שחר רון 203018254
מקסים ברגילובסקי 205695612

דו"ח חלק 2- מנוע חיפוש:

a. הסבר מפורט על אופן פעולת המנוע- הסבר על המחלקות שהוספנו, את מטרתן וכיצד הן פועלות.

הסבר מפורט על אופן פעולת המנוע:

פעולות הכנה למנוע:

1. מקבלים נתיב לindexFile איפה נשמור את כל הקבצי ה- posting
2. נקבל נתיב לcorpus שם נמצא גם קובץ הstop-words
3. קוראים מאמר מאמר ושולחים אותך ל- parser והוא שומר אותו במבני נתונים שנמצא במחלקת corpusDictionary. לאחר מעבר על 8 קבצים ה-indexer קורא מה- corpusDictionary ושומר את המידע בו כקבצים זמניים.
4. בסופם של כל האיטרציות על ה- corpus האינדקס מאחד כל הקבצים ויוצר את הקבצים הבאים:
   - קובץ InvertedIndex -posting עבור term בפורמט הבא: #term@df,tf,listOfOcc
   - documentInfo.txt- קובץ posting המכיל מידע על המסמכים בפורמט הבא: doc$max_tf||wordOfMaxtf||amount of unique words||Title||
   - dictionary.txt- מילון הנוצר מקבצי הposting המילון מכיל עבור כל terms את הפרטים הבאים:

term> The amount of documents is listed|poinet to posting file

   - articleIndex.txt- הקובץ נותן מספר בודד יחודי עבור כל key של מסמך.

כל הקבצים שנשמרים הם קבצי txt.

5. לאחר יצירת הקבצים מתבצעת חלוקה של הקובץ invertedIndex לתתי קבצים משניים לפי חלוקה לאותיות ומספרים.

עד כאן חלק א' של המנוע.

6. קיימות 2 אפשרויות לאחזור מסמכים:
   - להכניס נתיב לקובץ של שאילתות.
   - להכניס שאילתה חופשית בתיבת טקסט.
7. המחלקה Searcher קוראת ל- ReadQueries ושמה במחרוזת את כל השאילתה.
8. השאילתה נשלחת לפרסור במחלקה parser על מנת שתוכל לדבר "באותה שפה" עם קבצי ה-posting.
9. את המילים שהתקבלו מהparser נקרא מה- corpusDictionary.
10. במחלקה Searcher לאחר החזרה מה- corpusDictionary ניתן score לכל מילה מהשאילתה. המשקלים יתנו בהתאם לתנאים הבאים:
    - Online/ofline semantic
11. את המילים ביחד עם ה-score שלהם נשלח למחלקה Ranker
12. הRanker שולח את השאילתה למחלקה bm25.
13. המחלקה bm25 משתמשת במחלקה tfIdfCompute על מנת לחשב עבור כל מילה בכל מסמך את הדירוג שלה לפי הנוסחה של bm25.
14. ה-searcher לאחר קבלת הדירוגים מהמחלקה bm25 מדרג את כל המסמכים ומחזיר את 50 המסמכים הרלוונטים ביותר (בעלי הדירוג הגבוהה ביותר).
15. 50 המסמכים הרלוונטים ביותר מוצגים על המסך
16. ישנה אפשרות עבור כל מסמך להחזיר את 5 היישויות הרלוונטיות ביותר.

הסבר מפורט על המחלקות שהוספנו לחלק זה בpackage שקיימים כבר מחלק א':

Package model. corpusStracture

Class InfoTerms – מחלקת סינגלטון הקוראת את ה- invertedIndex וה- documentInfo כדי לייצר את מבני הנתונים שבעזרתם נייצר את tf וה- idf על מנת לייצר דירוג לכל מילה במסמך.

המחלקה המכילה את מבני הנתונים הבאים:

- HashMap <Integer,Double> infoTermInDoc- מחזיק עבור כל מספר מסמך עבור הterm בעל הכמות המקסימלית את הכמו שלו
- HashMap <Integer,Double> infoDocSize- מחזיק עבור כל מסמך את כמות ה- terms שיש בו.
- HashMap <Integer,String> infoDocSize- מחזיק עבור כל מספר מסמך (שאנחנו נתנו באופן זמני) את המספר מסמך האמיתי שלו.

שיטות של המחלקה:

- getInstance- השיטה מחזירה את הinstance של המחלקה לכל מקום בפרוייקט באופן בלעדי.
- GetMaxWordInDoc- השיטה מחזירה את הכמות של המילה בעלת המופע המקסימלית במסמך הנתון.
- getDocSize- השיטה מחזירה את מספר הterms במסמך הספציפי.
- getConvert- השיטה מחזירה את שם המסמך האמיתי עבור המספר מספר הזמני שאנחנו נתנו לו.
- getUnConvert- השיטה מחזירה עבור שם המספר המקורי את השם הזמני שאנחנו נתנו לו.

