ת.ז : 206228462, 308235605

דו"ח מנוע חלק ב'

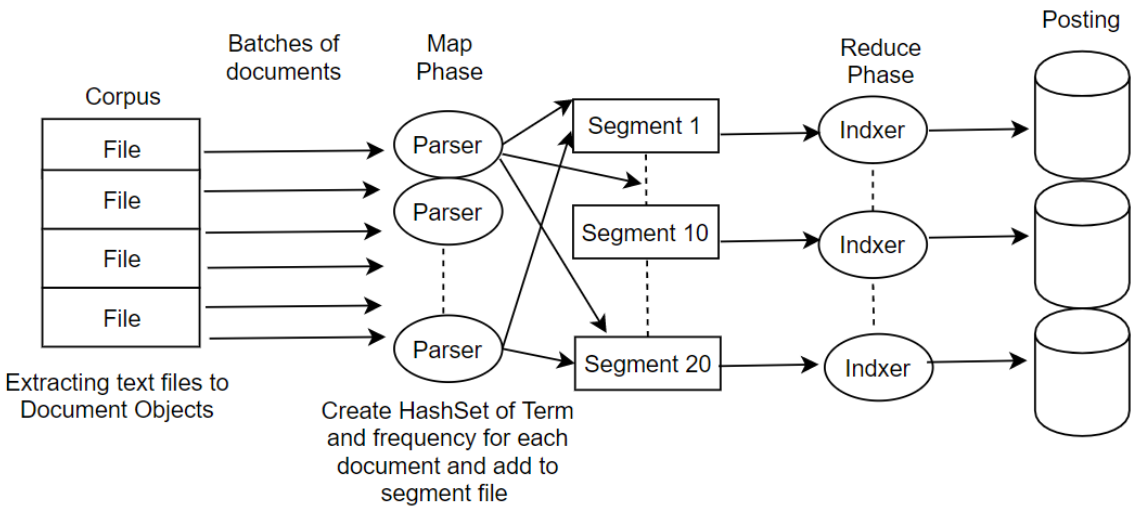
מבנה התוכנית:

חלק א:

מבנה התוכנית מבוסס על ארכיטקטרת ה**MapReduce** שנלמדה בביתה יחד עם תהליך הפרסור.
נפריד את התהליך למספר שלבים עיקריים:
קריאת המסמכים וחלוקתם לאובייקטים מסוג מסמך.
שלב הפרסור **(Parser)** וכתיבה לסמגנטים אשר מתבצע באופן מקבילי, ובו אנו מפרסרים כמות מוגדרת שנבחרה מראש של מסמכים, מבצעים את תהליך הפירוק למונחים, שמירתם למבני נתונים זמנים ולבסוף לקבצים זמניים (**Segments**).

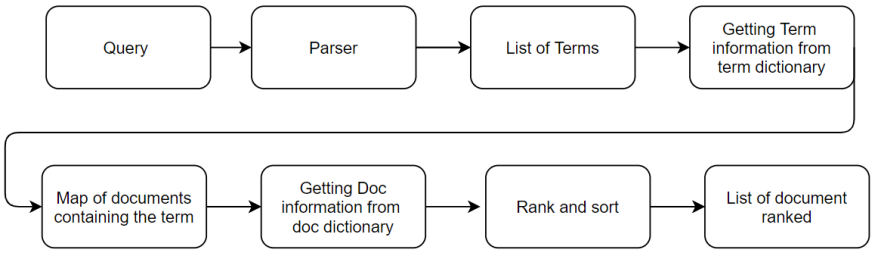
בשלב הבא **(Index)** אני מבצעים את תהליך ה**Reduce**, על ידי קריאה באופן מקבילי של קבצי ה**Segment**, ואיחודם אל קבצי **Posting**, אשר נכתבים על הדיסק ומכילים את הפרטים הנדרשים עבור כל מונח.

לבסוף אנו מבצעים שמירה של המילון מונחים, שיחד עם קבצי ה**Posting**, מרכיבים את מבנה האינדקס ההופכי.



