ניהול נתונים באינטרנט 2021

פרויקט תכנות – Information Retrieval

הוראות כלליות

יש לעלות את הפתרונות בקובץ ZIP לMOODLE, שכולל קובץ PDF בשם answers.pdf ובו הפתרון וקבצי קוד נוספים (HTML, XML או Python) לפי הדרישה של כל סעיף וסעיף. ההגשה היא בזוגות, כך שרק אחד מבני הזוג יגיש את התרגיל, אך יש להקפיד לכתוב את השמות והת.ז. של שני בני הזוג בתוך הקובץ.
שם של הקובץ ZIP צריך לכלול את הת.ז. של שני המגישים, למשל: HW_IR_123_456.zip

תאריך פרסום: 27/04/2021 **תאריך הגשה: 22/07/2021**

רקע

זהו פרויקט תכנות בנושא אחזור מידע (Information Retrieval) בו תממשו מנוע חיפוש מבוסס Vector Space Model. עליכם לבנות מערכת אוטומטית שבהנתן שאלה בשפה טבעית ומאגר מסמכים, תחזיר למשתמש את אוסף המסמכים הרלוונטים ביותר לשאלה.
הפרויקט להגשה עד לתאריך שמצוין למעלה ומהווה 11% מהציון הסופי בקורס.

תיאור המערכת

התכנית שתכתבו תבנה מנוע חיפוש בשני שלבים: בשלב הראשון, התוכנית תבנה Inverted Index מתוך מאגר של מסמכי XML. בשלב השני, התכנית תקבל שאלה מהמשתמש ובעזרת האינדקס (שבנתה offline) תדרג ותחזיר בזמן אמת למשתמש את המסמכים במאגר שרלוונטים לשאלה. למשל, אם יש לנו מאגר של מאמרים אקדמיים בנושאי רפואה, על התוכנית להתשתמש ב-Inverted Index כדי להחזיר מאמרים רלוונטים לשאלות כגון:

- Is salt (sodium and/or chloride) transport/permeability abnormal in CF?
- What abnormalities of insulin secretion or insulin metabolism occur in CF patients?
- Can CF be diagnosed prenatally?
- …

שלב א': בניית Inverted Index

התכנית תקבל כקלט נתיב לתיקייה המכילה מאגר מסמכי XML שישמשו לבניית ה-inverted index. כל מסמך במאגר הוא אובייקט XML ייחודי, ראו בהמשך.
תכנית המחשב תעבור על כל המסמכים במאגר ותבנה מתוכם את האינדקס כפי שלמדנו בהרצאה.

1. בניית המילון לאינדקס
   a. מעבר על המילים במסמך, למשל בכותרת ובסיכום שלו (extract)
   b. ביצוע tokenization למילים במסמך
   c. הסרת מילים שכיחות stopwords, ניתן למצוא מאגר stopwords בחיפוש אונליין
   d. ביצוע stemming בעזרת Porter Stemmer למילים שאינן stopwords (שלב זה אינו חובה)
2. חישוב ציוני TF-IDF לכל מילה במילון עבור כל מסמך שבו היא מופיעה

שלב ב': אחזור מידע בהנתן שאלה

התכנית תקבל כקלט את ה-inverted index שבנינו ושאלה בשפה טבעית מהמשתמש. בהנתן השאלה, המערכת תחזיר את רשימת המסמכים שרלוונטים לשאלה (אם ישנם כאלה). נחזיר את המסמכים מדורגים לפי ציון הרלוונטיות שלהם. לחישוב הדירוג נשתמש ב-Vector Space Model כפי שלמדנו בהרצאה.

1. נחשב בצורה אינקרמנטלית את ציון ה-cosine similarity של כל מסמך בהנתן המילים בשאלה:
   a. נעבור על כל מלה רלוונטית בשאלה (או על ה-stem שלה אם בחרנו להשתמש ב-stemming)
   b. נמצא כל מסמך שמכיל מילה זו
   c. נחשב את ציון המסמך בעזרת נתוני ה-TF-IDF הרלוונטים שנמצאים ב-index
2. נחזיר את רשימת המסמכים הרלוונטים מדורגים לפי cosine similarity

מאגר המידע לפרויקט: Cystic Fibrosis Database

מאגר המידע שלנו הוא Cystic Fibrosis Database, אשר מכיל 1,239 מסמכים ממאמרים שפורסמו בין השנים 1974 עד 1979 על מחלת הסיסטיק פיברוזיס. בנוסף, המאגר מכיל 100 שאלות באנגלית ולכל שאלה מצורפים שמות המסמכים הרלוונטים אליה.

