**מגישים:**

עידן רזי 204475255

אביב תורג'מן 203959523

**דוח פרויקט מנוע אחזור חלק ב':**

1. מבנה הפרויקט:

**מחלקות חדשות**

Searcher

* parseQuery – מפרסרת את השאילתה ונותנת לה דירוג תוך שימוש במחלקת ranker
* parseQuerys – מפרסרת קבוצת שאילתות אחת אחרי השניה תוך שימוש ב-parseQuery
* show -מציגה את המסמכים הרלונטים שחזרו מביצוע הדירוג
* top5 מחשבת את 5 המשפטים החשובים ביותר במסמך
* showTop5 – מציגה את 5 המשפטים לפי הסדר המבוקש

Ranker

* Calculate\_idf – חישוב df מנורמל למילה.
* Calaculate\_itf – חישוב tf מנורמל **בגדול המילה** עם הtf המקסימלי במסמך
* Cosim – חישוב דמיון בין המאגר לבין השאילתא ע"פ נוסחת cosim מחזיר מילון doc,cosim
* Getdocs - הפעלת שיטות דמיון והחזרת 50 המסמכים הכי רלוונטים ע"פ סדר חשיבות.
* Getdocweight – החזרה את המשקל המנורמל של מסמך (מכנה של נוסחת cosim)
* addTolist+writeList – פונקציות עזר שרשומות את התוצאות לresults.txt ע"פ פורמט של תוכנת qrel
* bm25 - מחשבת דמיון בין המאגר לבין השאילתא ע"פ נוסחת bm25 , מחזיר מילון doc,bm25sim
* Combine – מאחדת את תוצאות הדמיון של שיטות החישוב ע"פ משקלים שהגדרנו

**תוספות למחלקות מחלק א'**

Files

* sourceCode- שולחת בקשה http לקבלת קוד המקור של דף הויקיפדיה המבוקש
* viki – מפרסרת את הערכים איתם נרחיב את השאילתה
* querys -מפרסרת את השאילתות מתוך הקובץ הנתון
* description- מפרסרת את התיאור של השאילתות מתוך הקובץ הנתון
* top5- מפרסר את כל המשפטים מתוך המסמך המבוקש

Control

* choose\_btn\_Click- מבצע את דירוג 15 השאילתות מתוך הקובץ הנתון
* reset\_btn\_Click – מאפס את כל הנתונים השמורים בזיכרון
* run\_query\_btn\_Click – מתחיל את תהליך הדירוג עבור השאילתה בהתאם לבחירת מהשתמש (הרחבה,חיפוש במסמך)
* save\_qry\_Click -שומר את תוצאות הדירוג במקום שיבחר על ידי המשתמש

Indexer

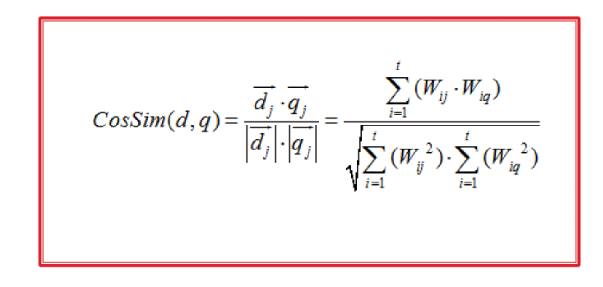
* calculate\_w – מחשבת בזמן בניית המאגר משקל של כל מסמך –
* post\_list – מחשבת tf של כל מסמך ע"פ שורת postlist (לדג' למילה dog הפונקציה תקרא מהpostlist את שורה הרלוונטית לterm ותחשב לכל מסמך את הtf) מחזירה רשימה ממונית ע"פ tf הגדול ביותר.
* Compute\_avg\_docsize – מחשבת את אורך המסמך הממוצע במאגר.

**אלגוריתם להרחבת שאילתא באמצעות האינטרנט**

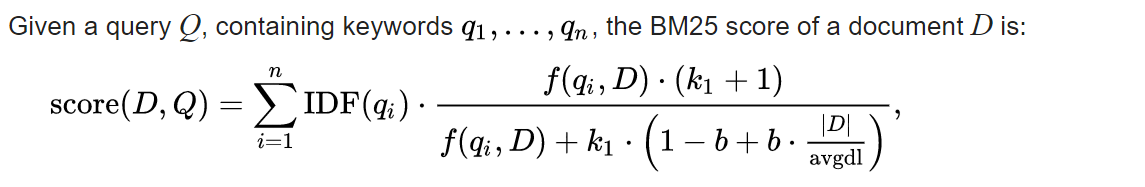
בתהליך הרחבת השאילתה ראשית סקרנו את אתר ויקיפדיה על מנת הבין איזה מידע אנו צריכים ממנו. בויקיפדיה יש תת קטגוריה בשם ראו גם (see also) שבה יש ערכים דומים למה שחיפשנו לכן החלטנו לחלץ אותה מתוך האתר, לשם כך שלחנו בקשת http בשיטת get לקבלת קוד המקור של אתר ויקיפדיה כאשר אנו מזריקים את נושא החיפוש אל כתובת ה-url של האתר. לאחר קבלת קוד המקור התחלנו לחתוך את המידע שהתקבל (דף html) על מנת להישאר רק עם הערכים הרצויים. לבסוף הוספנו את הערכים הרצויים לשאילתה (שרשור לשאילתה) וביצענו פעולת group by לשאילתה על מנת למנוע כפילויות. את התוצאה סופית שלנו לביצוע דירוג.

**אלגוריתם הדירוג:**

1. ביצוע עיבוד במחלקת Searcher - ביצוע Parse וStem .
2. ביצוע חישוב cosim לשאילתא ע"פ הנוסחה



1. ביצוע חישוב בעזרת נוסחת bm25 : כאשר k=1 ו b=0.5 , המקדמים נבחרו ע"י ניסויים והשוואה לtrec.