חלק ב':

בעת הכנסת השאילתא, היא עוברת את שלב הפרסור, ונשמור את כל המונחים שנמצא במבנה נתונים.
לאחר מכן ניגש ל **Term Dictionary**, ועבור כל מונח, נשמור את הפרטים אודות מספר הופעותיו ובמבנה נתונים נוסף נשמור את כל המסמכים שבהם הוא הופיע, את הגישה למילונים אנו עושים באמצעות **Random Access**, שמאפשר גישה ישירה למיקום בקובץ ללא צורך בפרסור המילונים.
לאחר שאספנו מידע זה ניצור מבנה נתונים מסוג מפה עבור כל המסמכים שמצאנו (במפתח) ובערך נשמור את רשימה המכילה את המידע אודות כל מונח שמופיע במסמך ורלוונטי וגם לשאילתא.
כעת כאשר בידנו מבני הנתונים שמייצג מסמך וכל רשימת המונחים והפרטים שלהם, בידנו היכולת לבצע את הדירוג.
נשלח את מבנה הנתונים לפונקציית הדירוג ולבסוף נוצר רשימת מסמכים מדורגת.



פירוט מחלקות שנוספו:

Query:
מחלקה שמחזיקה את המידע עבור שאילתה מסוימת לפי מס זיהוי של השאילתה, כותרת השאילתה ותיאור

שיטות:

- getQueryID() – מחזירה את הזיהוי של השאילתה
- getQueryTitle() – מחזירה את הכותרת של השאילתה
- getQueryDescription() – מחזיר את התיאור של השאילתה

Ranker:
מחלקה זו אחראית על ביצוע חישוב הדירוג עבור כל מסמך.

שיטות:

- rank(List<TermInfoInCorpus> termList) - פונקציה עיקרית האחראית על ביצוע הדירוג ומחזירה רשימה של כל המסמכים ודירוגם מסודרים לפי הסדר.
- generateMapDic(List<TermInfoInCorpus> termList) – פונקציה זו יוצרת מפה עבור המסמכים אשר מכילים את המונחים שנמצאו בשאילתא, עבור כל מסמך תמצא רשימה של כל המסמכים והפרטים לחישוב הנוסחא.
- **updateDocHashMap** – פונקציית עזר לייצירת מפת המסמכים.
- **rankedDocMap(HashMap<String, Pair<DocInfo, List<DocRankingTermInfo>>> docMap)**- פונקציית מעטפת לשליחת כל מסמך מהמפה לדירוג.
- **rankDocument(Pair<DocInfo, List<DocRankingTermInfo>> docInfoPair)**- פונקציית הדירוג, שמחזיר את הציון עבור המסמך הספציפי שאליו נשלח הדירוג.
- **rankTopFifty(List<Pair<String, Double>> rankedList)** – מחזיר רשימה המכילה את 50 המסמכים הרלוונטים.

מחלקת **Searcher**:

מחלקה זו אחראית על כלל תהליך החיפוש, גישה למילונים המתאמים ל**Posting files**, לבסוף מחזירה רשימה של מסמכים רלוונטים עבור שאילתא מתקבלת.

שיטות:

- **search(Query query, boolean isStem, boolean isSemantic)** – פונקציית החיפוש הראשית, אחראית על הפעלת ה**ranker**.
- **getInfoOnQueryTerms(Query query, boolean isStem, boolean isSemantic)** – פונקציה זו אחראית על גישה לקבצי הפוסטינג ולמילונים המתאמים, בנוסף מקבל ערכים בוליניסה מציינים האם יש צורך לבצע **stem** והאם להשתמש במודל הסמנטי.
- **getDocInfoFromPosting(String termPosting)**- מחזיר רשימה של **TermInfoInDoc** שמכילה את המידע הנדרש על המונח לביצוע הדירוג.
- **generateQueriesListFromFile(String pathToQueryFile)** – פונקציה זו אחראית על קריאת קובץ השאילתות והחזרת רשימה של שאילתות עליהם נבצע חיפושים.

מחלקת Word2Vec:
מחלקה זו הינה המחלקה האחראית על המודל הסמנטי, המחלקה מקבלת בבנאי של את רשימת המונחים שנרצה לבצע עליהם מודל סמנטי, ובייצרת האובייקט יוצרת מבנה נתונים המכיל את כל האופציות שנמצאו עבור מילה ספציפית.

