הפקולטה להנדסה
המחלקה להנדסת מערכות מידע

פרויקט בניית מנוע – חלק 2

מגישים: **עידו סולומון ת"ז 308111160**
ליאור פרי ת"ז 203722814

1. **<u>הסבר מפורט על אופן פעולת המנוע:</u>**

- מחלקות שהוספנו (במסגרת חלק ב')
<u>DocumentRank</u> – אובייקט המייצג דירוג של מסמך במסגרת שאילתא מסוימת (מכיל את השדות הבאים: מזהה שאילתא, שם המסמך, דירוג המסמך בשאילתא (מס' סידורי של המסמך כשהמסמכים הרלבנטים לשאילתא מדורגים בסדר יורד ברלבנטיות שלהם) והניקוד שהמסמך קיבל בהתאם למנוע.
<u>PostingfileRecord</u> – אובייקט המייצג רשומה שנקראה מקבצי ה-posting. המטרה – הרשומות כתובות כמחרוזת ארוכה ולא נוח לעבוד איתן, לכן לאחר שרשומה נקראת אנחנו מייצגים אותה באמצעות האובייקט הזה על מנת שיהיה נוח לעבוד איתן.
האובייקט מכיל: מיפוי של כמה פעמים המונח הופיע בכל אחד מהמסמכים (בהם הופיע), מיפוי של איזה מילים הופיעו אחרי המונח בכל אחד מהמסמכים, וסיכום כולל של אילו מונחים הופיעו אחרי המונח הנכוחי בכל המאגר.
<u>PostingFilesManager</u> – בגלל שגם ה-Searcher וגם ה-Indexer נגשים לקבצי ה-posting לצרכים שונים וכדי למנוע כפילות קוד, יצרנו מחלקה שתרכז את כל הפעילות של המנוע אל מול קבצי ה-posting. מתפקידי המחלקה: ליצור קבצים ריקים בעת אינדוקס של מאגר מאפס, לכתוב רשומות לקבצי ה-posting ולעדכן אותן במידת הצורך, קריאה של רשומות מקבצי posting של מונחים מסויימים ולהחזיר PostingfileRecord עבור כל אחד מהמונחים.
<u>LanguageSelection</u> - אובייקט שמסייע ב-GUI לקבל מהמשתמש באילו שפות הוא מעוניין לחפש (מכיל את שם השפה, וערך בוליאני האם המשתמש מעוניין לכלול את השפה בחיפש)

- <u>הסבר מפורט של כל המחלקות הרלבנטיות</u>:
בהנתן שאילתא שקיבלנו מהמשתמש, יבוצעו הפעולות הבאות:
  - השאילתא תועבר ל- Parser , ונקבל חזרה כפלט את אוסף המונחים בשאילתא לאחר parsing , ואת מספר ההופעות שלהם בשאילתא.
  - השאילתא אחרי parsing עוברת ל-Searcher . ה-Searcher (באמצעות ה-PostringFilesManager) מחזיר את כל רשומות קבצי ה-posting של המונחים מהשאילתא, ויוצר רשימה של כל המסמכים הרלבנטיים – כלומר כל המסמכים שאחד המונחים מופיע בהם פעם אחת לפחות.
  - הרשומות ואוסף המסמכים מועברים ל-Ranker שמדרג את אוסף המסמכים שהתקבלו. ה-Ranker מפעיל עבור כל מסמך פונקציית דירוג, ממיין את המסמכים בסדר דירוג יורד, ומחזיר את 50 המסמכים הרלבנטיים ביותר עבור השאילתא (מיוצג באמצעות <u>DocumentRank</u>).