Package model. Indexer

Class UnPackingInvertedIndex- המחלקה מפצלת את הקובץ invertedIndex לתתי קבצים על פי חלוקה לאותיות /מילים.

שיטות של המחלקה:

- UnpackFile- השיטה מקבלת נתיב לקובץ **invertedIndex** ומפצלת אותו לתתי קבצים לפי אותיות/ מספרים.

הערה:

על מנת לשמור על סדר ה- package המחלקות שהוספנו קיימות בסעיף ב' ביחד עם המחלקות אותן נדרשו לממש.

b. הסבר מפורט של כל המחלקות הרלוונטיות לחלק זה.

Package model. Ranker

Class Ranker- המחלקה מקבלת נתיב לתיקיה. מייצרת אובייקט חדש של bm25 ומחזירה את הscore של כל המילים.

שיטות של המחלקה:

- Rank- המחלקה מקבלת hashMap שמכיל את המילים שקיימות בquery והדירוג עבור כל מילה ובנוסף אובייקט בוליאני על מנת לדעת עם התהליך הוא עם/ בלי stemming
  השיטה מחזירה HashMap<String,Double>- המכיל עבור כל term את הדירוג שלו על פי השיטה bm25.

Class BM25- המחלקה נעזרת במחלקות TF ו-IDF על מנת להחזיר עבור כל term את הדירוג שלו על פי הנוסחא BM25. המחלקה מאותחלת בכל הנתונים אודות ה- corpus הנתון כגון גודלו ואורך מסמך ממוצע.

שיטות של המחלקה:

- getSCoreQueries- השיטה נעזרת במחלקה TfIdfCompute על מנת להחזיר HashMap<String,Double> - עבור כל term קיים הscore שלו על פי הנוסחא.

Class TfIdfCompute - המחלקה נעזרת במילון וב- invertedIndex על מנת לייצר את מבני הנתונים הנדרשים עבור כדי לממש את הנוסחא הסופית של BM25.

שיטות של המחלקה:

- getTfIdf- השיטה מייצרת מהמילון ומה- **invertedIndex** את מבני הנתונים הבאים על מנת לאפשר ביצוע אחזור בעזרת bm25:
  i. HashMap<String,Double>- עבור כל term את ערך ה-IDF שלו
  ii. HashMap<String,Double>- עבור כל term את ערך ה-TF שלו עבור כל מסמך ספציפי
  השיטה מחזירה List עם שני מבני הנתונים הנ"ל.

Class OnlineSemantic- המחלקה משתמשת באתר DataMouseAPI למימוש המודל הסמנטי ב-online.

שיטות של המחלקה:

- searchSynonym- המחלקה מחזירה עבור המילות שקיבלנו בשאילתה מילות נוספות שיש להן קרבה סמנטית. כאשר המילים שנוספו במודל הסמנטי מקבלות ניקוד על פי הקרבה שלהם למילה המקורית מהשאילתה.

Class OflineSemantic- המחלקה משתמשת בword2Vec ומשתמש במילון עם משקלים שאומן לפני על corpus קטן. ניתן גם לצרף לו בנוסף קובץ שאומן על corpus גדול הרבה יותר (של ויקיפדיה). אך מכיוון שהcorpus הגדול שוקל 4GB לעומת 7MB

שיטות של המחלקה:

- searchSynonym- המחלקה מחזירה עבור המילים שקיבלנו בשאילתה מילות נוספות שיש להן קרבה סמנטית. כאשר המילים שנוספו במודל הסמנטי מקבלות ניקוד על פי הקרבה שלהם למילה המקורית מהשאילתה.

<u>הערה:</u> מכיוון שהמודל הנל מאומן על corpus קטן התוצאות שהוא מחזיר אינן טובות מספיק. על מנת שיחזיר תוצאות טובות יש להשתמש בקובץ שאומן על ויקפדיה אך הרצת קובץ זה תתאפשר רק על מחשבים בעלי זיכרון RAM גדול הרבה יותר מ16GB.

Package model. Searcher

Class Searcher - המחלקה מחזירה את 50 המסמכים הרלוונטים ביותר עבור השאילתה הנתונה

שיטות של המחלקה:

- getTop50ByQueryFile- השיטה מבצעת parser עבור השאילתה שקיבלה. ושולחת ל-RANKER את השאילתה וה-description (עבור כל מילה בשאילתה עם score 1 ועבור כל מילה בdescription score של 0.35) מנת להחזיר את 50 המסמכים הרלוונטים ביותר עבור שאילתה נתונה.

Class Searcher - המחלקה מחזירה את 50 המסמכים הרלוונטים ביותר עבור השאילתה הנתונה

שיטות של המחלקה:

- getTop50ByQueryFile- השיטה מבצעת parser עבור השאילתה שקיבלה. ושולחת ל-RANKER את השאילתה וה-description (עבור כל מילה בשאילתה עם score 1 ועבור כל מילה בdescription score של 0.35) מנת להחזיר את 50 המסמכים הרלוונטים ביותר עבור שאילתה נתונה.