שיטות:
Word2Vec(List<String> terms)- בנאי המחלקה אשר אחראי על הגישה למודל הסמנטי והחזרת המילים הרלוונטיות שנמצאו.
String getQueryAfterSematicAdd- מחזירה את המילים הנוספות בתור **String**.

אלגוריתם לשיפור הסמנטי:
בחרנו להשתמש במודל אשר ממש את מול ה **Word2Vec**, אשר נלמד בביתה. מודל זה מבוסס על רשתות ניורונים שמשתמש ב**hidden layers** לביצוע חישובי הקשרים בין המילים.
המודל הינו מודל קוד פתוח מאומן שניתן למצוא באינטרנט.
עבור בחירת משתמש של שימוש במודל הסמנטי, הוא מחפש עבור כל מילה שקיימת בשאילתא את 3 המילים הנרדפות הדומות ביותר לפי המודל (אם קיימות).
תוצאות האחזור הראו שהשימוש במודל הסמנטי באחזור מורידה את איכות האיחזור לפי ה**TRECEVAL**.

בנוסף למחלקות שפורטו מלעיל, השתמשנו במחלקות הבאות:
המחלקות סיפקו לנו את המידע הצריך עבור נוסחאת הדירוג,
TermInfoInCorpus – מחלקה זו הכילה את המונח עצמו, מידע על המונח ורשימת המסמכים בהם הופיע.
TermInfoInDoc – מחלקה זו הכילה מידע המונח הספציפי במסמך.
DocRankingTermInfo- מחלקה זו הכילה מידע על המסמך עצמו והקשר בינהם.

לכל המחלקות הללו קיימים **getters** ו**setters** בלבד.

הסבר על אלגוריתם הדירוג -
התבססנו על נוסחת BM-25 אשר נלמדה בהרצאות.

להלן הפירוט של הנוסחה השלמה :

- נוסחת הבסיס – BM-25

$$BM25(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * (1 - b + b * (\frac{|D|}{avgdl})}$$

בחרנו b=0.6, k=1.2 מצאנו לנכון שערכים אלו הם הערכים היעילים ביותר, עבור תוצאות אופטמליות.

הסבר:
השאילתא תעבור תהליך Parse לאחר מכן נבדוק כל מילה במילון וניקח את המצביעים לקובץ הPosting הרלוונטי.
בקובץ הPosting , עבור כל מילה נאסוף את כל המסמכים שהיא מופיעה בהם ואותם נציב בנוסחה אחד אחד. לבסוף נבצע סכימה על הערכים המתקבלים עבור כל מסמך ונחזיר את הערכים.
יצירתיות בדירוג:
נעשו נסיונות לתעדף מונח מסוג ישויות, בכך שבמידה וזוהה מונח מסוג ישות (instance of entity) נעניק לו משקל גבוהה יותר משאר סוגי המונחים, אך זה לא סיפק תוצאות טובות יותר ולעיתים ההפך. זאת לאחר הצבות משקלים שונים. עבור המודל הסמנטי, ראינו ששימוש מוריד את איכות האחזור .
בנוסף ניסנו להעזר באורך הווקטור עבור כל מסמך, מידע שיצרנו בחלק א', אך גם זה לא עזר באיכות הדירוג.
מצאנו כי התוצאות האופטימליות התקבלו לאחר שימוש בנוסחאת BM 25 ללא תוספות נוספות לערך הדירוג.

הסבר על מצאית 5 הישויות הרלוונטיות ביותר:

בתהליך הפרסור השתמשנו במחלקה עבור כל סוג של ישות מה שהקל על מציאת הישויות עבורנו.
עבור כל אובייקט מסוג DocInfo יצרנו תור עדיפויות שהותחל בקומפרטור השווה , ועבור כל ישות אשר נמצא בתהליך הפרסור בצענו עדכון בתור העדיפויות. לטובת מימוש הפוצקציונלאיות באופן היעיל והמהיר ביותר, בחרנו להוסיף בDoc Dictionary, עבור כל מסמך מעבר לפרטים שסיפקנו בחלק א' הוספו גם 5 הישויות עבור כל מסמך.
כך שרשומה במילון המסמכים נראתה כך:

