שאילתות באינטרנט

תרגיל 1 – מבנה האינדקס

1. ערכת הנתונים

מטרתו של הפרויקט שייבנה לאורך הסמסטר, הוא לבנות מנוע חיפוש עבור חוות דעת על מוצרים. לצורך כך נשתמש בנתונים הקיימים באוספים כגון "web-Fine Foods", "web-Movies" וכדומה ב Stanford Large Network Dataset Collection.

http://snap.stanford.edu/data/index.html#reviews

בנוסף, נשתמש גם בנתוני קלט אחרים שיהיו באותו מבנה.

כל אחת מחוות הדעת על המוצרים היא במבנה הבא:

product/productId: B001E4KFG0
review/userId: A3SGXH7AUHU8GW
review/profileName: delmartian
review/helpfulness: 1/1
review/score: 5.0
review/time: 1303862400
review/summary: Good Quality Dog Food
review/text: I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.

שימו לב כי זהו מידע אמתי, וככזה, עלול להכיל תוכן משונה. למשל ייתכן כי יימצא תו של newline באמצע profileName, או מילים מאוד ארוכות.

מכיוון שלחוות הדעת אין מזהה (ID), נמספר את חוות הדעת בסדר עולה. כלומר חוות הדעת הראשונה תקבל את המזהה 1, חוות הדעת השנייה תקבל את המזהה 2 וכן הלאה.

מתוך מגוון הנתונים שיש לכל אחת מחוות הדעת, נהיה מעוניינים בשמירה של השדות הבאים:

- product/productId
- review/helpfulness (two integers)
- review/score (integer between 1 and 5)
- review/text

לצורך שמירת הטקסט של חוות הדעת יש לבצע את הפעולות הבאות :

- חלוקת הטקסט למילים נפרדות בכל מקום שבו יש תו שאינו אלפאנומרי (אינו אות או ספרה). התווים שאינם אלפאנומריים צריכים להיות מושלכים.
- נרמול הטקסט על ידי הפיכת כל תווי האותיות לאותיות קטנות (lowercase)

ניתן להוריד ערכת נתונים ישירות מהקישור שלמעלה. כדי להקל עליכם את תחילת העבודה, ניתן להוריד מאתר הקורס שתי ערכות נתונים קטנות (אחת עם 100 חוות דעת והשנייה עם 1000 חוות דעת).

2. תיאור התרגיל

בהינתן קובץ הקלט עם הנתונים הגולמיים, עליכם ליצור אינדקס שיאפשר גישה יעילה למידע. על האינדקס להיות מאוחסן על הדיסק כדי שיהיה אפשר להשתמש בו כאשר מבצעים שאילתות על מוצרים שונים. לכן, קובצי האינדקס צריכים להישאר על הדיסק גם כאשר התכנית שלכם אינה רצה.

המבנה המדויק של האינדקס מהווה חלק מהחלטות התכנון שתקבלו. מימוש אינדקס עבור נתונים טקסטואליים יידון בהרחבה במסגרת השבועות הראשונים של הקורס. התפקיד שלכם הוא להשתמש ברעיונות שיילמדו (או להציע רעיונות אחרים משלכם) כדי לבנות את האינדקס שישרת את דרישות הפרויקט.

עליכם להיות מסוגלים לנתח את הגודל הצפוי של האינדקס שבניתם עבור גדלים שונים של קובצי קלט גולמיים (ניתוח של גודל הנתונים ייעשה במסגרת הקורס). עליכם להשתמש בכמה סוגים של טכניקות דחיסה כדי להבטיח שהאינדקס יהיה בגודל סביר. וודאו כי אתם מרכזים את רוב מאמצי הדחיסה שלכם על הנתונים שצפויים לגדול מאוד ככל שהקלט הגולמי גדל.

