוכה יותר לעומת מסמך שפורסם בשנת 1998. לצורך כך, חישבנו את המרחק בין תאריך פירסום המסמך לבין תאריך עתידי ולבסוף נרמלנו (נפרט על כך בחלק הבא). לבסוף נחזיר את התוצאה.

2) fillDates- פונקציה זו מוסיפה למילון הdates את החודשים בשנה בצורות שונות (אותיות קטנות\גדולות\קיצורים) עם הערך שנתנו להם לצורך מתן דירוג לתאריך המסמכים.

3) BM25Function- פונקציה שמקבלת מסמך ומחשבת את הדירוג של מסמך בהתאם לנוסחת BM25 ומחזירה את התוצאה. על הנוסחה נפרט בחלק הבא.

4) rankByTitle - מקבלת מסמך ובודקת אם אחת ממילות מהשאילתא מופיעה בכותרת המסמך, במידה וכן תחזיר 1.

5) rankByEntity – מקבלת מסמך ובודקת אם אחת ממילות מהשאילתא מוכלת בישות מתוך 5 הישויות החזקות במסמך, במידה וכן תחזיר 1.

בנוסף, קיימים Setters&Geterrs שמסייעים בשליפה ועדכון המידע הנחוץ.

2. מחלקות מחלק א' שביצענו בהן שינויים:

1) Document– הוספנו נתונים נוספים על כל מסמך על מנת לשמר את המידע הרלוונטי לנו לדירוג, למשל כותרת מסמך, 5 אישיות דומיננטיות ודירוגן, אורך מסמך. את הנתונים הללו הדפסנו לקובץ posting ייעודי על מנת לטעון אותם בהמשך.

2) ReadFile– הוספנו רשימת docs המכילה מסמך עם כל הנתונים הרלוונטים עבורנו (אישיות, כותרת, תאריך, אורך ומספר מסמך). בנוסף, מכוון שנתקלנו בבעיות זיכרון החלטנו לאפס רשימות ולנקות את הזיכרון על ידי clear לparse אחרי כל 40 מסמכים.

3) parse- שיפרנו את פונקציה הparse הראשית של המחלקה על ידי טיפול במקרי קצה שנתקלנו בהן אחרי חלק א'.

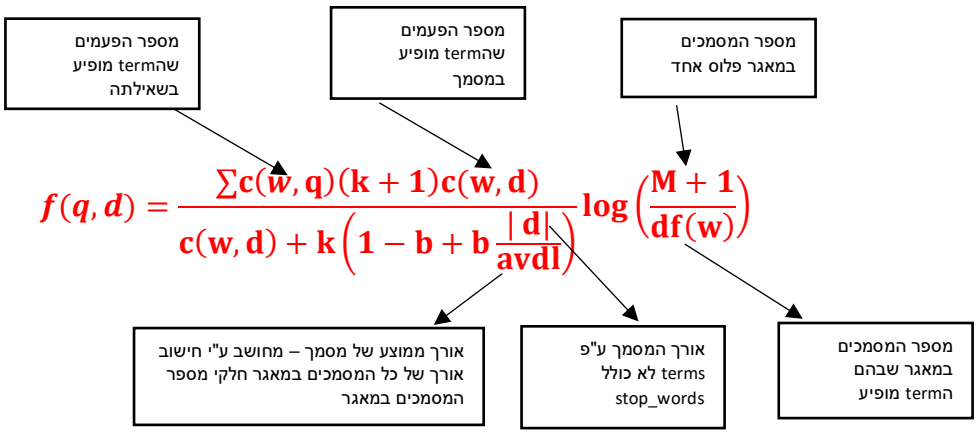
3. אלגוריתמים שכלולים במנוע

אלגוריתם הדירוג:

כולל בתוכו את נוסחת הbm25 (כולל התייחסות לterm שמופיעים בכותרת של מסמך), התייחסות לdescription של שאילתה מהקובץ, התייחסות לתאריך פרסום המסמך, כותרת המסמך ואישיויות הדומיננטיות במסמך.

ובאופן מפורט:

- **BM25:**



בחרנו בk שערכו 2 וb שערכו 0.8, לאחר הרבה ניסיונות הרצה שמנו לב כי ערכים אלו הביאו לנו את האחזור הטוב ביותר.

