דו"ח הפרויקט – חלק ב'

a. אופן פעולת המנוע:

באמצעות ממשק גוי, תחילה המשתמש מזין נתיב לטעינת קבצי ה-posting שנשמרו בדיסק בחלק א' ולוחץ "Load Dictionary". המערכת קוראת את קבצי הפוסטינג ומשחזרת את כלל הנתונים – אודות המילים השונות, ומידע נוסף לגבי כל מסמך – אורכו, הישויות הדומיננטיות שהופיעו בו, ועוד.

המשתמש מכניס את הקלט הבא: שאילתא או נתיב לקובץ שאילתות, ולוחץ על כפתור RUN.

המערכת שולחת את השאילתא או הקובץ אל מחלקת Searcher, אשר מזהה באיזה **מ**קלטי השאילתות מדובר, ומעבירה להמשך טיפול מתאים במחלקת Ranker.

בנוסף, מחלקת Searcher שולחת את מילות השאילתות אל מחלקת Parse לצורך התאמתן לפרסור שהתבצע עבור הקורפוס עצמו. במחלקת Ranker מבוצע תהליך מציאת המסמכים הרלוונטיים לשאילתא, והחזרתם בצורה ממויינת על פי נוסחת דירוג שפיתחנו. כל התוצאות חוזרות למחלקת Controller ומוצגות למשתמש באמצעות ממשק הגוי.

קיימת למשתמש האפשרות לייצא את התוצאות לקובץ results.txt בפורמט TREC_EVAL, וכן להציג את 5 הישויות הדומיננטיות עבור כל אחד מהמסמכים שאוחזרו בשאילתא.

b. פירוט המחלקות השונות בתוכנית:

Parse:

מחלקה זו מפרקת את קטע הטקסט מכל מסמך ל-terms נפרדים, על סמך מגוון חוקים של השפה בנוגע למספרים, שמות, אותיות גדולות וקטנות, וכו'. שיטות שהוספנו למחלקה:

- ***private void handleEntity(Document newDoc)*** – השיטה עוברת על המסמך כולו ונעזרת בפונקציות הרלוונטיות להמשך טיפול בביטויים החשודים כישויות. השיטה מוסיפה את הישויות שמצאה לשדה המחלקה entitys (HashMap).
- ***private String firstTermEntity(String term)*** – השיטה מקבלת מחרוזת החשודה בתור המילה הראשונה של הישות, מוודאה שהאות הראשונה היא אות גדולה, ושאר המילה כתובה באותיות קטנות. אם המחרוזת אינה עומדה בהגדרת 'ישות' – תחזיר NULL. באותה הצורה בדיוק פועלת השיטה ***secondTermEntity(String term)***.

Indexer:

מחלקה זו מקבלת את כלל המילים מה-parse ויוצרת את ה- inverted index התואמים: מילון שנשמר בזכרון הראשי, והמשך טיפול עבור יצירת קבצי posting שנשמרים בדיסק. שיטות שהוספנו למחלקה:

- ***private void deleteEntity()*** – נקרא לשיטה בסיום פעולת האינדקסר, נרצה לוודא שמילון הישויות מעודכן. השיטה עוברת על מבנה הנתונים – במידה ורואה כי ישות מופיעה במסמך אחד בלבד (כלומר לא נחשבת ישות על פי הנחיות העבודה) – מסירה אותה ממבנה הנתונים.

- ***private void updateEntity (Document doc)*** – השיטה מעדכנת את מבנה הנתונים הכולל של הישויות בתוכנית מתוך המסמך הנוכחי doc שקיבלה – מוסיפה ישויות חדשות, או מעדכנת ערכים עבור ישויות שכבר קיימות.
- ***Private void updateDocs()*** - נקרא לשיטה בסיום פעולת האינדקסר. שיטה זו תוסיף למבנה הנתונים של כלל המסמכים בקורפוס את המידע אודות הישויות עבור כל מסמך. לבסוף תקרא למחלקה Writer שתכתוב את המידע עצמו לקובץ פוסטינג נוסף.

Writer:

מחלקה זו כותבת את כל המידע לדיסק, ל-8 קבצי posting שונים לכל היותר, בחלוקה לטווחי אותיות בשפה האנגלית, קובץ נוסף עבור ביטויים מספריים, וקובץ נוסף המכיל מידע על המסמכים במאגר. שיטות שהוספנו למחלקה:

- ***public void writeDocuments(HashMap<String,String> docs)*** – השיטה מקבלת את מבנה הנתונים על כל המסמכים במאגר וכותבת את המידע לקובץ docs.txt בנתיב הנבחר.