Class ReadQuries – המחלקה מחזירה את השאילתה וה-description שלה עבור כל query מהקובץ שאילתות הנתון בפורמט שקיים במודל

שיטות של המחלקה:

- getQueries- השיטה מוציאה את מילות השאילתה וה-**description** שלה ומכניסה אותה למבנה נתונים HashMap<Integer, String> - עבור כל מספר שאילתה המילים הרלוונטיות אליה.

Class Entities – המחלקה מחזירה את 5 הישוויות שקיימות הכי הרבה במסמך מסוים.

שיטות של המחלקה:

- get5EntitiesForEachDoc- השיטה עוברת על המילון והIndexer ועבור מסמך ספציפי מחזירה מערך של חמשת הישויות בעלות מספר המופעים הרב ביותר במסמך.

Class ConvertDocName – המחלקה ממירה את מספר המסמך הזמני שאנחנו נתנו למסמך למספר המסמך המקורי- עבור 50 המסמכים הרלוונטים ביותר עבור השאילתה הספציפית.

שיטות של המחלקה:

- convert- השיטה מחזירה מערך של 50 השמות המקוריים של המסמכים הרלוונטים ביותר לשאילתה מסויימת.

Package model. Exceptions

Class EmptyTextFieldException – זורק שגיאה במקרה שלא הוקלד נתיב ב**TextField**
Class NoCorpusPathException – זורק שגיאה במקרה שלא הוקלד נתיב **CorpusPath**
Class NoDicInTheFieldException- זורק שגיאה במקרה שנתתי path אבל לא קיים מילון ב-path
Class NoDictionaryLoadedException- זורק שגיאה במקרה שלא הוטען מילון.
Class NoIndexerPathInsertedException – זורק שגיאה אם לא הוכנס נתיב לindexer
Class NoPathToSaveResultsException – זורק שגיאה אם לא הוכנה נתיב לשמירה של התוצאות

c. הסבר על האלגוריתמים הכלולים במנוע:

i. אלגוריתם הדירוג-

השתמשנו באלגוריתם הדירוג BM25 אשר נעזר בשיטות – TF ו-IDF על מנת להגדיר משקלים לביטויים.

פירוט הרכיבים של הנוסחא:

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

הנוסחא עבור TF: $score * f_{ij}$ כאשר:

- עבור מילה מהשאילתה- score=1
- עבור מילה מה-description- score=0.4

הנוסחא עבור IDF: $\log_{10}(\frac{N}{df})$

קבועים:

K=1.8

B=0.7

בחרנו בקבועים אלה לאחר ניסויים רבים מכיוון שהם אחזרו את התוצאות בצורה הטובה ביותר.

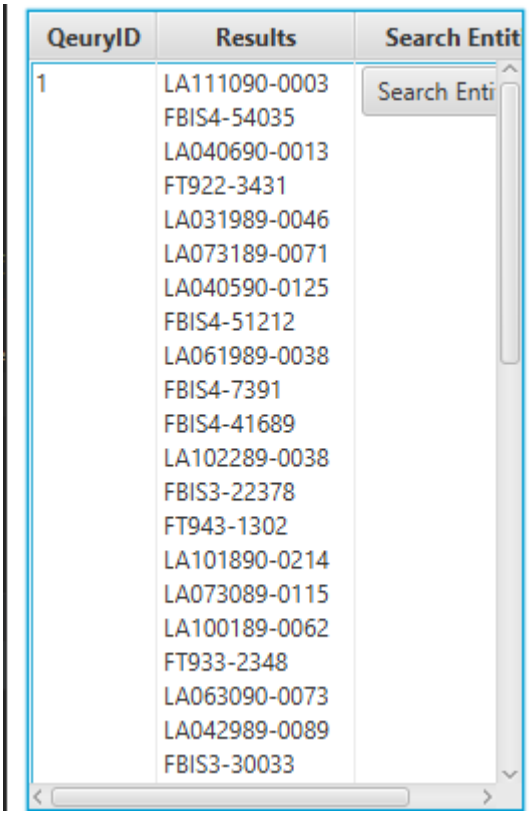
<u>דוגמא לאלגוריתם הדירוג שלנו איך הוא מחשב את הציון למסמך :</u>

```
For File ::471637::
Start to Compute the score of the Article
For the term : ocean in the query
with k = 1.8
b = 0.7
idfNumber = 2.0849219945176576
frequncy * wight = 2.0
length of the document :416.0
avrgDocLength : 450
The score for the term = 3.1514692208213346 =2.0849219945176576*((2.0*(1.8+1)/(+2.0 + k*(1-b+b*(416.0/450)))))
```