דוגמא להשפעה של האלגוריתם:

נניח ובמאגר ישנם 2 מסמכים-d1,d2 .

ונניח אנו מריצות שתי שאילתות, כך ש q1=paz , q2=meytal.

נניח ו-term=paz מופיע 3 פעמים במסמך d1, ו5 פעמים במסמך d2.

נניח ו-term=meytal מופיע 3 פעמים במסמך d1, ו6 פעמים במסמך d2.

אורך המסמך d1 הוא 70, אורך המסמך d2 הוא 94.

נקבל כי עבור המסמך d1 ערך הbm25 הוא 0.5652=0.2826+0.2826

ונקבל כי עבור המסמך d2 ערך הbm25 הוא 0.6269=0.308+0.3189

ולכן, נרצה להחזיר את מסמך d2 בעדיפות גבוהה יותר (כי הוא מביא תוצאה טובה יותר).

- **התייחסות לdescription של שאילתה:**

את הterms בdescription שרשרנו לשאילתה עצמה, ראינו לנכון לעשות זו כי terms אלו מייצגים "ומחזקים" את השאילתה, בנוסף, רוב מילות השאילתה חזרו על עצמן בdescription ולכן זה שיפר את הערכים (בעזרת שדה ה-c(w,q) ואחזר מסמכים בצורה מדויקת ונכונה יותר.

- **התייחסות לתאריך פרסום המסמך:**

בחרנו תאריך עתידי ונתנו לו ערך, ממנו אנו מחסירות את ערך תאריך פרסום המסמך. ככל שההפרש הזה ייצא קטן יותר, כך נדע שמדובר במסמך שפורסם בסמוך יותר לתאריך העתידי בעזרת נרמול.

שנה קיבלה אצלנו ערך של השנה כפול 390, למשל שנת 2020 קיבלה את הערך 2020*390= 787,800.

כל חודש מקבל ערך של 32 יותר מקודמו. כלומר, חודש ינואר יקבל ערך של 32, וחודש פברואר יקבל ערך של 64.

כל יום קיבל את ערכו ללא שינוי.

את הערך של השנה, חודש ותאריך סכמנו וזהו הייצוג של התאריך.

הערך העתידי קיבל את הערך שתואם לתאריך ה1\2\2020.

נקודה חשובה- שמנו לב כי בחלק מהמסמכים אין תאריך מפורט ולכן בחרנו בערכים דיפולטיביים כדי להימנע מנפילות או מתן חשיבות גדול מידי לתאריך פרסום המסמך.

כדי לנרמל את התוצאה עשינו 1.5 חלקי ההפרש שקיבלנו, כלומר:

$$\sum_{d=1}^{relevantDocs} \frac{1.5}{future\ date\ value - publication\ date\ value}$$

מדוע בחרנו בערכים הללו עבור יום, חודש ושנה?
לאחר התלבטויות רבות של איך לייצג את התאריך בצורה כזו שאין סיכוי שתאריך מוקדם יותר יביא תוצאה טובה יותר, בחרנו ללכת לפני סדר החשיבויות, ליום חשיבות נמוכה ביותר ולכן קיבל את ערכו ללא שינוי. כיוון שבחודש ישנם לכל היותר 31 ימים, נתנו לכל חודש ערך של 32 במינימום(עבור חודש ינואר), וכל חודש נוסף קיבל תוספת של 32 וכך אנו מבטיחות שככל שהחודש יותר מאוחר

בשנה, ערכו גדול יותר. ומכיוון שבשנה ישנם 12 חודשים, הערך שלה חייב להיות גדול מ384 (12*32) ולכן החלטנו לעגל כלפי מעלה ל390.

לסיכום, מכיוון שראינו כי אנו מגיעות לערכים גדולים יחסית מבחינת ההפרש בין התאריך העתידי לתאריך פרסום המסמך, בחרנו לנרמל את התוצאה עם ערך שגדול מ1. לאחר ניסוי וטעייה הערך שהתאים ביותר היה 1.5. וכך, ככל שתאריך קרוב יותר לתאריך העתידי, זה אומר שהוא חדש יותר. ההפרש יהיה קטן יותר וכאשר נחלק את 1.5 בערך הזה, נקבל תוצאה גדולה יותר לעומת מסמך שפורסם בתאריך ישן יותר.