- השלמה אוטומטית מתבצעת בדרך הבאה: בזמן ה-Parsing , אנו שומרים עבור כל מונח את כל המונחים שהופיעו אחריו, ובאיזו שכיחות. כדי ליצור השלמה אוטומטית למונח מסויים, אם המונח קיים במילון הראשי: אנחנו שולפים את רשומת ה– posting שלו, ממיינים את כל המונחים שהופיעו אחרי המונח הנוכחי בסדר שכיוחויות יורד, ומחזירים את 5 המונחים שהופיעו הכי הרבה פעמים בכל המאגר הכי המונח הנוכחי.
הרציונל – על מנת ליצור השלמה אוטומטית רלבנטית ביחס למאגר הקיים, אנו מניחים כי ככל שצמד מונחים הופיעו יותר פעמים במאגר, כך השילוב ביניהם חשוב יותר ונכון במגוון רחב של הקשרים. רצינו להמנע מאלגוריתמי השלמה אוטומטית כללים (כאלו שאינם לוקחים בחשבון את המאגר) משום שלא ראינו ערך בהשלמה אוטומטית שתקבל מענה רדוד יחסית במנוע.

- פרמטרים חשובים להשלמה אוטומטית:
  - בגלל שתמכנו בהשלמה אוטומטית רק בשפה האנגלית, ולא בכל מסמך מצויינת השפה בה נכתב, לעתים ההשלמה אוטומטית מתבססת על שפה זרה ומתקבלת השלמה שאנה מועילה למשתמש.

- אוסף של מושגים (אישים, מאורעות, מונחים) שמוזכרים במאגר בכדי לשפר השלמה שלהם.

- אלגוריתם הדירוג:

אלגוריתם הדירוג פועל על פי נוסחת הדירוג הבאה:

$$Rank\ (Q,D) = BM25(\mathrm{Q},\mathrm{D}) + w_1 * TitleScore(Q,D)$$
$$+w_2 * SemanticScore(Q,D) + w_3 * AdjacentTermsScore(Q,D)$$
$$+w_4 * SpecificityScore(Q,D)$$

כאשר:

$$w_1 = 1.8\ , w_2 = 0.5, w_3 = 0.2, w_4 = 0.65$$

נפרט על מרכיבי החישוב:

- $\boldsymbol{BM25(\mathrm{Q},\mathrm{D})}$– כפי שנלמד בתרגולים, מניסויים אמפיריים המשלבים קומבינציות שונות של ערכים עבור הקבועים בנוסחה, נמצא כי $k_1 = 1.6$,
- $k_2 = 100, b = 0.25$ נותנים את הדירוג המיטבי.

- $\boldsymbol{TitleScore(Q,D)}$ – ניקוד הניתן למסמך בהתייחס למונחים מהשאילתא המופיעים בכותרת המסמך. מחושב באופן הבא:

$$TitleScore(Q,D) = \sum_{i \in Q} \frac{qf_i}{ql} * \frac{tf_i}{tl}$$

כאשר:

qf_i – מספר ההופעות של המונח ה-i בשאילתא.

ql - אורך השאילתא.

tf_i – מספר ההופעות של המונח ה-i בכותרת.

ql - אורך הכותרת.

- $\boldsymbol{SemanticScore(Q,D)}$ – באמצעות API שמוצא מילים בעלות משמעות סמנטית דומה, אנחנו יוצרים שאילתא Q' חדשה המורכבת מכל המילים המקוריות בשאילתא, וכל המילים שנמצאו כבעלות משמעות סמנטית דומה למילים המקוריות של השאילתא. את השכיחויות של המונחים בשאילתא החדשה אנחנו מגדירים באופן הבא:

אם i הוא מונח שהופיע בשאילתא המקורית, אז השכיחות שלו בשאילתא החדשה תהיה $q'f_i = qf_i$ (כלומר זהה לזו שהייתה בשאילתא המקורית.