ניתן לראות כי עבור המסמך : 471637

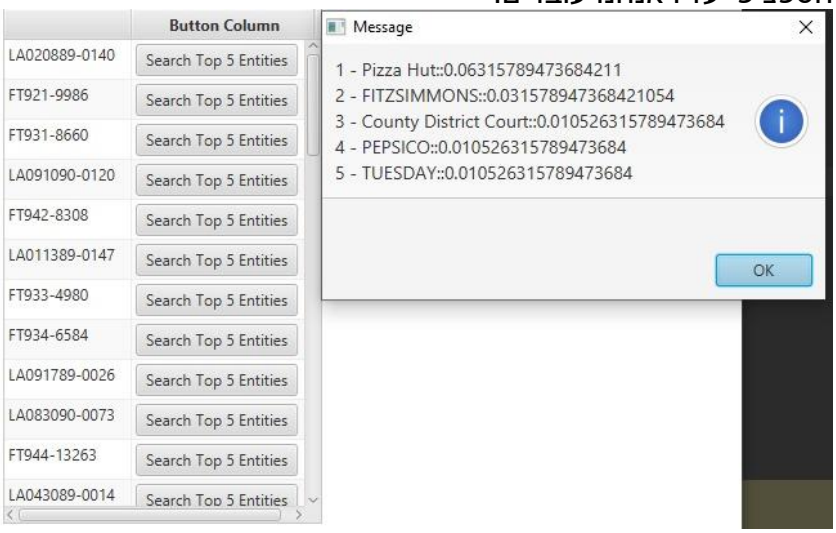
המילה ocean בשאילתה הופיע פעמיים מכיוון שהמשקל של מילה שנמצאת בשאילתה היא 1 ולכן סה"כ ה frequency=2 לאחר מכיוון שהציון שניתן למסמך הנ"ל הוא 3.15 לפי נוסחאות הbm-25.

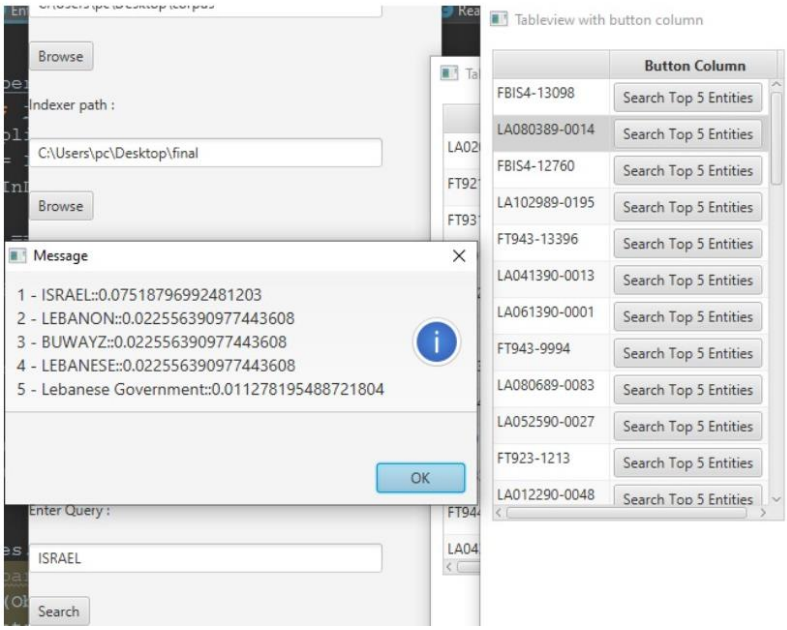
דוגמא לפלט של חמישים המסמכים הרלוונטיים ביותר שהוחזרו עבור המנוע :



QeuryID	Results	Search Entit
1	LA111090-0003	Search Enti
	FBIS4-54035	
	LA040690-0013	
	FT922-3431	
	LA031989-0046	
	LA073189-0071	
	LA040590-0125	
	FBIS4-51212	
	LA061989-0038	
	FBIS4-7391	
	FBIS4-41689	
	LA102289-0038	
	FBIS3-22378	
	FT943-1302	
	LA101890-0214	
	LA073089-0115	
	LA100189-0062	
	FT933-2348	
	LA063090-0073	
	LA042989-0089	
	FBIS3-30033	

ii. **אלגוריתם למציאת 5 הישויות הדומיננטיות במסמך, כולל 2 דוגמאות.**
האלגוריתם שבחרנו הוא משתמש בנוסחה של TF- נותן משקל עד כמה המילה דומיננטית במסמך. כאשר הנרמול בנוסחה זו הוא על ידי כמות המילים הייחודית במסמך הנוסחה מזהה עד כמה הישות דומיננטית במסמך הספציפי עליו אנחנו עובדים.