דוגמא להשפעה של האלגוריתם:

נניח ועד כה, הרצנו חיפוש שאילתה על מאגר מאוד קטן, וחזרו לנו בסה"כ 2 מסמכים רלוונטיים לשאילתה. תאריך הפרסום של המסמך הראשון הוא 1994\1\1, ותאריך הפרסום של המסמך השני הוא 1994\3\3,
נקבל עבור המסמך הראשון ערך מבחינת התאריך של 390*1994+32+1=777693,
ונקבל עבור המסמך השני ערך מבחינת תאריך של 390*1994+96+3=777759.
ההפרש מהתאריך העתידי עבור המסמך הראשון הוא 10,172
ההפרש מהתאריך העתידי עבור המסמך השני הוא 10,106
לאחר נרמול, הערך עבור המסמך הראשון הוא 0.0001474
לאחר נרמול, הערך עבור המסמך השני הוא 0.0001484
ניתן לראות כי עבור המסמך השני, הערך הוא גדול יותר, זאת כיוון שתאריך פרסומו קרוב יותר לתאריך העתידי.

- **התייחסות לכותרת של המסמך:**

עבור המסמך הרלוונטי נבדוק האם כל אחת ממילות השאילתא מופיעה בכותרת המסמך, נשמור ערך count ובמידה וכן נעלה את הcount בהתאם ונוסיפו לדירוג הכולל. בחרנו להשתמש באלגוריתם זה משום שהנחנו כי אם מילה מופיעה בכותרת המסמך, סביר להניח כי חשיבותה עולה על חשיבות המילים האחרות ולכן רצינו לתת לכך התייחסות.

<u>דוגמא להשפעה:</u> מסמך המכיל בכותרתו אחת ממילות השאילתא יקבל דירוג גבוה יותר מאשר מסמך שלא מכיל בכותרתו מילה מהשאילתא.

- **התייחסות לאישיויות הדומיננטיות של המסמך:**

עבור המסמך הרלוונטי נבדוק האם אחת ממילות השאילתא מוכלת בישות מתוך 5 הישויות החזקות במסמך, במידה וכן נעלה את הcount בהתאם ונוסיפו לדירוג הכולל..

<u>דוגמא להשפעה:</u> מסמך המכיל באחת מחמשת אישיותיו הדומיננטיות את אחת ממילות השאילתא יקבל דירוג גבוה יותר ממסמך אחר. למשל, d1 בעל אישיות דומיננטית "DR PAZ" ,

מסמך d2 לא מכיל אישיות זו.

לכן אם נחפש את השאילתא DR PAZ , מסמך d1 יקבל דירוג גבוה יותר.

את כל התוצאות מהנוסחאות השונות סכמנו באופן הבא:

```
docRank = bm25 + titalRank + dateRank + entityRank;
```

וזו למעשה התוצאה הסופית.

אלגוריתם למציאת 5 הישויות הדומיננטיות במסמך:

בתהליך הפירסור שלנו שמרנו עבור כל מסמך אישיות אופציונליות. לאחר תהליך הפירסור וידאנו כי כל ישות מופיעה בלפחות 2 מסמכים. לאחר מכן אנו עוברות על כל הישויות במסמך ומכניסות לtreeMap שממיין לפי התדירויות (כמות מופעים המסמך) מהגבוה לנמוך. גודל הרשימה ישאר קבוע בגודל 5 כך שכל ישות ששייכת למסמך תכנס לרשימה רק אם גודל הרשימה קטן מ5 או שהתדירות שלה יותר גבוה מאחת מהאישיות הנמצאות ברשימה זו.
כלומר נניח וישנם שני term שתואמים לתנאי של ישות במסמך, הראשון חוזר על עצמו 5 פעמים, השני חוזר על עצמו 6 פעמים, נסיק כי הterm השני הוא דומיננטי יותר. ראינו לנכון כי זהו המדד המתאים ביותר לדומיננטיות של ישות במסמך ספציפי.
רשימת אישיות זו נשמר בקבצי הpost עבור כל מסמך ובקריאת קבצי הpost נכניס את הישיות ל treeMap שתחזקנו עבור כל מסמך. נשתמש ברשימה זו לדירוג הכולל של המסמך.