Controller:

מחלקה זו מציגה את הפרויקט למשתמש בתור ממשק נוח וידידותי, ולכן מתממשקת עם כל שאר המחלקות שתוארו לעיל. שיטות שהוספנו למחלקה:

- ***public void onRun ()*** – השיטה מפעילה את אחזור המסמכים הרלוונטיים עבור השאילתא. השיטה יוצרת אובייקט מסוג Searcher לפי מחרוזת הקלט שקיבלה או לפי קובץ השאילתות שהמשתמש טען אל המערכת. השיטה מעבירה את המשך הטיפול למחלקה Searcher.
- ***private void onSaveResults (String info)*** – השיטה מקבלת נתיב לתיקייה בה המשתמש רוצה לשמור את המידע שהתקבל – רשימת המסמכים שחזרו עבור אותה שאילתא/שאילתות שהריץ.

Searcher:

מחלקה זו מקבלת את השאילתא מהמשתמש (מילה או אוסף מילים עם רווחים בינהם), או קובץ שאילתות, מנתחת אותן ומחזירה את המסמכים הרלוונטיים ביותר בצורה מדורגת (בעזרת מחלקת Ranker). שיטות המחלקה:

- ***Public Searcher (String query, String postingPathSaved, String queryFilePath, boolean semantics, boolean stemm)*** – בנאי המחלקה. מעדכן את נתיבי התיקייה להיות אלו שהתקבלו, את המשתנה הבוליאני אודות הפעלת stemming על המאגר, את המשתנה הבוליאני אודות הפעלת סמנטיקה על המאגר.
- ***Public void proccessQuery()*** – השיטה מעבדת את השאילתא/שאילתות שקיבלה ומוסיפה את מילות השאילתא השונות למבנה נתונים (HashMap<String, String[]>queriesToSearchAndRank), ושולחת אותו אל המחלקה Ranker.

- ***Private String[] prepareToRank(Document doc)*** – השיטה מקבלת את השאילתא בתצורה של מסמך, ומחזירה מערך של מחרוזות, אשר כל מחרוזת הינה מילה בשאילתא, לטובת נוחות קליטת השאילתא במחלקה Ranker בהמשך.
- ***private void readQueryFile()*** – השיטה עוברת על מסמך הטקסט של שאילתות שקיבלה מהמשתמש – ועבור כל אחת יוצרת אובייקט תואם מסוג Document ומוסיפה אותו לתור המסמכים querySet הקיים כשדה במחלקה, לצורך העברתו למחלקת Parse. את רשימת המסמכים המפורסרים שחזרו – מעדכנת בשדה querySet.
- ***Public String addSemantics(String queryTerms) –*** השיטה מקבלת את כל מילות השאילתא כמחרוזת אחת, ומחזירה עבור כל מילה – שלוש מילים נרדפות נוספות, גם ללא חיבור לאינטרנט, לצורך הרחבת רשימת המסמכים שיאוחזרו בהמשך עבורה.
- ***Public String addSemanticsWordsOffline(String term) –*** השיטה מקבלת מילה בודדת ובאמצעות חיפוש בקובץ ג'אר חיצוני שהורדנו – מחזירה מספר מילים נרדפות בהתאם לבחירתנו (3). כל זה מתאפשר ללא חיבור לאינטרנט כלל.

.:Ranker

מחלקה זו מדרגת את רשימת המסמכים שנמצאו מתאימים עבור השאילתא שקיבלה, או קובץ השאילתות. הדירוג מתבצע על פי פונקציה שפיתחנו, המתבססת על נוסחת BM25, ולה הוספנו רכיבים נוספים שמצאנו לנכון. שיטות המחלקה:

- ***public Ranker (String path) –*** בנאי המחלקה. מעדכן את נתיב התיקייה בה נמצאים קבצי הפוסטינג הנדרשים לקריאת וטעינת המידע אודות המסמכים השונים בקורפוס (אורך, ישויות..).
- ***public HashMap<String, List<String>>rankAllQueries (HashMap<String, String[]allQueries)*** – השיטה המרכזית במחלקה – מקבלת את כלל השאילתות מהמשתמש ומחזירה HashMap, אשר המפתח שלו הוא שם השאילתא (String), והערך הוא רשימת שמות המסמכים הרלוונטים שאוחזרו, על פי דירוג.
- ***private List<String> rank(String[] terms) –*** השיטה מקבלת את מילות השאילתא בצורת מערך, ומחזירה רשימה של שמות המסמכים המאוחזרים הרלוונטיים, בצורה מדורגת. תחילת השיטה מוצאת את הקשר בין כל מילה לבין המסמכים בהם הופיעה, ובהמשך מאגדת את כל המידע אודות המסמך (ביחד עם מילות השאילתא הנוספות), וכך מעניקה לו דירוג מתאים.
- ***private String getFilePath(String term) –*** השיטה מקבלת מילה ועל פי התו הראשון שלו – מייעדת אותו אל מסמך הפוסטינג הרלוונטי, על פי החלוקה לטווחי אותיות שביצענו בחלק א'.
- ***private HashMap<String, int[] getDataForTerm(String term) –*** השיטה מקבלת מילה מהשאילתא ומחזירה HashMap בו המפתח הוא שם המסמך הרלוונטי, והערך יהיה מערך מספרים השומרים מידע אודות המסמך (בהקשר למילה הספציפית), בצורה הבאה: tf, idf, אורך המסמך, ייצוג בינארי (0 או 1) האם המילה נמצאת בתחילת המסמך או לא (כך תקבל דירוג גבוה יותר), ייצוג בינארי (0 או 1) האם המילה נמצאת בכותרת או לא.
- ***private int getNumberOfDocs() –*** השיטה מחזירה את כמות המסמכים הנמצאים ב-HashMap של כלל המסמכים הנמצאים בשדה של האובייקט אינדקסר.

- ***private int getLengthOfDoc(String docID)*** – השיטה מחזירה את אורך המסמך שקיבלה את שמו במחרוזת בקלט.
- ***private double rankTermDoc(int[] data)*** – השיטה מקבלת את מערך המספרים המייצגים מידע עבור מסמך יחיד ביחס לשאילתא המסוימת, ומחזירה דירוג של המסמך בהתאם לנוסחה שפיתחנו, שמתבססת על מדד BM25 בתוספת נתונים ומשתנים נוספים שבחרנו.
- ***private double rankQueryDoc(List<int[]> dataQuery)*** – השיטה מחזירה ציון עבור מסמך (המיוצג על ידי מערך של מספרים) בהתאם לשאילתא, על ידי כך שמתכללת את כלל הציונים שקיבלו ה-terms השונים במסמך עבורה. שיטה זו נעזרת בשיטה הממוקדת יותר – rankTermDoc.

c. <u>הסבר אלגוריתמים במנוע:</u>

I. <u>אלגוריתם הדירוג:</u>

אלגוריתם הדירוג בו השתמשנו מבוסס על הנוסחא של BM25 כאשר הוספנו פרמטרים נוספים לטובת שיפור דירוג המסמכים שהתקבלו:

$$\mathrm{score}(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\mathrm{avgdl}}\right)} \qquad \mathrm{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

כאשר:

N – מספר המסמכים הכולל בקורפוס.

$n(q_i)$ – מספר המסמכים שהמילה הופיעה בהם.

$f(q_i, D)$ - כמות הופעות המילה במסמך.

D - אורך המסמך (כמות המילים).

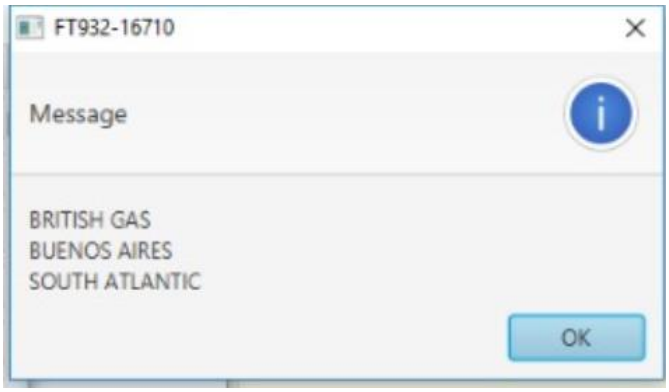
$avgdl$ - אורך ממוצע של המסמכים בקורפוס.