להלן כמה מגבלות על המימוש :

- אסור להשתמש במערכת של מסד נתונים כדי לשמור את המידע. עליכם לממש את האחסון בעצמכם.
- ניתן להשתמש ביותר מקובץ אחד כדי לאחסן את האינדקס. אולם, מספר הקבצים שייווצרו צריך להיות קבוע ולא תלוי במספר חוות הדעת או גודל המילון וכדומה.

בשלב זה אין צורך לדאוג לכך שתהליך בניית האינדקס יהיה יעיל (זו מטרתו של התרגיל השני) . עם זאת, מבנה האינדקס והמימוש צריך להיות ניתן לשדרוג כך שיוכל לאפשר בנייה ואחסון של כמות נתונים ענקית כך שתוכלו להשתמש בו כמו שהוא כאשר נרצה לבנות אינדקס עבור כמות גדולה יותר של נתונים.

3. דרישות הקוד

התכנית תכיל לפחות את שתי המחלקות הבאות: (ככל הנראה התכנית תכלול מחלקות רבות נוספות הנחוצות לצורך מימושן)

3.1. SlowIndexWriter: בהינתן נתונים גולמיים, המחלקה תיצור אינדקס על הדיסק שאפשר יהיה לגשת אליו מאוחר יותר. כל הנתונים שישתמשו בהם מאוחר יותר צריכים להיות מאוחסנים באינדקס שעל הדיסק.

המילה "slow" בשם המחלקה מעידה על כך שהתכנית שבונה את האינדקס יכולה לעבוד בצורה לא יעילה בשלב זה. בתרגיל זה ניתן להניח כי כאשר בונים את האינדקס, כל הנתונים יכולים להיות מאוחסנים בזיכרון (כמובן שהם צריכים להיות מאוחסנים על הדיסק בצורה חסכנית).

המחלקה מאפשרת גם למחוק את האינדקס מהדיסק על ידי מחיקת כל הקבצים מהספרייה של האינדקס.

3.2. IndexReader: לאחר שנוצר אינדקס על הדיסק ניתן להשתמש במחלקה כדי לגשת למגוון רב של נתונים הקיימים באינדקס. השתמשו במתודות אלו כהכוונה והדרכה לתכנון מבנה האינדקס. כלומר מבנה האינדקס צריך לתמוך במימוש יעיל של מתודות אלו. ניתן להניח כי מתודות אלו יופעלו רק לאחר שהאינדקס ייבנה על ידי SlowIndexWriter. המימוש צריך להיות באופן שהוא יהיה יעיל גם כאשר האינדקס יכיל כמויות עצומות של נתונים.

תיאור של הממשק שצריך להיות ממומש מתואר בעמודים הבאים. לאחר תיאור הממשק יש תיאור של הניתוח אותו תצטרכו להגיש ביחד עם הקוד.

```
class SlowIndexWriter:
     def slowWrite(self, inputFile, dir):
          """Given product review data, creates an on
          disk index
          inputFile is the path to the file containing
          the review data
          dir is the directory in which all index files
          will be created
          if the directory does not exist, it should be
          created"""

     def removeIndex(self, dir):
          """Delete all index files by removing the given
          directory"""


class IndexReader
     def __init__(self, dir):
```

```
        """Creates an IndexReader which will read from
        the given directory"""

    def getProductId(self, reviewId):
        """Returns the product identifier for the given
        review
        Returns null if there is no review with the
        given identifier"""

    def getReviewScore(self, reviewId):
        """Returns the score for a given review
        Returns -1 if there is no review with the given
        identifier"""

    def getReviewHelpfulnessNumerator(self, reviewId):
        """Returns the numerator for the helpfulness of
        a given review
        Returns -1 if there is no review with the given
        identifier"""

    def getReviewHelpfulnessDenominator(self, reviewId):
        """Returns the denominator for the helpfulness
        of a given review
        Returns -1 if there is no review with the given
        identifier"""

    def getReviewLength(self, reviewId):
        """Returns the number of tokens in a given
        review
        Returns -1 if there is no review with the given
        identifier"""

    def getTokenFrequency(self, token):
        """Return the number of reviews containing a
        given token (i.e., word)
        Returns 0 if there are no reviews containing
        this token"""

    def getTokenCollectionFrequency(self, token):
        """Return the number of times that a given
        token (i.e., word) appears in
        the reviews indexed
        Returns 0 if there are no reviews containing
        this token"""
```

```
def getReviewsWithToken(self, token):
     """Returns a series of integers of the form id-
     1, freq-1, id-2, freq-2, ... such
     that id-n is the n-th review containing the
     given token and freq-n is the
     number of times that the token appears in
     review id-n
     Note that the integers should be sorted by id
     Returns an empty Tuple if there are no reviews
     containing this token"""


def getNumberOfReviews(self):
     """Return the number of product reviews
     available in the system"""

def getTokenSizeOfReviews(self):
     """Return the number of tokens in the system
     (Tokens should be counted as many times as they
     appear)"""

def getProductReviews(self, productId):
     """Return the ids of the reviews for a given
     product identifier
     Note that the integers returned should be
     sorted by id
     Returns an empty Tuple if there are no reviews
     for this product"""
```