דוגמא ליישות: (מקובץ הpostingשל docs)

FBIS4-30825#Zantovsky Interviewed on U.S. Economic Ties#477#20#20 May 1994#UNITED STATES&8,

ניתן לראות כי במסמך FBIS4-30825 נמצאת האישות "UNITED STATES" 8 פעמים.
כאשר משתמש ירצה לראות מי הן הישויות, הוא יבחר מסמך, ואנו נחלץ את הישויות משדה topFiveEntites של אותו document.

אלגוריתם לשיפור סמנטי:

במנוע שלנו ישנה אפשרות לחיצה להרצת סמנטיקה בשתי דרכים:

Online:
המימוש של האלגוריתם מתבצע במחלקת ה-Searcher ע"י שימוש בapi - (https://api.datamuse.com/words?rel_syn) במידה והמשתמש בוחר להריץ שאילתא עם שיפור סמנטי, האלגוריתם עובר על ערך ה-String של השאילתא ועבור כל מילה בשאילתא ניגש ל api הנ"ל ובודק האם קיימות מילים נרדפות לאותה מילה. במידה וקיימת מילה נרדפת, האלגוריתם מוסיף אותה לשאילתא.

Offline:
שימוש בקובץ jar - Word2VecModel ובקובץ טקסט הנמצא בresources שלנו. באותו אופן נוסיף לרשימת המילים שבשאילתא שלנו את המילים הנרדפות השמורות בקובץ הטקסט. בחרנו להגביל את כמות ה term שנוסיף לרשימה עבור כל term בשאילתה, זאת ע"מ לא להעלות משמעותית את זמני הריצה.

4. נתונים בקובץ הpostingה והמילון שתומכים באלגוריתמים:

1) בקבצי הpostingה נשמרים:

א) מספר החזרות של כל term במסמך- מסייע בחישוב ה-tf

ג) המסמכים בהם מופיע כל term (docNo) - מסייע בחישוב ה-idf.

2) במילון נשמרים:

א) נתיב קובץ הפוסט ופוינטר כדי לייבא את הנתונים של המילה

ב) כמות החזרות של המילה בכל הקורפוס

ג) מספר המסמכים שרלוונטיים לכל term.

שאר הנתונים נשמרים באובייקט הDocs, שכולל פרטים על כל document במאגר, כמו כותרת המסמך, תאריך פרסום המסמך, אורך המסמך והמספר הכולל של המסמכים במאגר. כמו כן, עבור כל מסמך נשמור את 5 האישיות הדומיננטיות וה-rank של כל אחת.

5. שימוש בקוד פתוח:

בחלק זה של הפרויקט השתמשנו ב- API לצורך הסמנטיקה:

https://api.datamuse.com/words

עשינו שימוש בקוד זה במחלקת ה-Searcher כאשר המשתמש בחר באופציה של שיפור סמנטי

(פירוט לגביי אופן פעולת האלגוריתם לשיפור סמנטי בסעיף ב' בדו"ח).

הרצות trece eval:

without stemmer & without semantic

Query num	Queries's word	Precision	Recall	Precision@5	Precision@15	Precision@30	Precision@50	running time for query (sec)
351	**Falkland petroleum exploration** What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?	0.111	0.354166667	0.1	0.2667	0.333	0.333	1.05
352	**British Chunnel impact** What impact has the Chunnel had on the British the life style of the British? economy and/or	0.009	0.024390244	0.4	0.2	0.1333	0.12	0.579
358	**blood-alcohol fatalities** What role does blood-alcohol level play in automobile accident fatalities**?**	0.199	0.431372549	0.2	0.5333	0.5	0.44	0.991
359	**mutual fund predictors** Are there reliable and consistent predictors of mutual fund performance?	0.0196	0.142857143	0	0.1333	0.1	0.08	1.66
362	**human smuggling** Identify incidents of human smuggling.	0.0507	0.205128205	0.4	0.2	0.1667	0.16	0.54
367	**piracy** What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?	0.0136	0.064864865	0.2	0.1333	0.1667	0.24	0.879
373	**encryption equipment expor** Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.	0.1157	0.3125	0.2	0.3333	0.1667	0.1	1.89
374	**Nobel prize winners** Identify and provide background information on Nobel prize winners.	0.0811	0.12254902	0.8	0.5333	0.5667	0.5	0.699
377	**cigar smoking** Identify documents that discuss the renewed popularity of cigar smoking.	0.0363	0.222222222	0.2	0.0667	0.1667	0.16	1.04
380	**obesity medical treatment** Identify documents that discuss medical treatment of obesity.	0.0649	0.428571429	0	0.133	0.1	0.06	0.9
384	**space station moon** Identify documents that discuss the building of a space station with the intent of colonizing the moon.	0.0691	0.254901961	0.2	0.2667	0.2333	0.26	0.872
385	**hybrid fuel cars** Identify documents that discuss the current status of hybrid automobile engines, (i.e., cars fueled by something other than gasoline only).	0.0687	0.235294118	0	0	0.2667	0.4	1.36
387	**radioactive waste** Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.	0.0286	0.150684932	0	0.1333	0.2333	0.22	1.055
388	**organic soil enhancement** Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.	0.0909	0.24	0.2	0.333	0.333	0.24	1.05
390	**orphan drugs** Find documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.	0.0408	0.081967213	0.4	0.4667	0.3	0.2	1.687
Map	0.0666							