אם i הוא מונח שהתווסף לשאילתא המקוית, אז השכיחות שלו בשאילתא החדשה תהיה $q'f_i = w_s * qf_i$ כשמצאנו ש $w_s = 0.9$ מביא לתוצאות מיטביות. (הפחתה של הערך של המונחים שאינם היו בשאילתא המקורית). מכאן החישוב מתבצע באופן הבא:

$$SemanticScore(Q,D)$$
$$= BM25(\mathrm{Q}',\mathrm{D}) + w_1 * TitleScore(\mathrm{Q}',D) + w_3$$
$$* DdjacentTermsScore(\mathrm{Q}',D) + w_4$$
$$* SpecificityScore(Q'.D)$$

(כלומר השאילתא החדשה מחושבת באופן זהה לשאילתא המקורית, מלבד הטיפול הסמנטי שלא מתבצע שוב.

- $\boldsymbol{AdjacentTermsScore(Q,D)}$ - ניקוד שניתן למסמך בהתבסס על כך ששני מונחים מהשאילתא הופיעו בסמיכות זה לזה במסמך. מחושב באופן הבא:
נסמן כ $adjacent(i1,i2,D)$ את מספר הפעמים שהופיע המונח i2 ישר לאחר מונח i1 במסמך D.

$$AdjacentTermsScore(Q,D) = \sum_{i_1 \in Q} \sum_{i_2 \in Q, i_2 \neq i_1} \frac{adjacent(i1,i2,D)}{qf_i}$$

כאשר:
qf_i – מספר ההופעות של המונח ה-i בשאילתא.

- $\boldsymbol{SpecificityScore(Q,D)}$ – ניקוד שניתן למסמך בהסתמך על ה"ספציפיות" שלו במונחי השאילתא (כלומר באיזה יחס עוסק בנושא השאילתא מבין כל הנושאים המופיעים בו, כאשר נושאים מיוצגים על ידי מונחים)

$$SpecificityScore(Q,D) = \frac{Unique\ (Q,D)}{ql} * \frac{D_{uniqe}}{dl}$$

$Unique\ (Q,D)$ – מספר המונחים הייחודיים המופיעים בשאילתא Q ומופיעים לפחות פעם אחת במסמך D .
ql - אורך השאילתא.
D_{uniqe} – מספר המונחים הייחודיים המופיעים במסמך D.
dl - אורך המסמך.

- האלגוריתם הסמנטי מופיע כחלק מחישוב פונקציית הדירוג .
- כל רשומת posting מכילה את הנתונים הבאים: עבור כל מסמך מופיע: שם המסמך, מספר ההופעות של המונח במסמך, ואת אוסף המונחים שהופיעו במסמך אחרי המונח הנוכחי, בתוספת השכיחות בה זה קרה (כלומר מספר הפעמים במסמך בו הופיע כל מונח אחרי המונח הנוכחי). הנתון של שכיחות המונח במסמך משמש כמרכיב לחישוב ברוב המרכיבים של פונקציית הדירוג, המונחים שהופיעו אחרי אחרי המונח הנוכחי משמשים הן להשלמה אוטומטית, והן מקבלים משקל בעת חישוב פונקציית הדירוג. במילון הראשי אנחנו מחזיקים עבור כל מונח: את המונח עצמו, את מספר המסמכים השונים בהם הוא מופיע, באיזה קובץ posting הוא מופיע ובאיזו שורה בתוכו. מספר המסמכים השונים בהם הוא מופיע משמש לחישוב BM25 , יתר הנתונים נועדו לשליפה של רשומת ה-posting .
- אלגוריתם הדירוג והנוסחה הסופית מופיעים לעיל. משקלי הדירוג חושבו בניסויים אמפיריים: כתבנו קוד שמזין משקולות לנוסחת הדירוג, מדרג את קובץ השאילות שקיבלנו בעבודה, מריץ את התוצאות שהתקבלו ב-treceveal ובודק מה הם מספר המסמכים הרלבנטיים שהוחזרו ומה היה הMAP. כך יכולנו לבחון מספר רב של קומבינציות שונות של משקולות, ולבחור את אלו שהביאו לתוצאות הטובות ביותר.
- השתמשנו ב- API בשם WordNet 3.0 שעבור כל מילה מסויימת, מחזיר קבוצות של מילים בעלות קשר סמנטי דומה למילה. השתמשנו בשירות זה על מנת להרחיב את השאילתא, ולהוסיף רכיב בפונקציית הדירוג המשתמש בטיפול סמנטי.