המאגר כולו זמין ב- Moodle כקובץ tar בגודל 1.54Mb הכולל את המאגר והשאלות כולן כקבצי XML. מסמכי ה-Document Type Definition (DTD) של קבצי ה- XML כלולים גם הם במאגר.

קבצי המסמכים:
ישנם 6 קבצים cf74.xml, …, cf79.xml בהם כל המסמכים הרלוונטים לבניית האינדקס. בכל קובץ XML מסמך (מאמר) מיוצג כאובקייט XML בשם RECORD ובו המידע על כל מסמך. פירוט על כל אובייקט ניתן למצוא בקבצי ה-DTD. האובייקטים הרלוונטים הם:

- RECORDNUM מזהה המסמך בו נעשה שימוש גם בקובץ השאלות
- TITLE כותרת המסמך
- EXTRACT חלק מהמסמך
- ABSTRACT סיכום קצר של המסמך

ניתן, אך לא חובה, לעשות שימוש באובייקטים נוספים לצורך שיפור תוצאות החיפוש לבחירתכם (למשל ציטוטי מאמרים). במידה ועשיתם שימוש באובייקטים נוספים, הוסיפו הסבר קצר בקובץ ה-answers.pdf שאתם מגישים.

קובץ השאלות:
קובץ השאלות cfquery.xml מכיל 99 שאלות בשפה טבעית בנושאי סיסטיק פיברוזיס. כל שאלה בקובץ נמצאת באובייקט XML בשם QUERY. האובייקט QueryText מכיל את תוכן השאלה. האובייקט Results מכיל את מספר המסמכים הרלוונטים לשאלה. האובייקט Records מכיל את כל מזהי המסמכים הרלוונטים לשאלה (בצירוף ציון של 4 מדרגים). שימוש לב ל"הערה על בדיקת הפרויקט" בנוגע לשימוש נאות בקובץ השאלות.

זכויות יוצרים:
כל זכויות היוצרים על המאגר שמורות למחברים מ- School of Information and Library Science, University of North Carolina, Chapel Hill, NC 27599-3360, USA. המאמר הרלוונטי בו פורסם המאגר הוא:

- Shaw, W.M. & Wood, J.B. & Wood, R.E. & Tibbo, H.R. The Cystic Fibrosis Database: Content and Research Opportunities. LISR 13, pp. 347-366, 1991

מסמך דרישות והרצת הקוד

- על הקוד להיות כתוב באמצעות Python 3 בלבד
- על קובץ ה-zip להכיל מסמך דרישות בשם requirements.txt ובו כל הספריות החיצוניות שנחוצות לשם הרצת הקוד, כל חבילה בשורה נפרדת. לדוגמה:

```
numpy
nltk
lxml
json
…
```

- פרויקט שלא יכיל מסמך דרישות ולא יסיים לרוץ ללא שגיאות <u>לא יקבל ציון עובר</u>

- הפרויקט יכיל קובץ פייתון בשם vsm_ir.py אשר ממנו תתבצע הרצת הקוד
  - קובץ ההגשה יכול לכלול קבצי פייתון נוספים אך יש לתעד את השימוש של כל קובץ במסמך answers.pdf שמצורף להגשה
- כדי ליצור את ה-inverted index התוכנית תרוץ משורת הפקודה באופן הבא:

```
python vsm_ir.py create_index [corpus_directory]
```

  - create_index משתנה שמצביע על יצירת ה-inverted index
  - corpus_directory הוא ה-path לתיקייה בה נמצאים קבצי ה-XML של מאגר המידע
  - לדוגמה:

```
python vsm_ir.py create_index /my_dir/data/cfc-xml
```

  - התוכנית תשמור את inverted index לדיסק תחת השם vsm_inverted_index.json
    - הפורמט של הקובץ נתון לבחירתכם
    - הריצו את התכנית על המאגר וצרפו את הקובץ vsm_inverted_index.json להגשה
- כדי להחזיר את המסמכים הרלוונטים לשאלה התכנית תרוץ באופן הבא:

```
python vsm_ir.py query [index_path] "<question>"
```