stemmer & without semantic

Query num	Queries's word	Precision	Recall	Precision@5	Precision@15	Precision@30	Precision@50	running time for query (sec)
351	**Falkland petroleum exploration** What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?	0.1652	0.416666667	0	0.4	0.4	0.40	1.03
352	**British Chunnel impact** What impact has the Chunnel had on the British the life style of the British? economy and/or	0.009	0.024390244	0.4	0.2	0.1333	0.12	0.665
358	**blood-alcohol fatalities** What role does blood-alcohol level play in automobile accident fatalities**?**	0.1658	0.392156863	0.2	0.4667	0.4	0.4	1.47
359	**mutual fund predictors** Are there reliable and consistent predictors of mutual fund performance?	0.0306	0.178571429	0	0.2	0.1333	0.1	1.76
362	**human smuggling** Identify incidents of human smuggling.	0.0671	0.205128205	0.2	0.2	0.1667	0.16	0.664
367	**piracy** What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?	0.015	0.07027027	0.2	0.0667	0.1667	0.26	0.966
373	**encryption equipment expor** Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.	0.2425	0.4375	0.4	0.4667	0.2333	0.14	1.55
374	**Nobel prize winners** Identify and provide background information on Nobel prize winners.	0.0522	0.093137255	0.8	0.4667	0.3333	0.38	0.87
377	**cigar smoking** Identify documents that discuss the renewed popularity of cigar smoking.	0.1213	0.444444444	0	0.1333	0.3	0.32	1.56
380	**obesity medical treatment** Identify documents that discuss medical treatment of obesity.	0.0592	0.428571429	0	0.1333	0.1	0.06	1.2
384	**space station moon** Identify documents that discuss the building of a space station with the intent of colonizing the moon.	0.0813	0.22	0.6	0.2667	0.2	0.22	0.872
385	**hybrid fuel cars** Identify documents that discuss the current status of hybrid automobile engines, (i.e., cars fueled by something other than gasoline only).	0.0665	0.223529412	0	0.0667	0.3333	0.38	1.78
387	**radioactive waste** Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.	0.0337	0.150684932	0.2	0.1333	0.2	0.22	1.044
388	**organic soil enhancement** Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.	0.1632	0.32	0.6	0.4667	0.4667	0.32	1.22
390	**orphan drugs** Find documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.	0.0608	0.12295082	0.4	0.4	0.4333	0.3	1.899
Map					0.0889			