2. **<u>הערכה של המנוע:</u>**
<u>בלי Stemming</u>

p@30	p@15	p@5	R	P(=p@50)	סך המסמכים הרלבנטים לשאילתא	מס' מסמכים רלבנטים שאוחזרו	מילות השאילא	מספר השאילתא
0.3	0.1333	0.2	0.185	0.340	92	17	Space Program	11
0.9333	0.9	0.8	0.136	0.620	228	31	Water Pollution	12
0.6	0.8	0.8	0.382	0.420	55	21	Genetic Engineering	82
0.4667	0.5333	0.6	0.065	0.420	324	21	International Terrorists	118
0.9	0.9333	1	0.053	0.860	808	43	Impact of Government Regulated Grain Farming on International	142
0.4667	0.4667	0.4	0.038	0.440	584	22	Real Motives for Murder	189
0.2667	0.3333	0.8	0.381	0.160	21	8	Airport Security	341
0.8	0.7	0.6	0.520	0.780	75	39	Wildlife Extinction	347
0.2667	0.4	0.6	0.106	0.200	94	10	piracy	367
0.1	0.133	0.4	0.182	0.080	22	4	Nobel prize winners	374
0.1667	0.3333	0.8	0.167	0.200	60	10	oceanographic vessels	399
0.6	0.8667	1	0.947	0.360	19	18	Schengen agreement	410
0.4667	0.5333	0.2	0.643	0.360	28	18	Three Gorges Project	416
0.2	0.4	0.6	0.156	0.140	45	7	robotic technology	431
0.4333	0.3333	0.2	0.137	0.500	183	25	King Hussein, peace	450

<u>Map = 0.1853</u>

עם Stemming

מספר השאילתא	מילות השאילא	מס' מסמכים רלבנטים שאוחזרו	סך המסמכים הרלבנטים לשאילתא	P(=p@50)	R	p@5	p@15	p@30
11	Space Program	14	92	0.28	0.152	0.2	0.2	0.3
12	Water Pollution	29	228	0.58	0.127	1	0.7333	0.7333
82	Genetic Engineering	22	55	0.44	0.400	1	0.8	0.6333
118	International Terrorists	18	324	0.36	0.056	0.2	0.4	0.45
142	Impact of Government Regulated Grain Farming on International	47	808	0.94	0.058	1	0.9	0.8667
189	Real Motives for Murder	25	584	0.5	0.043	0.4	0.6	0.5667
341	Airport Security	10	21	0.2	0.476	0.8	0.3333	0.2667
347	Wildlife Extinction	34	75	0.68	0.453	0.6	0.7333	0.7667
367	piracy	10	94	0.2	0.106	0.6	0.4	0.2667
374	Nobel prize winners	5	22	0.1	0.227	0.2	0.1333	0.1
399	oceanographic vessels	11	60	0.22	0.183	0.8	0.6	0.3333
410	Schengen agreement	18	19	0.36	0.947	1	0.8667	0.6
416	Three Gorges Project	17	28	0.34	0.607	0.2	0.5333	0.4
431	robotic technology	31	45	0.62	0.689	1	1	0.8
450	King Hussein, peace	18	183	0.36	0.098	0.2	0.2667	0.3667

MAP = 0.2174

3. **סיכום**

האתגר הגדול ביותר בפרויקט לדעתנו: חשיבה על מרכיבים בפונקציית דירוג שישפרו את מדדי המנוע. מרגע שהשתמשנו ב– BM25 נדרשנו לחשוב על מרכיבים חדשים ויצירתיים כדי לשפר את מדדי המנוע, כאשר כל שיפור נהיה קשה יותר מקודמו. שילוב מדדים חדשים מצריך ניסוי וטעייה רב במהלכו לעתים האינטואיציה לגבי פונקציית דירוג חדשה היא נכונה ולעתים לא, ואיזון מחדש של המשקולות הקיימים לאחר שרכיב מוכיח את עצמו כאפקטיבי בניסויים על סט האימון שקיבלנו.