דוגמא איך הוא מדרג ישויות עבור מסמך :

```
For the term :ARAB the score is :0.007518796992481203by the formula :2/266.0
For the term :BFN the score is :0.0037593984962406013by the formula :1/266.0
For the term :BFN Text In the score is :0.0037593984962406013by the formula :1/266.0
For the term :BUWAYZ the score is :0.022556390977443608by the formula :6/266.0
For the term :Council Resolution the score is :0.0037593984962406013by the formula :1/266.0
For the term :EGYPT the score is :0.0037593984962406013by the formula :1/266.0
For the term :FARIS the score is :0.0037593984962406013by the formula :1/266.0
For the term :Faris Buwayz the score is :0.0037593984962406013by the formula :1/266.0
For the term :Foreign Minister Faris the score is :0.0037593984962406013by the formula :1/266.0
For the term :ISRAEL the score is :0.07518796992481203by the formula :20/266.0
For the term :ISRAELI the score is :0.0037593984962406013by the formula :1/266.0
For the term :ISRAELIS the score is :0.0037593984962406013by the formula :1/266.0
For the term :JORDAN the score is :0.0037593984962406013by the formula :1/266.0
For the term :LEBANESE the score is :0.022556390977443608by the formula :6/266.0
For the term :LEBANON the score is :0.022556390977443608by the formula :6/266.0
For the term :Lebanese Government the score is :0.011278195488721804by the formula :3/266.0
For the term :Middle East the score is :0.0037593984962406013by the formula :1/266.0
For the term :Minister Faris Buwayz the score is :0.0037593984962406013by the formula :1/266.0
For the term :PALESTINIAN the score is :0.011278195488721804by the formula :3/266.0
For the term :SYRIA the score is :0.0037593984962406013by the formula :1/266.0
For the term :SYRIAN the score is :0.007518796992481203by the formula :2/266.0
For the term :Security Council Resolution the score is :0.0037593984962406013by the formula :1/266.
For the term :Syria Egypt the score is :0.0037593984962406013by the formula :1/266.0
For the term :Text In the score is :0.0037593984962406013by the formula :1/266.0
ISRAEL::0.07518796992481203 - 0.07518796992481203
LEBANON::0.022556390977443608 - 0.022556390977443608
BUWAYZ::0.022556390977443608 - 0.022556390977443608
LEBANESE::0.022556390977443608 - 0.022556390977443608
Lebanese Government::0.011278195488721804 - 0.011278195488721804
```

iii. אלגוריתם לשיפור סמנטי.

מימשנו 2 אלגורימים לטיפול סמנטי:

1. **Online-** משתמשת באתר DataMouseAPI למימוש המודל הסמנטי ב-online. מחזירה עבור המילות שקיבלנו בשאילתה מילות נוספות שיש להן קרבה סמנטית. כאשר המילים שנוספו במודל הסמנטי מקבלות משקל על פי הקרבה שלהם למילה המקורית מהשאילתה. אם המילה ממש קרובה אז היא נוספת לשאילתה עם משקל של 0.8 אם היא קרובה באופן בנוני נוספת עם משקל 0.2 לשאילתה אחרת לא נוספת.

```
The semantic word for the term ::
oceanis ::sea
the score between them by the online algorithm is4743 this term not added to the query because of the low score
```

https://www.datamuse.com/api

2. **Ofline**

- המחלקה משתמשת בword2Vec ומשתמשת במילון עם משקלים שאומן על corpus קטן. ניתן גם לצרף לו בנוסף קובץ שאומן על corpus גדול הרבה יותר (של ויקיפדיה). אך מכיוון שהcorpus הגדול שוקל GB4 לעומת MB7 קשה מאוד עד בלתי אפשרי להשתמש במחשבים שקיימים אצלנו.

האלגוריתם מחזיר עבור המילים שקיבלנו בשאילתה מילות נוספות שיש להן קרבה סמנטית. כאשר המילים שנוספו במודל הסמנטי מקבלות ניקוד על פי הקרבה שלהם למילה המקורית מהשאילתה.

<u>הערה</u>: מכיוון שהמודל הנל מאומן על corpus קטן התוצאות שהוא מחזיר אינן טובות מספיק. על מנת שיחזיר תוצאות טובות יש להשתמש בקובץ שאומן על ויקפדיה אך הרצת קובץ זה תתאפשר רק על מחשבים בעלי זיכרון RAM גדול הרבה יותר מGB16.

מוסיפה לכל מילה בשאילתה עוד מילה עם הקרבה הסמנטית הגובה ביותר עם משקל של 0.05.

```
The semantic word for the term ::
oceanis ::atlantic
 this term added to the query with the wight of 0.05
querie Number ::1 took ::5.086
```

https://code.google.com/archive/p/word2vec
https://github.com/medallia/Word2VecJava

d. הסבר על הנתונים בקובצי ה- posting התומכים באלגורתמים שמימשנו.

