א)

ב) קישור ל:

https://github.com/tal-benhamou/IR-Project-talido.git

ג) קישור ל

https://console.cloud.google.com/storage/browser/irproject315450569

ד) כל הקבצים שבהם נעזרנו לבנות את האינדקסים נמצאים בקישור לבסעיף ב.  
תחת התיקייה בשם בקובץ בשם   
הסבר על הקבצים איתם בנינו את האינדקסים -

\* לטובת מימוש פונקציה בנינו אינדקס ל *ללא סטמינג כפי שנדרש, אשר כל שמור בצורה הבאה:*

*ה הינו המילה, וה הינו רשימה של טאפלים עם 4 איברים בפורמט הבא:*

*בנוסף בקובץ זה ניתן לראות תיעוד של כתיבת מיליון ל בפורמט הבא:*

וזאת על מנת לעמוד בדרישות החזרה של הפונקציות בקובץ .

ניתן לראות תיעוד בקובץ זה לטובת כתיבה של כל האינדקס ל.

\* לטובת מימוש פונקציה בנינו אינדקס ל *ללא סטמינג כפי שנדרש, אשר כל שמור בצורה הבאה:*

*ה הינו המילה, וה הינו רשימה של טאפלים עם 4 איברים בפורמט הבא:*

ניתן לראות תיעוד בקובץ זה לטובת כתיבה של כל האינדקס ל.  
\* לטובת מימוש פונקציה ו בנינו שתי אינדקסים נוספים, אחד ל *והשני ל , אשר כל שמור בצורה הבאה:*

*ה הינו המילה, וה הינו רשימה של טאפלים עם איבר יחיד בפורמט הבא:*

ניתן לראות תיעוד בקובץ זה לטובת כתיבה של כל האינדקס ל.

ה) תחילה ביצענו ניסיון על מנת להבין אילו מדדים נרצה להשתמש בפונקציית

לטובת כך בדקנו שימוש במדדים שונים, תחילה עשינו איחוד בין התוצאות של כל זוג אפשרי מבין 2 אינדקסים בלבד מתוך ה . הטסטים הרלוונטיים לבדיקה זו נמצאים בתיקייה בקובץ test\_for\_serach\_function שנמצא ב().  
לאחר מכן ביצענו איחוד בין 3 אינדקסים.)   
וקיבלנו כי התוצאות הטובות ביותר הן של איחוד התוצאות בין האינדקסים כך ש ה מחושב ע"י מדד ו ה ע"י מדד .

כל התוצאות() מפורטות בתיקייה בקובץ בשם - test\_for\_serach\_function.  
ניתן למצוא את התוצאות עבור הטסט הנבחר , בקובץ test6\_BM25.txt.

בנוסף לאחר שבחרנו את המדדים שאנו רוצים לממש בפונקציית כלומר ההכי טובים למנוע שלנו, ניסנו למצוא את המשקלים הנכונים שיביאו לנו תוצאות אופטימליות.  
לטובת כך ביצענו אופטימיזציה בין כל קומבינציה אפשרית מתוך רשימות אלו:

כלומר 36 קומבינציות וקיבלנו כי המשקלים הטובים ביותר הם :

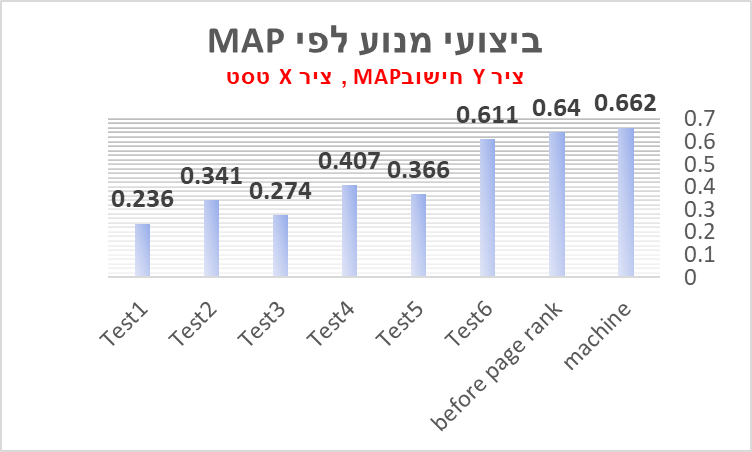
וביצענו אופטימיזציה נוספת עם משקלים אלו, על כל זוג אפשרי של משקלים מתוך:  
כלומר 12 קומבינציות וקיבלנו כי המשקלים הטובים ביותר הם :

