**דוח דרישות – פרויקט אחזור מידע**

1. מגישות:

רותם אמיר: 319041208 [amirrot@post.bgu.ac.il](mailto:amirrot@post.bgu.ac.il)

ליאור אפטבי: 209259993 [lioraft@post.bgu.ac.il](mailto:lioraft@post.bgu.ac.il)

1. קישור לגיט-האב:
2. קישור ל:Google Storage Bucket
3. רשימה של מסמכי אינדקס:
4. תיאור ניסויים: (כולל הערכות ומסקנות): כל הנסיויים הרצנו על המחשב דרך **הPyCharm**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 3.235 | 0.034 | 0.080 | 0.070 | 0.035 | 0.182 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.034 | 0.011 | 0.021 | 0.037 |  |  |

1. **ניסוי ראשון:** הרצת tf-idf בשילוב cosine similarity על כותרות בלבד.   
   בצענו pre-processing לכותרות על ידי סינון stopwords בלבד.  
   הניסוי כלל פתרון יצירתי בו ניסינו להחליף את ה-tf בנוסחה של cosine similarity בציון tf-idf. בניסוי זה, מנוע החיפוש מחזיר 100 מסמכים.  
   תוצאות הניסוי: ממוצע המדדים

מסקנות: בכשליש מן השאילתות המנוע לא החזיר מסמכים כלל. הבנו שיש צרוך בשיפור המנוע שיצליח להחזיר מסמכים לרוב השאילתות.

1. **ניסוי שני:** ביצענו ניסוי דומה לראשון, השוני היחיד הינו בשלב ה-pre-processing בו הוספנו שימוש בPorter stemmer. ניסינו בcosine similarity לחלק באורך המסמך אך קיבלנו ביצועים פחות טובים ולכן החלטנו להשאיר את החלוקה בשורש הריבועי של הווקטורים.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 3.312 | 0.121 | 0.187 | 0.170 | 0.131 | 0.311 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.114 | 0.027 | 0.048 | 0.109 |  |  |

תוצאות הניסוי: ממוצע המדדים

מסקנות: ביצוע הstemming תרם למציאת מסמכים רלוונטיים עבור השאילתות עליהן לא הצלחנו לענות ללא stemming. בנוסף ניתן לראות שיפור במדדים.

1. **ניסוי שלישי:** החלטנו להוסיף למנוע אינדקס על הגוף המסמך בנוסף לכותרת. את הדמיון בין המסמך לשאילתה חישבנו על ידי cosine similarity על פי הנוסחה המוכרת ללא שינויי. בחישוב הרלוונטיות של המסמך נתנו משקל שונה לכל השוואה.

30% נתנו לחישוב הדמיון מול כותרות המסמכים ושאר האחוזים לגוף המסמך.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 5.467 | 0.139 | 0.215 | 0.196 | 0.149 | 0.354 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.130 | 0.031 | 0.055 | 0.125 |  |  |

תוצאות הניסוי: ממוצע המדדים

מסקנות:המדדים השתפרו אך משך הזמן למענה על שאילתה גדל. בנוסף היו כמה timeouts על מספר קטן של שאילתות (4/30), בעיקר על השאילתות הארוכות.

1. **ניסוי רביעי:** הוספנו חישוב tf-idf על גוף הטקסט ובעצנו חישוב דמיון בעזרת זה במקום cosine similarity. השארנו את משקול של גוף הטקסט והכותרת

תוצאות הניסוי: ממוצע המדדים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 6.432 | 0.121 | 0.186 | 0.172 | 0.131 | 0.334 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.114 | 0.027 | 0.049 | 0.110 |  |  |

מסקנות: המדדים ירדו והזמן שאילתה גדל אבל המנוע הצליח להחזיר תשובה עבור כל השאילתות למעט שאלתה אחת.

1. **ניסוי חמישי:** שינינו את שטית ההשוואה של הטקסט שתהיה BM25. שיטת ההשוואה על הכותרות נשארה אותו דבר, cosine similarity. עדיין החזרנו 100 מסמכים עבור כל שאילתה. הפרמטרים שקבענו הם:

* [B=0.75 k1=2 k3=0] [B=0.75 k1=2 k3=1] בין שני ניסויים אלה לא השתנו התוצאות של המדדים.
* [B=0.75 k1=1.2 k3=1] בחירת פרמטרים זהו שיפר את המדדים אבל לא בצורה משמעותית.
* [B=0.5 k1=1.2 k3=1.2] פרמטרים אלה הורידו את רמת הביצועים.

תוצאות הניסוי: ממוצע המדדים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 5.349 | 0.110 | 0.157 | 0.146 | 0.111 | 0.208 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.097 | 0.022 | 0.041 | 0.093 |  |  |

מסקנות: תוצאות ניסוי זה לא היו טובות ולכן ניסינו לבצע אופטימיזציה בדרך אחרת.

1. **ניסוי שישי:** שמנו לב שעבור שאילתות קצרות יותר המנוע מתקשה להחזיר מסמכים, למשל עבור genetics לא קיבלנו מסמכים רלוונטיים כלל – כאשר עשינו stem קיבלנו genet שזה גחן, ועבור genetics לא חזרו מסמכים כלל. לכן ניסינו להוסיףquery expansion בעזרת word2vec עבור שאילתות קצרות יותר מ-2 מילים (בדקנו גם על שאילתות ארוכות, אך זה הגיע ל-timeout ולא היה שיפור משמעותי). בניסוי ביצענו את ההרחבה על כל השאילתות. את הדמיון על המסמכים עשינו בעזרת cosine similarity על הכותרות וגם על הטקסט, עם משקול של 30/70.

תוצאות הניסוי: ממוצע המדדים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | rq | precision@5 | precision@10 | precision@30 | average precision |
| 13.854 | 0.135 | 0.223 | 0.204 | 0.140 | 0.372 |
| f1@30 | recall@5 | recall@10 | recall@30 |  |  |
| 0.118 | 0.031 | 0.055 | 0.110 |  |  |

מסקנות: בעזרת שיטה זו הצלחתנו להחזיר מסמכים רלוונטיים על שאילתות שלא החזרנו עבורן בעבר, למשל genetics עלה מ-0 בכל המדדים ל-precision של 0.166.

1. **ניסוי שביעי**: ניסינו להוסיף Page Rank לציונים של המסמכים, אך זה הגדיל מאוד את זמן החישוב ולכן החלטנו להשמיט זאת.
2. גרף תיאור ביצועי המנוע (השוואת גרסאות)
3. גרף ממוצע זמן אחזור עבור שאילתה (השוואת גרסאות)
4. הערכת רלוונטיות עבור 10 המסמכים הראשונים שהוחזרו עבור שאילתה אחת שהמנוע עבדה בצורה טובה ועבור שאילתה שבה המנוע לא עבדה בצורה טובה.