מגיש 1: אופק ברנסקי
ת.ז. 311170138
מגיש 2: מתן טולדו
ת.ז. 314023409

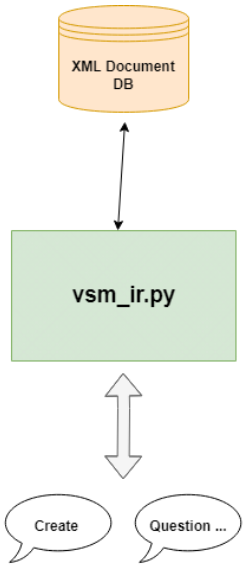
03683062 ניהול נתונים באינטרנט
פרויקט תכנות 2

1. <u>תיאור קוד הפרויקט:</u>

הקוד שלנו מורכב מקובץ python ומאגר קבצי XML (ה Corpus)

vsm_ir.py- מודול המקבל את הקלט מהמשתמש, מבצע pre-computation של ה inverted index, ומוצא את הקבצים הדומים ביותר לשאילתא, במודל bm25 ובמודל tf-idf.

Tester.py- טסטר שבנינו לפרויקט, מפורט עליו בהמשך.



כעת נרחיב על מודול:

א) המודול מקבל את קלט משתמש:

a. במידה והקלט שאילתא, המודול מבצע את הפעולות הבאות:
   i. מפרסר את סוג המודל המתבקש – **bm25** או **tfidf**
   ii. מחשב ע"פ הנלמד בשיעור את הציון **bm25** או **tfidf** של השאילתא עם כל המסמכים שבוצע אליהם preprocessing.
   iii. מחזיר את ה Top k מסמכים הדומים ביותר (כאשר כמות המסמכים שמוחזרים configurable בתוך הקוד.

b. במידה והקלט create, הוא מבצע את חישוב ה inverted index באופן הבא:
   i. מבצעים איטרציה על כל המסמכים (עבור כל מסמך, שולפים בעזרת xpath את ה Title, Extract ו Abstract שלו.

ii. עבור כל מסמך אנו מוחקים את כל הstop words (בעזרת החבילה nltk), מבצעים stemming בעזרת nltk.PorterStemmer וכמובן מוחקים סימני פיסוק ותווים שאינם אותיות באנגלית.

iii. לאחר "טיפול" בטקסט, אנו מבצעים tokenize של המילים שנותרו לנו, ומחשבים את הinverted index כפי שהוצג בכיתה.

iv. לאחר מכן מבצעים חישוב של ה idf scores, וה document vector norms

v. לסיום, אנו שומרים את התוצאה בקובץ json

ב) הקוד מוחלק לאובייקט המייצג את ה Vector Space Model :

```
class VSM:
    """Vector Space model class, holds corpus and index data"""
...
    RESULTS_THRESHOLD = 12
    BM_25_K = 1.4
    BM_25_b = 0.75

    def __init__(self):...

    """adds document data to index"""
    def add_doc_data_to_index(self, data_str, doc_num):...

    """ Iterate on every document, pre-process and build inverted index
    Return inverted_index and n = number of documents"""
    def build_inverted_index(self, corpus_directory):...

    """ Compute IDF for every token, and add to the index"""
    def add_idf_scores_to_index(self):...

    """Compute document vector norm for every document"""
    def compute_document_vector_norms(self):...

    """Saves indexes and lengths to JSON file"""
    def save_index_and_lengths(self):...

    """Loads indexes and lengths from JSON file"""
    def load_index_and_lengths(self, index_path):...
```

ג) והכנסנו עוד מספר פונקציות עזר לטובת שמירה וחישובים של ערכים שונים:

```
"""Get a string representing document/query
Tokenize the string using nltk, and convert to lowercase
Remove unwanted words (punctuation, numbers, stopwords) and do stemming using Porter Stemmer
Return the list of valid tokens"""
def extract_tokens(data_str):...


"""Compute term frequencies for a list of tokens (document/query) saved in curr_tokens
Normalize by maximum count"""
def compute_term_frequencies(data_str):...


"""Saves top K documents to file"""
def save_top_docs(top_docs):...


"""Modifies and calculates BM25"""
def modify_tf_bm25(tf, d, avgdl):...
```

2. **<u>ניתוח התוצאות:</u>**

ע״מ לנתח את תוצאות המודל, ולמקסם את תוצאותיו, ביצענו מספר דברים:

א) ראשית יצרנו קובץ test המריץ את כל השאילתות שניתנו (99 כאלו) על המודל שלנו,
עבור כל שאילתא, חישבנו את כלל הציונים האפשריים – NDCG, Precision, Recall, F value .
כאשר הפלט הסופי של הטסטר היה ממוצע הציונים של המודל על 99 השאילתות הרלוונטיות.