<DocID>#<NumOfTerms><maxTf><DocVectorLength>

אלגוריתם 5 הישויות הינו מבוסס תור עדיפויות יחד עם קומפרטור השוואתי שבודק את כמות ההופעות של כל ישות, הליך זה מתבצע עוד בתהליך הפרסור.
דוגמא לפועלת התור:
נניח כי בהליך הפרסור התגלו 5 ישויות בלבד עם הופעה אחת, לאחר המשך הריצה התגלת ישות עם 2 הופעות אשר נכנסה לתור וכעת היא נמצא במיקום הראשון בתור. (אני יוצאים מנקודת הנחה שמדובר בהסבר טריוואלי ולכן לא מוספים עוד דוגמא נוספת).
כמו שציינו מעלה השפעת הישויות לא תרמה לנו באיכות האחזור.

מבנה קבצי הפוסטינג:

```
1.894K-95|2: FBIS4-26238,1# LA091990-0008,1#
Etela-Pohjanma|1: FT941-14825,1#
TEXPET|1: FBIS3-10943,1#
ITALIANAMERICAN|2: FT922-15134,1# LA052390-0042,2#
auto-dentistry|1: LA111489-0031,1#
-285.04|1: LA021290-0072,1#
MIB-30|2: FT944-10710,2# FT944-13969,2#
Higashi-machi|1: FBIS4-44919,2#
(Mintopenergo)|2: FBIS3-24461,1# FBIS4-41225,1#
Mr David Beddall|4: FT932-12136,1# FT941-3618,1# FT941-4256,1# FT942-14842,1#
```

- בכחול – שמרנו את המילה עצמה
- באדום – שמרנו את כמות המסמכים בהם היא הופיעה
- בירוק – שמרנו את שם המסמך בה היא הופיעה
- בצהוב – שמרנו את כמות הפעמים בהם הופיעה בכל מסמך

```
FT923-4374:78#4#134#Hong Kong,2;Cheung Kong,2;The Ch
FT923-4373:97#6#165#Guy Laroche,1;Dickson Concepts,1
FT923-4372:187#5#373#New Zealand,2;Countrywide Bank,
FT923-4371:284#13#786#Salomon Brothers,2;Mr Robert F
FT923-4370:105#8#267#Mini Disc,4;Electric Industrial
FT923-4369:167#5#265#Thai Airways,5;Mr Chatrachai,3;
```

מבנה מילון המסמכים:

- בכחול – שם המסמך
- בשחור- מספר מונחים
- באדום – מספר הופעות מקסימלי של מונח
- בירוק – הישויות הנפוצות ביותר עם מספר ההופעות
- בצהוב – אורך ווקטור המסמך

```
(011) 222-1341:1|15_190,63418|
(011) 222-3651:1|19_243,17386|
```

מבנה מילון המונחים:

- בכחול – שם המונח
- באדום – מספר מסמכים בהם מופיע
- בירוק – קובץ הפוסטינג המתאים
- בצהוב – ערכים בייטים עבור Random Access

קוד פתוח:

נעשה שימוש במודל הWORD2VEC, ספרית קוד פתוח שניתן להוריד מהאינרנט בקובץ JAR וקובץ המודל המאומן.
השתתמשנו בקוד זה בעת ביצוע החיפוש של השאילתא.
אופן שימוש: עבור בחירת משתמש של שימוש במודל הסמנטי, הוא מחפש עבור כל מילה שקיימת בשאילתא את 3 המילים הנרדפות הדומות ביותר לפי המודל (אם קיימות).