- **עבור הקובץ dictionary**

Alan Gleicher>2|115570

אנחנו משמתמשים בdictionary על מנת לבדוק בנקודה הראשונית באחזור- ברגע קבלת השאילתה לאחר הוספת כל המילים שיכולות להתווסף אליה (description ומודל סמנטי אם מופעל) האם המילה קיימת אצלנו במילון. אם המילה לא קיימת היא נמחקת ואינה לוקחת חלק בתהליך הדירוג והאחזור.

- **עבור הקובץ invertedIndex:**

financially-driven@**246141,1#**264274,1#280188,1#287355,1#296187,1#

  - עבור כל terms אנחנו מחזיקים את מספר המסמך הזמני שנתנו לו, כמות הפעמים שהterm מופיע במסמך.
  - **246141-** מספר המסמך
  - **1-** כמות המופעים במסמך **246141** של financially-driven – השתמשנו עבור הנוסחא בTF – מספר הפעמים שהביטוי מופיע במסמך.
  - **#-** מספר ה-# במסמך מלמד על כמות המסמכים שה-term מופיע בcorpus כולו. השתמשנו בנתון זה בנוסחא של IDF ב-df- מספר המסמכים שמכילים את הביטוי t במאגר.

- **עבור הקובץ DocumentInfo:**

1@512||PARTY||14|| FORMER YUGOSLAV REPUBLIC OF MACEDONIA: OPINION POLLS ON

  - 1- מספר המסממך- על מנת לצמצם את גודל הposting המרנו את שמות המסמכים למספרים.
  - 512- כמות המילים הייחודיות במסמך. בהתחלה נרמלנו את TF בכמות המילים הייחודיות אך בסופו של דבר החלטנו שלא להשתמש בנרמול ב-TF כיון שכך הצלחנו לאחזר מספר רב יותר של מסמכים.
  - 14- מספר המופעים של המילה בעלת התדירות הגבוהה ביותר במסמך. בהתחלה נרמלנו את TF בכמות המילים הייחודיות אך בסופו של דבר החלטנו שלא להשתמש בנרמול ב-TF כיון שכך הצלחנו לאחזר מספר רב יותר של מסמכים.
  - Title- הוספנו את ה title למילות השאילתה ולdescription על מנת לנסות לאחזר בעזרתנו מספר רב יותר של מסמכים אך כותרת המסמך לא הצליחה לשפר את האחזור ואף גרעה ממנו. לכן החלטנו להוריד את השימוש בכותרת.

- **עבור הקובץ articleIndex:**

FBIS3-1 -1

עבור כל שם מסמך שקיים ב- corpus נתנו לו מספר זמני על מנת להקטין את גודל הקורפוס. ההמרה מתבצעת רק פעם אחת ורק עבור 50 המסמכים הרלוונטים ביותר ולכן בסה"כ ההמרה מקטינה באופן משמעותי את כמות הזיכרון ולא מגדילה באופן משמעותי את זמן הריצה.

e. הסבר על השימוש בפרוייקט בקוד פתוח. לפרט את השירות, כתובת, איכן וכיצד השתמשנו.

- **אלגוריתם סמנטי online:**

- Class OnlineSemantic- המחלקה משתמשת באתר DataMouseAPI למימוש המודל הסמנטי ב-online.

https://www.datamuse.com/api

- **אלגוריתם סמנטי ofline:**
  Class OflineSemantic- המחלקה משתמשת בword2Vec ומשתמש במילון עם משקלים שאומן לפני על corpus קטן. ניתן גם לצרף לו בנוסף קובץ שאומן על corpus גדול הרבה יותר (של ויקיפדיה). אך מכיוון שהcorpus הגדול שוקל GB4 לעומת MB7

https://code.google.com/archive/p/word2vec
https://github.com/medallia/Word2VecJava

2. הערכה של המנוע- פלט בלי stemming

Queryid (Num):	351	
Text :	Falkland petroleum exploration	What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?
Precision	0.4	
Recall	0.4167	
Precision@5	0	
Precision@15	0.1333	
Precision@30	0.333	
Precision@50	0.4	
Time:	5.032 sec	
Queryid (Num):	352	
Text :	British Chunnel impact	What impact has the Chunnel had on the British economy and/or the life style of the British?
Precision	0.32	
Recall	0.065	
Precision@5	0	
Precision@15	0.2	
Precision@30	0.3	
Precision@50	0.32	
Time:	8.193 sec	
Queryid (Num):	358	
Text :	blood-alcohol fatalities	What role does blood-alcohol level play in automobile accident fatalities?
Precision	0.44	
Recall	0.43	
Precision@5	0.2	
Precision@15	0.5333	
Precision@30	0.5	
Precision@50	0.44	
Time:	4.119 sec	