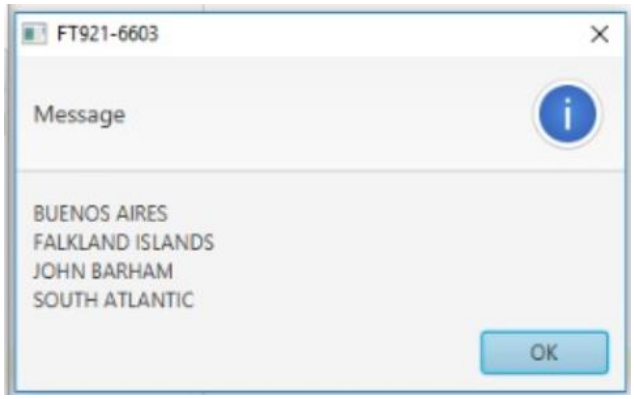
k, b - קבועים.

אלגוריתם הדירוג הניב תוצאות שונות עבור כל שינוי מזערי שעשינו בפרמטרים, לכן שינינו פעמים רבות את משקלי הערכים בכדי להגיע לתוצאה אופטימלית.

II. <u>אלגוריתם למציאת ישויות:</u>

כתבנו אלגוריתם שמחפש את הנתונים השמורים אודות מסמך מסוים בקובץ ההופכי, ומוציא ממנו את רשימת הישויות שהופיעו במסמך. האלגוריתם מחזיר לכל היותר את 5 הישויות שקיבלו את הציון הגבוה ביותר. הציון שהענקנו מורכב מכמות ההופעות של הישות בקורפוס כולו. האלגוריתם מחזיר את רשימת הישויות בצורה ממוינת. <u>דוגמאות להצגת הישויות:</u>





III. אלגוריתם לשיפור סמנטי:

אלגוריתם זה מסייע לנו לחפש במסמכים מילים נוספות הקשורות למילים שנכתבו בשאילתא עצמה. מילים אלו לרוב תהיינה בעלות קשר חזק למילים המקוריות, כלומר דומות במשמעותן, ולכן האחזור יהיה רלוונטי לשאילתא. השתמשנו באלגוריתם לטיפול סמנטי, אשר מחזיר עבור כל מילה שמקבל – מספר מילים נרדפות לבחירתנו (3). באופן זה, הרחבנו את טווח החיפוש עבור כל אחת מן השאילתות, תוך נסיון לאחזר כמות גדולה יותר של מסמכים רלוונטיים. השתמשנו בקוד פתוח של אלגוריתם לחיפוש סמנטי ללא חיבור לאינטרנט.

d. הסבר נתונים מתוך קבצי posting:

בקובץ docs.txt שמרנו מידע אודות כל אחד מהמסמכים שהופיעו בקורפוס. תחילת שמרנו את שם המסמך, את אורכו (ספירת כמות המילים המופיעות בו), ולאחר מכן את רשימת כל הישויות שהוזכרו בו. באורך המסמך השתמשנו לטובת שיפור הציון שמקבל כל מסמך במחלקת Ranker בעת דירוג המסמכים המאוחזרים על פי מידת רלוונטיות לשאילתא. ברשימת הישויות השתמשנו לצורך הצגתן למשתמש, באם מבקש זאת, עבור כל אחד מהמסמכים שאוחזרו עבור השאילתא.

בשאר שבעת קבצי ה- posting (מחולקים לטווחי אותיות) שיצרנו השתמשנו לטובת טעינת המילון עצמו בעת הפעלת חלק ב' של התכנית. כל שורה בקובץ מתארת לנו את המילה עצמה, רשימת המסמכים בהם הופיעה, ונתונים מספריים נוספים ששימשו אותנו בהמשך.

במילון שיצרנו אנו מחפשים כל מילה מהשאילתא/תיאורה/מילה בעלת הקשר סמנטי. משנמצאה, מתקבל לנו המידע הבא:

- מספר המסמכים בהם היא מופיעה
- מספר הופעות כולל בקורפוס כולו

ניתוח והערכת המנוע:

עבור הרצת קובץ השאילתות בלי stemming:

QueryID	Query words	Average precision	Precision at 5 docs	Precision at 15 docs	Precision at 30 docs	Precision at 50 docs	Recall	Running Time (seconds)
351	Falkland petroleum exploration What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?	0.1009	0	0.1333	0.3333	0.32	0.33333333	14.4052
352	British Chunnel impact What impact has the Chunnel had on the British economy and/or the life style of the British?	0.0053	0.2	0.1333	0.1667	0.12	0.02439024	8.98401
358	blood-alcohol fatalities What role does blood-alcohol level play in automobile accident fatalities?	0.1838	0.2	0.5333	0.5	0.4	0.39215686	10.5478
359	mutual fund predictors Are there reliable and consistent predictors of mutual fund performance?	0.0188	0	0.0667	0.0667	0.1	0.17857143	8.37202
362	human smuggling Identify incidents of human smuggling.	0.0212	0	0.1333	0.1333	0.12	0.15384615	5.64681
367	piracy What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?	0.0099	0	0.1333	0.1667	0.2	0.05405405	10.137
373	encryption equipment export Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.	0	0	0	0.1	0.06	0.1875	15.4546
374	Nobel prize winners Identify and provide background information on Nobel prize winners.	0.0747	0.8	0.5333	0.5333	0.46	0.1127451	8.66904