<u>הערכת המנוע:</u>

<u>תוצאות:</u>

Without Stem – הוחזרו סהכ 155 מסמכים

Stem off							
Query No	Query	Precision@5	Precision@15	Precision@30	Precision@50	Recall	Time Sec
351	Falkland petroleum exploration	0.2	0.4	0.33	0.34	0.35	0.994
352	British Chunnel impact	0.2	0.0667	0.133	0.08	0.016	3.077
358	blood-alcohol fatalities	0.2	0.533	0.5	0.42	0.4117	2.575
359	mutual fund predictors	0	0.067	0.133	0.14	0.25	1.383
362	human smuggling	0	0.067	0.133	0.08	0.102	0.551
367	piracy	0	0.267	0.1667	0.22	0.059	2.281
373	encryption equipment export	0	0.133	0.1	0.06	0.1875	2.247
374	Nobel prize winners	0.6	0.2667	0.3667	0.4	0.1	1.891
377	cigar smoking	0	0.0667	0.133	0.14	0.194	0.859
380	obesity medical treatment	0.2	0.0667	0.0667	0.04	0.285	1.127
384	space station moon	0.2	0.2	0.1667	0.22	0.21	2.032
385	hybrid fuel cars	0	0.067	0.2	0.13	0.15	2.647
387	radioactive waste	0.2	0.2	0.2	0.22	0.15	1.962
388	organic soil enhancement	0.4	0.2667	0.33	0.24	0.24	9.56
390	orphan drugs	0.2	0.4667	0.333	0.24	0.1	4.879
	TOTAL PRECISION	0.206667			SUM RECALL	0.124	

Average Precision over all rel docs = 0.0510

R-Precision = 0.1652

With Stem- הוחזרו 170 מסמכים

Stem On							
Query No	Query	Precision@5	Precision@15	Precision@30	Precision@50	Recall	Time Sec
351	Falkland petroleum exploration	0	0.2	0.2667	0.16	0.333	1.288
352	British Chunnel impact	0.2	0.067	0.1333	0.1	0.0203	3.18
358	blood-alcohol fatalities	0.2	0.4667	0.4	0.38	0.37	3.56
359	mutual fund predictors	0	0.133	0.133	0.12	0.214	2.71
362	human smuggling	0	0.067	0.133	0.1	0.12	0.817
367	piracy	0	0.2	0.15	0.2	0.054	4.115
373	encryption equipment export	0.2	0.2667	0.133	0.08	0.25	4.406
374	Nobel prize winners	0.6	0.4	0.333	0.26	0.063	2.856
377	cigar smoking	0	0.133	0.2	0.26	0.361	2.256
380	obesity medical treatment	0	0.133	0.1	0.06	0.42	2.08
384	space station moon	0.2	0.2	0.1667	0.2	0.196	4.026
385	hybrid fuel cars	0	0	0.667	0.17	0.2	4.4
387	radioactive waste	0.2	0.2667	0.2	0.28	0.191	4.298
388	organic soil enhancement	0.4	0.4	0.3667	0.4	0.4	3.051
390	orphan drugs	0.2	0.2	0.3667	0.3	0.122	8.219
	TOTAL PRECISION	0.226			SUM RECALL	0.136	

Average Precision over all rel docs = 0.0631

R-Precision = 0.1881

תצואות כולל שימוש במודל הסמנטי- הוחזרו 139 מסמכים

Semantic & Stem off							
Query No	Query	Precision@5	Precision@15	Precision@30	Precision@50	Recall	Time Sec
351	Falkland petroleum exploration	0.2	0.4	0.33	0.34	0.354	1.036
352	British Chunnel impact	0.2	0.067	0.133	0.08	0.016	3.18
358	blood-alcohol fatalities	0.2	0.533	0.5	0.42	0.411	2.74
359	mutual fund predictors	0	0.0667	0.1	0.1	0.178	2.53
362	human smuggling	0	0.0667	0.0667	0.06	0.076	0.894
367	piracy	0	0.2667	0.133	0.22	0.059	2.24
373	encryption equipment export	0.2	0.133	0.0667	0.04	0.125	3.58
374	Nobel prize winners	0.6	0.2667	0.3667	0.4	0.098	2.036
377	cigar smoking	0	0.0667	0.133	0.14	0.1944	0.976
380	obesity medical treatment	0	0	0	0	0	2.375
384	space station moon	0.2	0.2	0.133	0.14	0.13	2.56
385	hybrid fuel cars	0	0.2667	0.2	0.22	0.129	3.278
387	radioactive waste	0.2	0.1333	0.233	0.22	0.15	2.199
388	organic soil enhancement	0.4	0.2667	0.3	0.22	0.22	1.161
390	orphan drugs	0.2	0.4	0.2667	0.18	0.73	5.092
	TOTAL PRECISION	0.185			SUM RECALL	0.112	