התוצאות של האופטימיזציה הראשונה והשנייה מופיעות בתיקייה   
בקבצים אלה ב-testing\_wigths\_combinations , testing\_k\_b\_combinations   
התוצאות של עבור 30 שאילתות עם המשקלים שנמצאו האופטימליים מופיע בקובץ:  
testing\_best\_values   
באמצעות האופטימיזציה של המשקלים שיפרנו את התוצאה מ 0.57 ל 0.611 באינדקס הקטן.

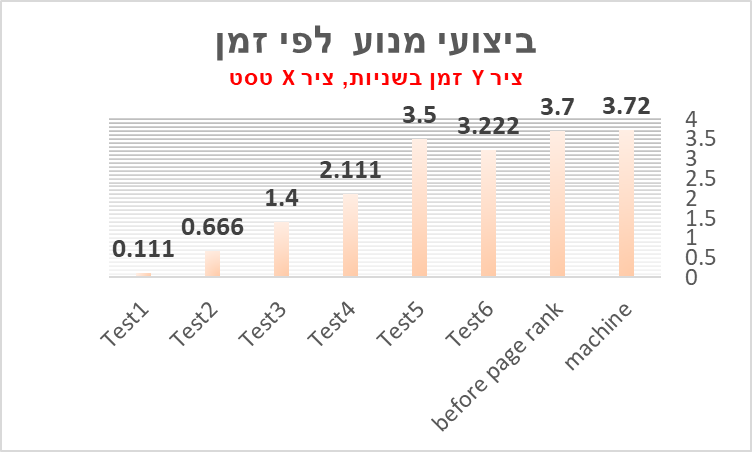
ו)

העמודות מסמלות את הגרסאות השונות אותם בדקנו, ומצאנו כי עמודה שמציגה את התוצאה הכי טובה לאינדקס הקטן, כלומר כפי שצוין בסעיף ה' הוא זה שהביא את התוצאה הטובה ביותר והוא זה שבחרנו למנוע

עמודת machine היא התוצאה הסופית מפונקציית של המנוע על האינדקס הגדול.



ז)  
 העמודות מסמלות את הגרסאות השונות אותם בדקנו, ניתן לראות כי הזמן הכי טוב הוא עבור איחוד בין 2 האינדקסים .  
אך אנו רצינו תוצאות טובות יותר למנוע מאשר הזמן, לכן בחרנו כפי שצוין לעיל.



ח) כפי שניתן לראות בקובץ AVG\_with\_10\_best\_Q\_with\_1\_lowest\_Qב את התוצאות הטובות ביותר שקיבלנו עבור המשקלים האופטימליים שמצאנו, השאילתה שנמצאה עם התוצאות הטובות ביותר היא : ושאילתה אחת שבה המנוע פחות הצליח היא באינדקס הקטן או  *באינדקס הגדול.  
בשאילתות עם התוצאות הטובות ניתן להגיד כי המדדים שבחרנו למנוע, ביחד מביאים קירוב אלגברי מספיק טוב לכל מילה בשאילתא מכיוון שלמשל תדירות המילה כנראה מופיעה בהרבה מסכמכים וגם בכותרות או אנקורים ולכן האיחוד של התוצאות יביא לנו מסמכים רלוונטיים יותר מאשר בשאילתות הפחות טובות. למשל השאילתא או הן בעלות משמעות עמוקה יותר מאשר המילים שיש בשאילתא עצמה, מכיוון שמסמכים על כסף או על סרט שנעשה ב 1992 לא יועילו לנו בשאילתא לכן תדירות המילה או חישוב לפי בשאילתא זו לא יחזיר לנו מסמכים רלוונטיים כנראה, הקירוב הלינארי יכול להשתפר אם היינו משתמשים במדדים כפי שלמדנו בסוף הקורס כמו .( או ).*