4. דרישות האנליזה

בנוסף להגשת הקוד, יש להגיש אנליזה של מבנה האינדקס שבניתם. בפרט:

- הצגת הפרטים המדויקים של מבנה האינדקס שמימשתם. הצגת הפרטים צריכה להיות מאוד ספציפית וצריכה לאפשר לקורא להבין בדיוק את הפורמט של קבצי האינדקס המאוחסנים בדיסק. יש להוסיף תרשים המתאר את מבנה האינדקס.
- הסבר אילו חלקים של האינדקס נקראים לזיכרון בעת יצירת האובייקט IndexReader ואילו חלקים ייקראו לפי הצורך.
- יש לנתח בצורה תיאורטית מהו הגודל הצפוי (בבתים) של האינדקס. בניתוח, גודל האינדקס צריך להיות פונקציה של גודל הקלט (לצורך כך השתמשו במשתנים כדי לייצג גדלים שונים של הקלט)

5. בדיקת התרגיל

בבדיקת התרגיל ייעשה שימוש בערכת נתונים קטנה בסדרי גודל של הערכות הנמצאות באתר הקורס.

בדיקת התרגיל נעשית בעזרת מערכת אוטומטית. כדי שבדיקת התרגיל שלכם לא תיכשל (ותגרום להורדה בציון) הקפידו היטב על ההנחיות שבסעיף 6.

6. הגשת התרגיל

- התרגיל יוגש דרך אתר המכללה בפורמט ZIP.
- עבור כל זוג יש להגיש רק הגשה אחת. שם הקובץ צריך להיות ID1_ID2.zip כאשר ID1 ו ID2 הם מספרי הזהות של הסטודנטים המגישים את הפרויקט.
- **קבצי הקוד צריכים לכלול שני קבצים עם השמות SlowIndexWriter.py ו IndexReader.py. תכנית הבדיקה תייבא (import) קבצים אלו כך שחשוב שתקפידו על השמות הנכונים (כולל אותיות גדולות וקטנות) קבצים אלו יכילו את המחלקות שנדרשתם לפתח.**
- **במידה והפרויקט שלכם מכיל קבצי קוד נוספים (מחלקות נוספות), באחריותכם לייבא אותם (import) מתוך הקבצים SlowIndexWriter.py ו IndexReader.py.**
- יש לוודא כי הקובץ שהועלה הוא בפורמט הנכון ומכיל את כל הקבצים הרלוונטיים כולל:
  - קבצי הקוד
  - קובץ README הכולל הנחיות רלוונטיות לקומפילציה או להרצה, שמות ותעודת זהות של הסטודנטים המגישים את התרגיל.
  - קובץ בשם analysis.pdf שיכיל את הניתוח שהתבקש בתרגיל.