<https://en.wikipedia.org/wiki/Okapi_BM25>

**bm25 היא נוסחת bag of words בדומה לcosim , משתמשת גם במדדי idf ,tf**

1. איחוד החישובים ע"י הנוסחה הבאה , נקבע ע"פ ניסויים.

**נתונים בposting ובמילונים:**

מילונים:

* מילון המילים הראשי : term => idf,post\_position, total frequency
* מילון cache : term=>postlist row.
* מילון DOC : docname=>file,maxtf,size,W

כאשר W - סכום המשקלים של המילים במסמך . משמש לחישוב cosim.

מבנה רשימת הפוסט : post => doc1,x,y#doc2,x,y כאשר # משמש כדלימטר בין המסכים. המספרים מייצגים את כמות המופעים של המיליה במסמך , החלוקה x,y כאשר x כמות המופעים עד ה100 מילים הראשונות וy כמות המופעים אחרי. המטרה הייתה להדגיש בנוסחת הדירוג את המילים שמופיעות בתחילת המסמך אך הדבר התגלה כלא יעיל בהשוואה למאגר הtrec.

**אלגוריתם למציאת חמשת מילים החשובות ביותר:**

איתרנו את המסמך המבוקש וחילקנו אותו למשפטים על פי סימני פיסוק.

עבור כל משפט ספרנו:

Length- כמות המילים במשפט

Numbers -כמות המספרים המשפט

Capital -מספר המילים שמכילים לפחות אות גדולה אחת לפחות

Long -כמות המילים שאורכם גדול מ-3

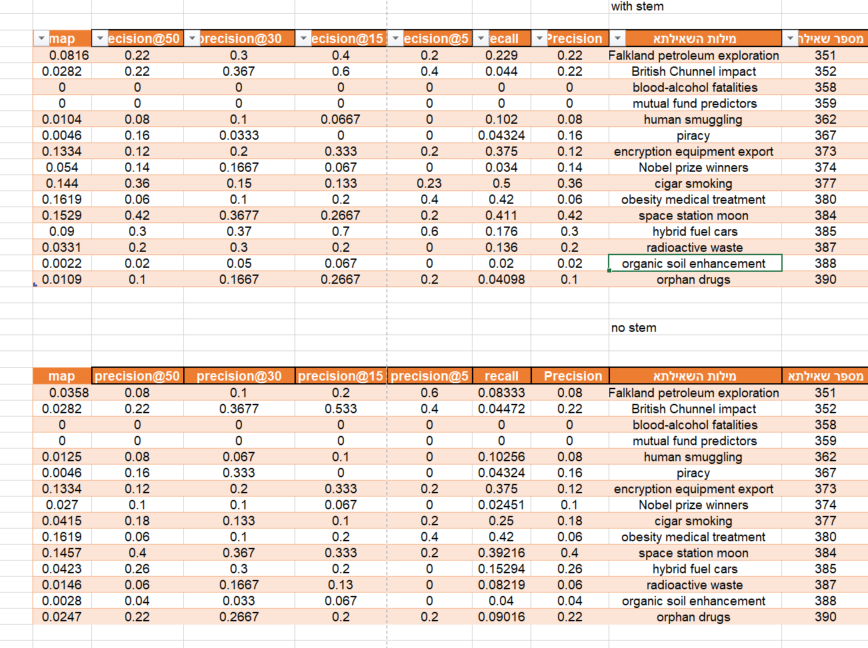
Sentences -כמות המשפטים במסמך

Location -מיקום המשפט בתוך המאמר

נוסחא:

Rank=

**2.הערכת המנוע:**



ניתן לראות שכאשר בצענו ניתוח בעזרת stem הצלחנו להחזיר יותר מסמכים רלוונטים ולכן עדיף לבצע את כל התהליך ביחד עם stem.

עם stem הצלחנו להחזיר 120 מסמכים רלוונטים בלי חזרו רק 102.

**סיכום:**

במהלך הפרויקט נתקלנו בשני קשיים עיקריים :

1. התמודדות עם דרישות הפרויקט.
2. תכנון הפרויקט וקבעת לוחות זמנים.

התמודדות עם דרישות הפרויקט:

התמודדות עם מאגר מידע גדול וניתוחו בזמן ריצה יעיל הווה אתגר, נאצלנו להשקיע זמן רב בתכנון הפרויקט , ביצוע מספר מימושים שונים ובחינת יעילות, ובסופו של דבר ביצוע קוד מקבילי כדי לבצע ביצועיים יעלים.

תכנון הפרויקט וקביעת לוחות זמנים:

במסגרת הקורס נתקלנו בפעם הראשונה בתואר בפרויקט בסדר גדול כזה, בנוסף הפרויקט נפרש לכל אורך הסמסטר בניגוד לצורה הסטנדרטית שנתקלנו בה במהלך התואר של עבודות שמחולקות לטווחי זמנים קצרים יותר. סגנון הפרויקט הזה דרש מאתנו לתכנן יעדים ולוחות זמנים כדי שנוכל לעמוד בדרישות הפרויקט.

במהלך התכנון דאגנו לחלק את העבודה לתחומים ושונים כדי שנוכל לעבוד במקביל לדג' מחלקת הparser ומחלקת הindexer שחולקה ביננו , ובסופו של דבר לשלב ולאחד את חלקי הפרויקט.

**המלצות לייעול המנוע :**

אם היה יותר זמן במהלך הסמסטר לשלב שיטת דירוג המבוססת על **CONCEPTS** כמו שלמדנו במהלך הקורס.

בניית משקל לשאילתה על בסיס נתונים של מודלי שפה.