המלצות לשיפור:

- טיפול סמנטי מתקדם יותר – האלגוריתם הסמנטי שלנו הוא פשטני מדי (אנו מוסיפים לכל מונח את כל המילים בעלות משמעות סמנטית דומה לשאילתא, ונותנים ניקוד לשאילתא זו כחלק משקלול הדירוג הסופי). הבעיה בדרך זו, היא שמילים נרדפות רבות עלולות ליצור קשרים לוגיים חדשים לשאילתא, שלא היו קיימים בשאילתא המקורית. כאשר ניסינו להשתמש בפונקציה שמדרגת את המסמכים על פי משמעות סמנטית בדרך פחות נאיבית (למשל, לצרף בכל פעם לשאילתא מילים בעלות משמעות דומה רק עבור מילה אחת בשאילתא) הגענו לזמן ריצה ארוך מאוד (כמה דקות). קיים מקום להשתמש באלגוריתם סמנטי מתקדם

יותר (למשל, כזה שיכול לתת דירוג לקשר בין זוג מילים, ולציין אילו זוגות מילים נוספות עומדים באותה משמעות סמנטית כמו זוג אחר).

- <u>סיווג לקונספטים</u> – האלגוריתם שלנו לא משתמש בחלוקה לקונספטים (למשל ב-LSI) מאחר שבדרך המימוש המוכרת לנו לאלגוריתם זה נדרש זכרון רב וחישוב מסובך. שילוב של אלמנטים מאלגוריתם זה יכולים לשפר את ביצועי המנוע.
- <u>קביעת רף מינימום לניקוד של מסמך על מנת להחשב כרלבנטי עבור השאילתא:</u> האלגוריתם שלנו מחשיב מסמך כרלבנטי עבור שאילתא, אם הוא נמצא בין 50 המסמכים המדורגים כגבוהים ביותר עבור שאילתא, ואם הניקוד שקיבל הינו חיובי. לא פעם מצאנו שרף זה נמוך מדי, ולעתים מכניס גם מסמכים לא רלבנטים לחלק מתוצאות השאילתא. אולם לא הצלחנו למצוא דרך טובה יותר לקבוע רף מינימלי אחר על מנת להחשיב מסמך כרלבנטי לשאילתא.
- <u>פסילת מסמכים לא רלבנטים:</u> לא פעם הסתננו למקום גבוה בדירוג מסמכים לא רלבנטיים. על מנת לשפר את האלגוריתם, יש צורך לחשוב על דרכים להפחית את הניקוד של מסמכים לא רלבנטיים.

<u>בעיות שנתקלנו בהן</u>

<u>בעיות</u>	<u>דרך התמודדות</u>
הוספת מרכיבים חדשים לפונקציית הדירוג	ניסויים אמפירים מרובים במהלכם בדקנו השפעה של הוספת רכיב אחד חדש לפונקציית הדירוג במשקולות שונות, ובאם נמצא שהוא משפר את הביצועים, שולב בפונקציית הדירוג.
קביעת משוקלות לפונקציות הדירוג	ניסויים אמפירים מרובים במהלכם נבדקו קומבינציית שונות של משקולות, תוך כדי צמצום הטווחים הנבדקים בכל ניסוי, עד להגעה למשקולות המביאות למקסום ביצועי המנוע.
חשיבה על מרכיבים לפונקציית הדירוג	ניסינו לקרוא את המסמכים שהמנוע לא סיווג כרלבנטים, ובאמצעותם לחשוב על מדדים חדשים שיהיו רלבנטים יותר עבור מסמכים אלו.