Average Precision over all rel docs = 0.0440

R-Precision = 0.1401

תצואות כולל שימוש במודל הסמנטי- הוחזרו 147 מסמכים

Semantic & Stem on							
Query No	Query	Precision@5	Precision@15	Precision@30	Precision@50	Recall	Time Sec
351	Falkland petroleum exploration	0	0.2	0.2667	0.32	0.337	1.45
352	British Chunnel impact	0.2	0.0667	0.133	0.1	0.02	3.361
358	blood-alcohol fatalities	0.2	0.4667	0.4	0.38	0.372	3.8
359	mutual fund predictors	0.2	0.133	0.0667	0.06	0.107	4.2
362	human smuggling	0.2	0.133	0.1	0.08	0.102	2.72
367	piracy	0	0.2	0.1667	0.2	0.054	4.28
373	encryption equipment export	0.2	0.2	0.1	0.06	0.1875	6.5
374	Nobel prize winners	0.6	0.4	0.33	0.26	0.063	2.95
377	cigar smoking	0	0.133	0.2	0.22	0.305	2.49
380	obesity medical treatment	0	0	0	0	0	4.7
384	space station moon	0	0.133	0.133	0.16	0.156	4.5
385	hybrid fuel cars	0	0.333	0.3	0.32	0.188	6.2
387	radioactive waste	0.2	0.133	0.133	0.2	0.13	4.9
388	organic soil enhancement	0.6	0.6	0.533	0.38	0.38	3.42
390	orphan drugs	0.2	0.333	0.3	0.2	0.081	8.35
	TOTAL PRECISION	0.196			SUM RECALL	0.118	

Average Precision over all rel docs = 0.0519

R-Precision = 0.0.1550

כפי שניתן לראות התוצאות המקסימליות התקבלו בעת חיפוש עם שימוש ב **Stemming** , וניתן לראות ששימוש במודל הסמנטי גרם לירידה באיכות האחזור.

<u>סיכום:</u>

אתגרים שעמם התמודדנו:

זמן ריצת המנוע במחשבים הביתים:

בחרנו לכתוב פרסר מבוסס REGEXים שאנו עושה שימוש בIFים רבים, כלומר הגדרנו PATTERNים ברורים מראש לפי החוקים. תהליך זה דורש כוח חישוב חזק ולכן בחשבים הביתים זמני הרציה נעו בסביבות ה15 דקות. לאחר הרצה במעבדות ועבודה עם מחשבים נייחים עם עוצמה משעמותית יותר זמן יצירת האינדקס קטן משמעותית למתחת ל4 דקות, זמן מהיר מאוד ביחס לשאר מנועי החיפוש.

שימוש בPATTERNים עובר תהליך אופטימזציה בJAVA שגורם ליעילות בזמני הריצה.

כתיבת רגקסים:

תהליך למידת השימוש בREGEX לקח זמן משמעותי בתחילת העבודה, עם זאת לטווח הארוך הארכיטקטורה העבודה שנשענה על אובייקט עבור כל סוג של חוק כך שקיים לו ביטוי רגולרי הוכיח את עצמו כיעיל מאוד ומשפר ביצועים.

אתגר הגדול ביותר פרוייקט:

מציאת זמן הריצה האופטמלי עבור אינדוקס והן גישה למסמכים.

הבעיה המרכזית איתה התמודדנו בעבודה היתה זמני הביצוע של התהליכים, עבור תהליך האינדוקס נאלצנו לשנות את מבנה התוכנית כיוון שזמן האינדוקס ערך הרבה מאוד זמן, בחרנו להשתמש בMAP REDUCE, שלמדנו בכיתה יחד עם שימוש בTHREDS . שימוש במיפוי לקבצים זמניים (סגמנטים) ואחר כך איחוד באופן ממוקבל הביא לזמן מינמלים.

בנוסף לכך, בחרנו לבצע שמירה של מיקום כל מונח במילון לפי BYTES לשימוש בRANDOM ACCESS שהועיל לנו מאוד זמני ריצה השאילתות.

המלצות לשיפור האלגורתים:

1.שמירת האינדקסים עבור כל מילה במסמך – שמירת האינדקסים הייתה יכולה לדרג את המונחים שמופיעים במיקומים קודמים במשקולות משמעותיות ביותר ואולי לעזור ברמת האחזור.

2.שמירת המידע בקבצים בינאריים במקום בקבצי text – יעיל יותר מבחינת זמני ריצה.