ב) ע״מ למקסם, בדקנו את השפעת הפרמטרים הבאים על התוצאות, ובחרנו בפרמטרים שמקסמו את הציונים עבור המודל:

a. **Bm25 <u>k</u> value-** ההיפר-פרמטר שנלמד בשיעור
b. **Bm25 <u>b</u> value-** ההיפר-פרמטר שנלמד בשיעור
c. **Document threshold-** כמות המסמכים המקסימלי שיוחזר עבור שאילתא מסוימת,

היה ניכר כי ככל שהעלינו את כמות המסמכים שהוחזרו, ה Recall גדל אך ה Precision קטן:

Document threshold – TF-IDF and BM25:

```
tfidf

Threshold = 2
Average NDCG@2 = 0.538
Average Precision = 0.692
Average Recall = 0.067
Average F = 0.113

Threshold = 4
Average NDCG@4 = 0.507
Average Precision = 0.609
Average Recall = 0.105
Average F = 0.161

Threshold = 6
Average NDCG@6 = 0.490
Average Precision = 0.559
Average Recall = 0.134
Average F = 0.191

Threshold = 8
Average NDCG@8 = 0.484
Average Precision = 0.520
Average Recall = 0.157
Average F = 0.211

Threshold = 10
Average NDCG@10 = 0.478
Average Precision = 0.494
Average Recall = 0.180
Average F = 0.229
```

```
bm25

Threshold = 2
Average NDCG@2 = 0.441
Average Precision = 0.571
Average Recall = 0.057
Average F = 0.095

Threshold = 4
Average NDCG@4 = 0.423
Average Precision = 0.510
Average Recall = 0.089
Average F = 0.134

Threshold = 6
Average NDCG@6 = 0.413
Average Precision = 0.468
Average Recall = 0.113
Average F = 0.160

Threshold = 8
Average NDCG@8 = 0.400
Average Precision = 0.428
Average Recall = 0.129
Average F = 0.172

Threshold = 10
Average NDCG@10 = 0.398
Average Precision = 0.404
Average Recall = 0.146
Average F = 0.186
```

BM25 – k value, b value:

```
bm25

k = 1.2
Average NDCG@7 = 0.406
Average Precision = 0.450
Average Recall = 0.122
Average F = 0.168

k = 1.3
Average NDCG@7 = 0.404
Average Precision = 0.443
Average Recall = 0.122
Average F = 0.166

k = 1.4
Average NDCG@7 = 0.400
Average Precision = 0.437
Average Recall = 0.119
Average F = 0.164

k = 1.5
Average NDCG@7 = 0.398
Average Precision = 0.431
Average Recall = 0.119
Average F = 0.162

k = 1.6
Average NDCG@7 = 0.395
Average Precision = 0.431
Average Recall = 0.119
Average F = 0.162

k = 1.7
Average NDCG@7 = 0.395
Average Precision = 0.433
Average Recall = 0.119
Average F = 0.162
```

```
bm25

b = 0
Average NDCG@7 = 0.443
Average Precision = 0.489
Average Recall = 0.133
Average F = 0.183

b = 0.1
Average NDCG@7 = 0.441
Average Precision = 0.486
Average Recall = 0.133
Average F = 0.183

b = 0.2
Average NDCG@7 = 0.433
Average Precision = 0.478
Average Recall = 0.131
Average F = 0.180

b = 0.3
Average NDCG@7 = 0.435
Average Precision = 0.483
Average Recall = 0.131
Average F = 0.181

b = 0.4
Average NDCG@7 = 0.428
Average Precision = 0.472
Average Recall = 0.129
Average F = 0.178

b = 0.5
Average NDCG@7 = 0.420
Average Precision = 0.462
Average Recall = 0.128
Average F = 0.176
```

```
b = 0.6
Average NDCG@7 = 0.413
Average Precision = 0.452
Average Recall = 0.124
Average F = 0.170

b = 0.7
Average NDCG@7 = 0.408
Average Precision = 0.450
Average Recall = 0.123
Average F = 0.169

b = 0.8
Average NDCG@7 = 0.401
Average Precision = 0.442
Average Recall = 0.121
Average F = 0.165

b = 0.9
Average NDCG@7 = 0.389
Average Precision = 0.433
Average Recall = 0.119
Average F = 0.162

b = 1
Average NDCG@7 = 0.383
Average Precision = 0.420
Average Recall = 0.114
Average F = 0.156
```

3. **<u>סיכום:</u>**

לאחר ניתוח של הפרמטרים וביצועי התוכנית, הגענו למסקנה כי הנתונים הטובים ביותר הינם:

Bm25_K = 1.2

זהו הערך שנותן ציון NDCG גבוה ביותר.

Bm25_b = 0

זהו הערך שנותן ציון NDCG גבוה ביותר.

Document_threshold = 7

זה ערך שנותן ציון NDCG גבוה, אבל גם מבטיח כיסוי מסמכים מסוים שנותן גם ציון F גבוה.

אלו המדדים הסופיים עם הפרמטרים האלו על פני 99 שאילתות:

```
tfidf
Average NDCG@7 = 0.488
Average Precision = 0.541
Average Recall = 0.145
Average F = 0.201
bm25
Average NDCG@7 = 0.443
Average Precision = 0.489
Average Recall = 0.133
Average F = 0.183
```