without stemmer & semantic

Query num	Queries's word	Precision	Recall	Precision@5	Precision@15	Precision@30	Precision@50	running time for query (sec)
351	**Falkland petroleum exploration** What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?	0.1623	0.4375	0	0.2	0.4333	0.42	1.33
352	**British Chunnel impact** What impact has the Chunnel had on the British the life style of the British? economy and/or	0.047	0.008130081	0.2	0.1333	0.0667	0.04	0.322
358	**blood-alcohol fatalities** What role does blood-alcohol level play in automobile accident fatalities**?**	0.1641	0.37254902	0.2	0.4667	0.4333	0.38	1.55
359	**mutual fund predictors** Are there reliable and consistent predictors of mutual fund performance?	0.0047	0.071428571	0	0	0.0667	0.04	0.66
362	**human smuggling** Identify incidents of human smuggling.	0.0013	0.025641026	0	0	0.0333	0.02	0.778
367	**piracy** What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?	0.0059	0.037837838	0	0.1333	0.1333	0.14	0.987
373	**encryption equipment expor** Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.	0.0729	0.125	0.4	0.1333	0.0667	0.04	1.06
374	**Nobel prize winners** Identify and provide background information on Nobel prize winners.	0.0537	0.112745098	0.2	0.4	0.4333	0.46	0.675
377	**cigar smoking** Identify documents that discuss the renewed popularity of cigar smoking.	0.0217	0.111111111	0.2	0.1333	0.1333	0.08	1.54
380	**obesity medical treatment** Identify documents that discuss medical treatment of obesity.	0	0	0	0	0	0	1.2
384	**space station moon** Identify documents that discuss the building of a space station with the intent of colonizing the moon.	0.0241	0.14	0	0.2	0.1333	0.14	0.788
385	**hybrid fuel cars** Identify documents that discuss the current status of hybrid automobile engines, (i.e., cars fueled by something other than gasoline only).	0.0355	0.129411765	0.2	0.2667	0.2	0.22	1.56
387	**radioactive waste** Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.	0.0206	0.136986301	0	0	0.1333	0.2	1.07
388	**organic soil enhancement** Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.	0.0529	0.16	0.2	0.333	0.2333	0.16	0.998
390	**orphan drugs** Find documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.	0.0219	0.06557377	0.4	0.2667	0.2333	0.16	1.522
Map	0.0431							

סיכום:

בעיות שנתקלנו בהן- הבעיה העיקרית הייתה עמידה בזמני ריצה וסיבוכיות מקום.

בעיה נוספת הייתה אחזור נכון של המסמכים- בשלב ההתחלתי של הרצת קובץ השאילתות שקיבלנו, חזרו לנו תוצאות לא טובות. זה גרם לנו לבדוק הרבה מהפונקציות שבדקנו מספר פעמים ואף לבדוק האם ישנן בעיות מחלק א של העבודה.

כיצד התמודדנו איתן- היינו מאוד יסודיות בחלק א ולכן זה גרם לאחזור טוב יחסית של מסמכים בחלק זה של העבודה, ומכיוון שייחסנו חשיבות למספר גורמים שונים בנוסחת הדירוג התוצאות השתפרו. שינינו את הפרמטרים b, k של האלגוריתם העיקרי BM25 ע"י ניסוי וטעייה, בנוסף לנרמולים של הפונקציות שבחרנו לתת להן חלק בנוסחת הדירוג הסופית שלנו כך שחל שיפור משמעותי באחזור המסמכים.

בכדי לעמוד בזמני ריצה סבירים, עשינו שימוש במבני נתונים שמאפשרים הכנסה ושליפה מהירה ברוב המקרים, כמו HashMap , TreeMap וכד'.

בנוסף, כאשר הזיכרון התמלא, ביצענו כתיבות לדיסק שכידוע מדובר בפעולה שאורכת זמן רב ולכן ניסינו ככל שביכולתנו לשמור על איזון בין הגישות והכתיבות לדיסק לבין השימוש בזיכרון בצורה יעילה בעיקר בחלק א' של הפרויקט.

האתגר הגדול לדעתנו- לא פשוט לדבג כמות כה גדולה של מסמכים וterms, ועל כן יש לכתוב את הקוד בריכוז מלא תוך התחשבות במקרי קיצון רבים. חשוב להיות יסודיים ביותר בחלק א של העבודה כי זהו הבסיס המרכזי של חלק ב. ללא ספק למדנו לדבג ביסודיות ולתת חשיבות רבה למחשבה על מקרי קיצון, רק כך הצלחנו לאחזר מסמכים בצורה טובה.

המלצות לשיפור האלגוריתם- אם היה לנו עוד זמן היינו מנסות לשנות עוד את הערכים שנתונים לבחירתנו ואולי כך היינו מצליחות לשפר את תוצאות האחזור. מבחינת נוסחת הדירוג, השתדלנו לדייק עד כמה שיכולנו תוך התחשבות בפרמטרים שלנו בנוסחה (אלו שהיו נתונים לבחירתנו ואלו שהיינו מחויבות לעשות). ראינו לנכון לתת חשיבות לכל אחד מהנתונים שבחרנו להוסיף-כותרות,description ותאריכי פרסום כיוון שהם חשובים מבחינתנו באחזור מידע. אנו מקוות שהתוצאות אכן יעידו על כך.