אחזור מידע – פרויקט מנוע חיפוש

חלק ב'

דו"ח

מגישים:

טל פרימרמן 207012055

קורן ישלח 207964586

עיצוב התוכנה

1. חלק ב' של המנוע מורכב משני רכיבי ליבה עיקריים: **searcher** ו-**ranker**.

התהליך הכללי בחיפוש שאילתא הוא כדלקמן: פרסור השאילתא, שליחה שלה (ושל מילים דומות סמנטית למילותיה אם המשתמש בחר) לsearcher, אשר מוצא את כל המסמכים המכילים term אחד לפחות שקיים בשאילתא, ועבור כל מסמך כזה הוא שולח את פרטיו של המסמך ופרטי השאילתא למחלקת ranker, אשר נותנת תוצאה של רלוונטיות המסמך לשאילתא (עוד על פונציית הדירוג בחלק c). לאחר מכן הsearcher ממיין את המסמכים לפי התוצאות ומוצא את 50 המסמכים בעלי התוצאות הגבוהות ביותר עבור השאילתא.

המחלקות שמימשנו בחלק זה:

Searcher

public ArrayList<DocumentDataToView> search(String query, boolean withEntities){

הפונקציה המרכזית של המחלקה. היא מקבלת שאילתא ומשתנה בוליאני המורה על האם יש להחזיר גם ישויות עבור מסמכים. הפעולה מחפשת מסמכים המכילים מילים מהשאילתא, מחשבת לכל מסמך את הדירוג שלו ומחזירה לפי הסדר את 50 המזמכים הרלוונטיים ביותר.

private String makeEntitiesString(ArrayList<Term> entities) {

הפונקציה מחזירה את כל הישויות המתקבלות במחרוזת אחת.

private void getDocsData(ArrayList<Pair<Term, String>> termPostingData,  
 HashMap<String, DocRankData> hashChecker, int recognizer)

הפונקציה ממלאת את המידע באובייקט DocRankData לפי הביטויים המתקבלים והמידע עליהם בקבצי הפוסטינג, ועם המידע מהמסמכים מקבצי המסמכים.

private String[] findDocNoAndTf(String docNoTfCurrent) {

הפונקציה מחזירה מערך המכיל את מזהה המסמך והTF הרלוונטי בהתאם למחרוזת המתקבלת.

private void initializeDocNecessaryData(DocRankData currentDocData, String[] splitter) {

אם זו הפעם הראשונה שאנו נתקלים במסמך, הפונקציה יוצרת מופע של האובייקט DocNecessaryData שמכיל נתונים על המסמך (כותרת, תאריך ומסמך מילים).

private ArrayList<Object> findDf(String termPostingData) {

הפונקציה מייצאת את הtf מתוך השורה של הביטוי בקובץ הפוסטינג.