Queryid (Num):	359	
Text :	mutual fund predictors	Are there reliable and consistent predictors of mutual fund performance?
Precision	0.12	
Recall	0.25	
Precision@5	0	
Precision@15	0.1333	
Precision@30	0.1667	
Precision@50	0.12	
Time:	4.687 sec	
Queryid (Num):	362	
Text :	human smuggling	Identify incidents of human smuggling.
Precision	0.14	
Recall	0.23	
Precision@5	0	
Precision@15	0.1	
Precision@30	0.2	
Precision@50	0.14	
Time:	2.277 sec	
Queryid (Num):	367	
Text :	piracy	What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?
Precision	0.34	
Recall	0.08	
Precision@5	0.6	
Precision@15	0.2	
Precision@30	0.2333	
Precision@50	0.34	
Time:	4.834 sec	

Queryid (Num):	373	
Text :	encryption equipment export	Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.
Precision	0.12	
Recall	0.375	
Precision@5	0.2	
Precision@15	0.3333	
Precision@30	0.2	
Precision@50	0.12	
Time:	5.355 sec	
Queryid (Num):	374	
Text :	Nobel prize winners	Identify and provide background information on Nobel prize winners.
Precision	0.48	
Recall	0.11	
Precision@5	1	
Precision@15	0.533	
Precision@30	0.5667	
Precision@50	0.48	
Time:	3.697 sec	
Queryid (Num):	377	
Text :	cigar smoking	Identify documents that discuss the renewed popularity of cigar smoking.
Precision	0.22	
Recall	0.305	
Precision@5	0	
Precision@15	0.0667	
Precision@30	0.2	
Precision@50	0.22	
Time:	5.493 sec	

Queryid (Num):	380	
Text :	obesity medical treatment	Identify documents that discuss medical treatment of obesity.
Precision	0.06	
Recall	0.428	
Precision@5	0	
Precision@15	0.0667	
Precision@30	0.0667	
Precision@50	0.06	
Time:	3.109 sec	
Queryid (Num):	384	
Text :	space station moon	Identify documents that discuss the building of a space station with the intent of colonizing the moon.
Precision	0.26	
Recall	0.254	
Precision@5	0.6	
Precision@15	0.2667	
Precision@30	0.2333	
Precision@50	0.26	
Time:	7.536 sec	
Queryid (Num):	385	
Text :	hybrid fuel cars	Identify documents that discuss the current status of hybrid automobile engines, (i.e., cars fueled by something other than gasoline only).
Precision	0.28	
Recall	0.164	
Precision@5	0	
Precision@15	0.0667	
Precision@30	0.2333	
Precision@50	0.28	
Time:	8.535 sec	

Queryid (Num):	387	
Text :	radioactive waste	Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.
Precision	0.22	
Recall	0.15	
Precision@5	0.2	
Precision@15	0.2	
Precision@30	0.1667	
Precision@50	0.22	
Time:	6.0 sec	
Queryid (Num):	388	
Text :	organic soil enhancement	Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.
Precision	0.22	
Recall	0.22	
Precision@5	0.4	
Precision@15	0.333	
Precision@30	0.3667	
Precision@50	0.22	
Time:	7.637sec	
Queryid (Num):	390	
Text :	orphan drugs	documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.
Precision	0.28	
Recall	0.1147	
Precision@5	0.5	
Precision@15	0.4667	
Precision@30	0.4	
Precision@50	0.28	
Time:	4.715 sec	
average precision over all rel docs		0.0735

הערכה של המנוע- פלט עם stemming

Queryid (Num):	351	
Text :	Falkland petroleum exploration	What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?
Precision	0.46	
Recall	0.479	
Precision@5	0	
Precision@15	0.2667	
Precision@30	0.4	
Precision@50	0.46	
Time:	5.191 sec	
Queryid (Num):	352	
Text :	British Chunnel impact	What impact has the Chunnel had on the British economy and/or the life style of the British?
Precision	0.26	
Recall	0.0528	
Precision@5	0	
Precision@15	0.2	
Precision@30	0.2333	
Precision@50	0.26	
Time:	8.065 sec	
Queryid (Num):	358	
Text :	blood-alcohol fatalities	What role does blood-alcohol level play in automobile accident fatalities?
Precision	0.4	
Recall	0.392	
Precision@5	0.2	
Precision@15	0.4667	
Precision@30	0.4333	
Precision@50	0.4	
Time:	4.728 sec	

Queryid (Num):	359	
Text :	mutual fund predictors	Are there reliable and consistent predictors of mutual fund performance?
Precision	0.12	
Recall	0.214	
Precision@5	0	
Precision@15	0.1333	
Precision@30	0.1667	
Precision@50	0.12	
Time:	4.793 sec	
Queryid (Num):	362	
Text :	human smuggling	Identify incidents of human smuggling.
Precision	0.16	
Recall	0.205	
Precision@5	0	
Precision@15	0.1333	
Precision@30	0.1667	
Precision@50	0.16	
Time:	2.708 sec	
Queryid (Num):	367	
Text :	piracy	What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?
Precision	0.34	
Recall	0.091	
Precision@5	0.6	
Precision@15	0.2	
Precision@30	0.2333	
Precision@50	0.34	
Time:	5.554 sec	

