**README**

בתרגיל זה מימשנו מנוע חיפוש המורכב משלושה חלקים: crawler , inverted index , queries.

ה- **crawler** יוצר את אוסף המסמכים (הקורפוס), אשר בהם מתבצע החיפוש. הוא ניגש ל- seedfile אשר קיבל כקלט מהמשתמש ומתחיל לאסוף את כל הלינקים המצויים בו. יצירת הקורפוס מתבצעת לפי העומק שהתקבל כקלט או הערך ברירת המחדל שהגדרנו (2), ופועל כך שאם הוא נתקל בדף (לינק) שכבר הכניס לקורפוס הוא לא יכניס אותו שוב אך ימשיך לחפש דפים נוספים. הכנסנו לקורפוס רק את הדפים אשר בתוך wikitravel.org/en (מתחילים ב- <http://wikitravel.org/en>). בנוסף, השתמשנו בפונקציה parseURL, כדי למנוע מלינקים שהם תמונות/ קבצים/ לינקים שמובילים לתוך העמוד עצמו להיכנס לקורפוס.

ה- **Inverted Index** יוצר את הקובץ InvIndex.csv, אשר מכיל את האינדקס המהופך. בחלק זה השתמשנו בתחילה ב- tokenizer, הורדנו תווים שהם לא באנגלית (כולל מספרים), השתמשנו ב- nltk.stemmer על מנת להוריד סופיות של מילים (כמו ing, s, ly וכו'...). בנוסף, ע"י הפונקציות bigrams ו- trigrams יצרנו בכל קובץ זוגות ושלשות של מילים והכנסנו גם אותם לאינדקס המהופך.

ה- **Queries** מקבל מהמשתמש כקלט את השאילתא לחיפוש, מבצע את החיפוש בתוך הקורפוס שה- crawler יצר ולבסוף מחזיר בדף html עם קישורים את תוצאות החיפוש הרלוונטיות ממוינות (מהרלוונטית ביותר לרלוונטית פחות) .

הערות:

* הקורפוס – אוסף הקבצים שאנו מצרפים לתרגיל זה הוא הרצה של ה- myCrawler , על seed : <http://www.wikitravel.org/en/Ireland> , בעומק 1.
* כפי שדיברנו איתך בהצגת התרגיל, אין התייחסות לקובץ robots.txt. ההתייחסות שלנו לבקשת הנימוס של הפסקת הורדת דפים מה Server לאחר כל דף מתבצע בתרגיל באופן הבא: לאחר הבאת עמוד ה - HTML מה - Server מתבצע ניקוי של העמוד מתגיות, לאחר מכן כתיבה של הקובץ לדיסק ורק לאחר מכן מתבצעת הקריאה לעמוד הבא. לכן זמן ההפסקה בין קריאה לקריאה לעמוד היא זמן העיבוד של הקבצים וכתיבתם על הדיסק הקשיח.
* קבצי pickle : dicVec.pkl – ייצוג של hash table של וקטורי המסמכים:

Filename -> [term: w = tf\*idf/norma]

במבנה זה אנו משתמשים עם myQueries כדי לבצע קוסינוס עם וקטור השאילתא.

idfDic.pkl מכיל עבור כל מילה באוסף המילים את ערך ה idf שלה. גם בייצוג של Dictionary. גם לשימוש ב myQueries .

אלו נוצרים במודול של myI שיוצר את האינדקס המהופך. אלו נקראים ממודול myQueries.

* ידוע לנו כי כתיבת הקבצים הנ"ל וקריאתם גוזלת זכרון רב עבור קורפוס גדול. זאת היות ואנו רוצים לבצע חיפוש מהיר ולכן אלו יושבים בזכרון ה RAM של המחשב.
* מימוש bigrams ו trigrams: כאשר אנו מעבדים קובץ TokenFile אנו מצרפים לאינדקס המהופך גם זוגות ושלשות של מילים. כאשר אנו מבצעים שאילתא אנו יוצרים גם כן זוגות ושלשות ( אם קיימות ) ממילות השאילתא – יוצרים וקטורים אחרים בנוסף לוקטור הרגיל, וגם עליהם אנו מבצעים קוסינוס עם הוקטורים של שאר המסמכים.

התוצאות בדף ה HTML מחולקות לפי : תוצאות לשלשלה, לאחר מכן תוצאות לזוג מילים ולבסוף תוצאות רגילות.

גודל הK תוצאות הטובות ביותר מוגדר ב myQueries.

חשוב מאוד: כדי להפעיל את myQueries , myI , יש להכניס בשורת הפקודה (command (prompt את ה outputdir שהכנסנו תחילה ב myCrawler.

יש להריץ תחילה את myI ורק לאחר מכן את myQueries כדי לבצע חיפוש, כי קבצי ה pickle נוצרים ב myI ונקראים ב myQueries.