  - index_path הוא ה-path לקובץ ה-inverted index שיצרה התכנית
  - התוכנית תקבל כקלט שאלה באנגלית מהמשתמש, תטען את ה-inverted index מהדיסק ותחזיר את כל שמות המסמכים הרלוונטים לשאלה מדורגים בסדר יורד
    - התוכנית תשמור את התוצאות בקובץ ranked_query_docs.txt
    - המסמך יכול להיות ריק במידה ואף מסמך אינו רלוונטי לשאילתא
    - קובץ תוצאות לדוגמה:

```
494
536
1143
78
```

    - בדוגמה חזרו 4 מסמכים רלוונטים לשאלה. המסמכים מדורגים כך שמסמך 494 הוא הכי רלוונטי ו-78 הכי פחות
- הרצת הקוד תתבצע משורת הפקודה בלבד
- על התכנית להסתיים לאחר הרצת הפקודה (create_index או query), אין להשאיר את התכנית רצה

הוראות הגשה

יש להגיש קובץ ZIP יחיד בשם HW_IR_[id1]_[id2].zip עם מספרי תעודת הזהות של שני המגישים. על קובץ הזיפ לכלול את המסמכים הבאים:

- answers.pdf ובו תיאור כללי של הפרויקט. יש לכתוב את תעודות הזהות של המגישים בראש המסמך
- vsm_ir.py קובץ פייתון שמריץ את התכנית
- requirements.txt קובץ ובו כל ספריות הפייתון הדרושות לשם הרצת התכנית
- vsm_inverted_index.json קובץ ה-inverted index שיצרה התכנית כאשר הרצתם אותה על המאגר
- כל קובץ פייתון נוסף שבחרתם להוסיף לפרויקט. יש לכלול הסבר קצר בקובץ answers.pdf
- נא לתעד את קבצי הקוד בצורה סבירה

בדיקת הפרויקט

הפרויקט יבדק באופן אוטומטי על 30 שאלות באנגלית. לכל שאלה נחשב ציוני Precision, Recall, F, עבור המסמכים שהוחזרו ודירוגם לעומת תוצאות האמת.
לפני ההגשה מומלץ מאוד להריץ את המערכת שכתבתם על מאגר Cystic Fibrosis Database ועל כמה מהשאלות שבו. מאחר ודירוגי המסמכים ביחס לכל שאלה נתונים, אנו ממליצים לכתוב פונקציות שערוך לביצועי המערכת שלכם. ודאו שהמערכת שלכם מחזירה מסמכים הגיוניים בהינתן השאלה ושהמסמכים אכן מדורגים.
תזכורת לפונקציות השערוך:

- $Precision = \frac{|\{\text{retrieved documents}\} \cap \{\text{relevant documents}\}|}{|\{\text{retrieved documents}\}|}$

- $Recall = \frac{|\{\text{retrieved documents}\} \cap \{\text{relevant documents}\}|}{|\{\text{relevant documents}\}|}$
- $F = \frac{2 \cdot Precision \cdot Recall}{(Precision + Reca\ \ )}$

קלטי הרצה לדוגמה:

- `python vsm_ir.py query /mydir/vsm_inverted_index.json "What factors are responsible for the appearance of mucoid strains of Pseudomonas aeruginosa in CF patients?"`
- `python vsm_ir.py query /mydir/vsm_inverted_index.json "What is the prognosis for survival of patients with CF?"`
- `python vsm_ir.py query /mydir/vsm_inverted_index.json "Are there abnormalities of taste in CF patients?"`

<u>הערה בנוגע לבדיקה:</u>

הקובץ cfquery.xml מכיל 99 שאלות בצירוף שמות המסמכים הרלוונטים לכל שאלה ודירוגם.

כפי שצוין, ניתן להשתמש בשאלות ובדירוגי המסמכים לצורך שערוך הביצועים ה-Vector Space Model שכתבתם. אסור לעשות כל שימוש בדירוגי המסמכים לצורך שאינו performance evaluation. למשל, קוד שלא יממש VSM או קוד שיבצע hardcoding משאלות למסמכים בהתבסס על דירוגי המאגר <u>יקבל ציון 0</u>.

בהצלחה!