Queryid (Num):	373	
Text :	encryption equipment export	Identify documents that discuss the concerns of the United States regarding the export of encryption equipment.
Precision	0.12	
Recall	0.375	
Precision@5	0.2	
Precision@15	0.3333	
Precision@30	0.2	
Precision@50	0.12	
Time:	4.427 sec	
Queryid (Num):	374	
Text :	Nobel prize winners	Identify and provide background information on Nobel prize winners.
Precision	0.4	
Recall	0.098	
Precision@5	0.6	
Precision@15	0.4	
Precision@30	0.3667	
Precision@50	0.4	
Time:	4.298 sec	
Queryid (Num):	377	
Text :	cigar smoking	Identify documents that discuss the renewed popularity of cigar smoking.
Precision	0.26	
Recall	0.4166	
Precision@5	0	
Precision@15	0.0667	
Precision@30	0.2	
Precision@50	0.26	
Time:	4.869 sec	

Queryid (Num):	380	
Text :	obesity medical treatment	Identify documents that discuss medical treatment of obesity.
Precision	0	
Recall	0.428	
Precision@5	0	
Precision@15	0.1333	
Precision@30	0.0667	
Precision@50	0.06	
Time:	2.538 sec	
Queryid (Num):	384	
Text :	space station moon	Identify documents that discuss the building of a space station with the intent of colonizing the moon.
Precision	0.22	
Recall	0.215	
Precision@5	0.6	
Precision@15	0.2667	
Precision@30	0.2	
Precision@50	0.22	
Time:	6.549 sec	
Queryid (Num):	385	
Text :	hybrid fuel cars	Identify documents that discuss the current status of hybrid automobile engines, (i.e., cars fueled by something other than gasoline only).
Precision	0.34	
Recall	0.2	
Precision@5	0	
Precision@15	0.0667	
Precision@30	0.2	
Precision@50	0.34	
Time:	7.399 sec	

Queryid (Num):	387	
Text :	radioactive waste	Identify documents that discuss effective and safe ways to permanently handle long-lived radioactive wastes.
Precision	0.2	
Recall	0.136	
Precision@5	0.2	
Precision@15	0.1333	
Precision@30	0.1333	
Precision@50	0.2	
Time:	6.444 sec	
Queryid (Num):	388	
Text :	organic soil enhancement	Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.
Precision	0.34	
Recall	0.34	
Precision@5	0.6	
Precision@15	0.5333	
Precision@30	0.5	
Precision@50	0.34	
Time:	6.872 sec	
Queryid (Num):	390	
Text :	orphan drugs	documents that discuss issues associated with so-called "orphan drugs", that is, drugs that treat diseases affecting relatively few people.
Precision	0.3	
Recall	0.122	
Precision@5	0.2	
Precision@15	0.2667	
Precision@30	0.3333	
Precision@50	0.3	
Time:	5.215 sec	
average precision over all rel docs		0.0769

3. בעיות שנתקלנו בהם וכיצד התמודדתם איתן. מה האתגר הגדול לדעתכם בפרויקט. המלצות לשיפור האלגוריתם שלכם/מה הייתם עושים אחרת?..

הבעיות שנתקלנו בהם במהלך הפרוייקט:

- התמודדות עם פרוייקט גדול בו כל שלב מסתמך על השלב הקודם- היה קושי לחזור לחלקים קודמים בקוד ולתקן בו דברים שהיה לנו צורך בהם בשלבים מתקדמים יותר.
- עבודה מול מאגר מסמכים גדול מאוד ביחס למה שאנחנו מכירים- קושי גדול להבין בעיות שנובעות מריצה של עשרות אלפי מסמכים.

כיצד התמודדנו עם הבעיות:

- ההבנה הגדולה ביותר שקיבלנו מפרוייקט זה הוא הצורך לבדוק כל קטע קוד, גם הקטן ביותר לעומק ועבור כל מקרי הקצה על קלטים קטנים כיוון שברגע שהרצנו את הקוד על עשרות אלפי מסמכים היה קושי גדול לזהות בעיות ובפרט את המקור לבעיות.

האתגר הגדול ביותר שנתקלנו בו בפרוייקט היה ההתמודדות עם מאגר גדול מאוד של מסמכים והיכולת להבין/ לזהות שגיאות בזמן הרצת הפרוייקט על כל הcorpus.

מכיוון שהcorpus עליו אנחנו עובדים הוא סטטי לשיפור האלגוריתם ולאחזור טוב יותר של מסמכים היינו יכולים לממש אלגוריתם אחזור נוסף כגון CosSim ולבדוק איזה מודל מאחזר טוב יותר עבור ה corpus הנתון.