377	cigar smoking Identify documents that discuss the renewed popularity of cigar smoking.	0.0323	0	0	0.1667	0.16	0.22222222	10.9896
380	obesity medical treatment Identify documents that discuss medical treatment of obesity.	0.1086	0	0.2	0.1	0.08	0.57142857	10.1817
384	space station moon Identify documents that discuss the building of a space station with the intent of colonizing the moon.	0.0326	0	0.1333	0.1667	0.18	0.17647059	16.5765
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).	0.0188	0	0	0.1667	0.2	0.11764706	22.6602
387	radioactive waste Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.	0.0172	0.2	0.0667	0.2	0.16	0.10958904	13.6453
388	organic soil enhancement Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.	0.0979	0.6	0.4667	0.3333	0.22	0.22	18.0004
390	orphan drugs Find documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.	0.0459	0.2	0.4667	0.3667	0.44	0.18032787	17.5091

מדד ה-MAP: 0.0521

עבור הרצת קובץ השאילתות עם stemming:

QueryID	Query words	Average precision	Precision at 5	Precision at 15	Precision at 30	Precision at 50	Recall	Running Time (seco
351	Falkland petroleum exploration What information is available on petroleum exploration in the South Atlantic near the	0.0501	0	0.0667	0.2333	0.24	0.25	16.746
352	British Chunnel impact What impact has the Chunnel had on the British economy and/or the life style of the	0.0073	0.2	0.0667	0.1667	0.18	0.03659	11.395
358	blood-alcohol fatalities What role does blood-alcohol level play in automobile accident fatalities?	0.144	0.2	0.4667	0.3333	0.36	0.35294	8.71
359	mutual fund predictors Are there reliable and consistent predictors of mutual fund performance?	0.0077	0	0.0667	0.0667	0.04	0.07143	6.9312
362	human smuggling Identify incidents of human smuggling.	0	0	0	0	0	0	4.588
367	piracy What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?	0	0	0	0	0	0	8.5116
373	encryption equipment export Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.	0.0104	0	0.0667	0.0333	0.02	0.0625	13.466
374	Nobel prize winners Identify and provide background information on Nobel	0.1028	1	0.6667	0.6	0.6	0.14706	9.9587

377	cigar smoking Identify documents that discuss the renewed popularity of cigar smoking.	0.076	0	0.0667	0.1	0.28	0.38889	9.3297
380	obesity medical treatment Identify documents that discuss medical treatment of obesity.	0	0	0	0	0	0	8.9299
384	space station moon Identify documents that discuss the building of a space station with the intent of colonizing the moon.	0.0292	0.4	0.1333	0.1	0.14	0.13725	14.56
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).	0.0197	0	0.0667	0.2	0.18	0.10588	18.463
387	radioactive waste Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.	0.0059	0	0.0667	0.0667	0.1	0.1233	12.076
388	organic soil enhancement Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.	0.0103	0.2	0.0667	0.1	0.08	0.06849	14.493
390	orphan drugs Find documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.	0.0017	0	0	0.0333	0.04	0.04	16.871

מדד ה-MAP: 0.0286

לסיכום

נתקלנו בכמה בעיות עיקריות במהלך עבודה על הפרויקט:

- עבודה עם ת'רדים

- זמני ריצה ארוכים

- דירוג יעיל

התמודדנו עם הבעיות לעיל על ידי התייעצות עם חברים למחלקה, וחיפוש נרחב באינטרנט וכמובן ניסויים רבים. האתגר הגדול בעבודה עם ת'רדים היה הסנכרון ביניהם, בו ניסינו להשתמש לצורך שיפור זמני הריצה של יצירת המילון מהקורפוס כולו. שימוש זה עזר לנו לשפר את זמני הכתיבה והקריאה מהדיסק, דבר שאפשר לנו לבצע את החיפוש כולו באופן מהיר יותר.