# אנליזה של הקוד

בר אלוני - [redacted] - baraloni
לאה אורלין - [redacted] - orlinleah

## 1. מבנה האינדקס

תחת התיקייה "**index**" אנחנו שומרות 2 "מבני נתונים" (הם אינם אובייקטים, אלא מערכים שאינם מקובצים יחד. מימשנו כך כי רצינו להמנע מהoverhead המתווסף להגדרת אובייקטים שישמשו לקיבוץ הפונקציונליות).
נפרט עליהם:

1. <u>Lexicon</u>:
בתיקייה "**Lexicon**" תחת "**index**" .
בחרנו לממש את המילון בשיטת ה-blocking.

למילון כזה יש 4 שדות:
- Frequencies.
- Posting Ptr.
- Length.
- Term Ptr.

כל אחד מהם מיוצג ע"י מערך, והוא נשמר לדיסק בקובץ משלו (הסברים נוספים בנוגע לשמירת כל המסמכים תחת 1.1).

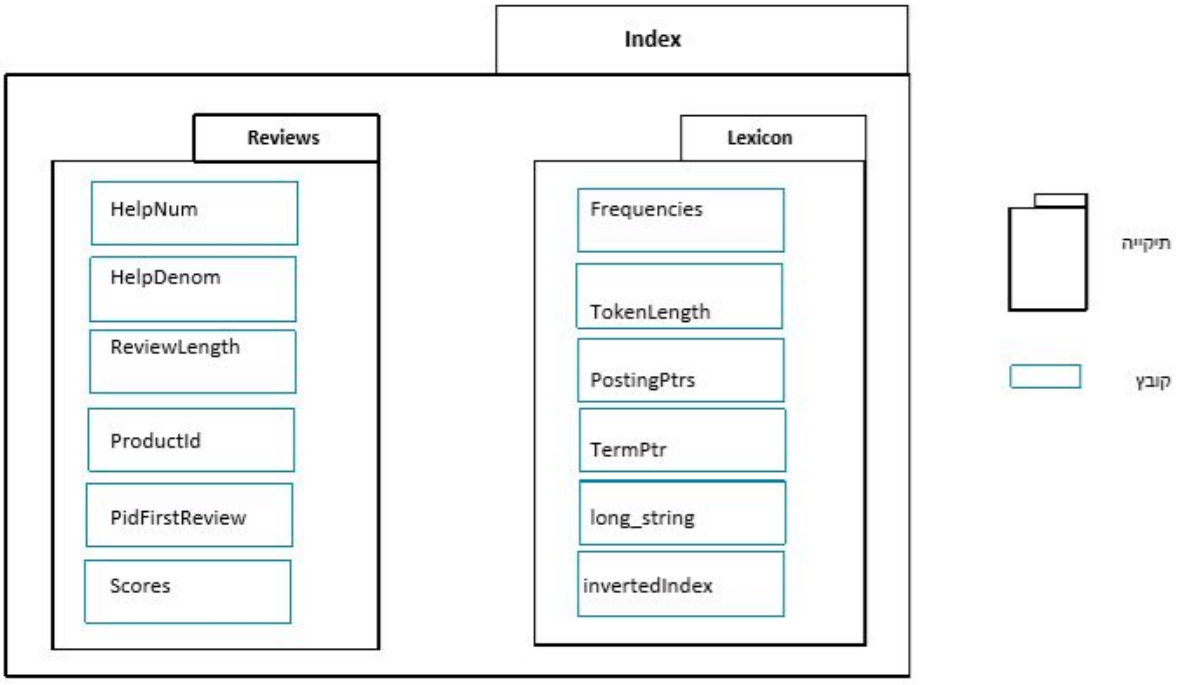
בנוסף נשמור בקבצים נפרדים את:
- terms (כל המילים באופן משורשר) .
- Inverted Index (כל ה- posting lists משורשרות).

2. <u>Reviews</u>:
בתיקייה "**Reviews**" תחת "**index**" .

למבנה נתונים זה יש 6 שדות:
- Helpfulness numerator.
- Helpfulness denominator.
- Review Length.
- Product Id.
- Product Id´s first Review Id.
- Scores.

כל שדה הוא למעשה מערך, והוא נשמר לדיסק בקובץ משלו.

**איור של האינדקס השמור:**




## 1.1. שמירת הקבצים לדיסק:

- בכל הקבצים (פרט ל-inverted index) שמרנו את האלמנטים שנמצאים במערכים כאוסף בתים לפי הגודל המקסימלי המצופה מאותו משתנה, כדי לקיים ככל האפשר שמירה יעילה מבחינת זכרון.
  שמרנו כל משתנה בקובץ נפרד כדי לתמוך בגדילה של התוכנית, וגם כדי לתמוך ב:
  - כתיבה מהירה- של מספר בתים גדול במקום בקריאה של שורה-שורה.
  - קריאה מהירה- לקרוא תוכן קובץ במקום לחפש בקובץ ולקרוא מהסמן.

- את ה- inverted index כתבנו בצורה דחוסה לפי Group Varint, ושמרנו בקובץ נפרד על הדיסק כאמור.

## 2. ייבוא לזכרון המרכזי בזמן קריאה

1. כל הלקסיקון נקרא לזכרון המרכזי בעת אתחול ה-Reader.
2. רשימת ה-terms המשורשרת נקראת לזכרון המרכזי בעת אתחול ה-Reader.
3. ה- Inverted Index **לא** נקרא לזכרון המרכזי בעת אתחול ה-Reader, הוא נקרא לפי צורך- כאשר יש לגשת ל-posting list מסויים. במקרה כזה אנחנו קוראים את מספר הבתים ש-posting list זה מחזיק.
4. כל "טבלת" ה-Reviews נקראת לזכרון המרכזי בעת אתחול ה-Reader.

## 3. הערכת גודל האינדקס

<u>נסמן:</u>

- T- מספר הטוקנים השונים.
- K- גודל בלוק.
- TE- מספר האותיות במילה ממוצעת.
- R- מספר הביקורות השונות.
- P- מספר המוצרים השונים.
- D- מספר הביקורות שבה מופיעה מילה בממוצע (מספר הכניסות ב-posting list בממוצע).

<u>שמירת הלקסיקון:</u>

- Frequencies:
  נשמור את השדה בתור int, ומכיוון שאנחנו שומרים כניסה אחת לכל מילה, נשמור סה"כ: $T \times 4\ bytes$.
- Posting Ptr:
  נשמור את השדה בתור long, ומכיוון שאנחנו שומרים כניסה אחת לכל מילה, נשמור סה"כ: $T \times 8\ bytes$.
- Length:
  נשמור את השדה בתור byte, ומכיוון שאנחנו שומרים כניסה אחת לכל K-1 מתוך K מילים, נשמור סה"כ: $\frac{K-1}{K} \times T\ bytes$.
- Term Ptr:
  נשמור את השדה בתור int, ומכיוון שאנחנו שומרים כניסה אחת למילה אחת בלבד מכל K מילים, נשמור סה"כ: $\frac{1}{K} \times T \times 4\ bytes$.

כלומר בסה"כ נשמור: $(4 + 8 + \frac{K-1}{K} + \frac{4}{K}) \times T\ bytes = (12 + \frac{K+3}{K}) \times T\ bytes$.

<u>שמירת ה-terms:</u>

נשמור T מילים, כל אחת באורך ממוצע של TE אותיות. כל אות נשמרת ב $byte$ אחד.
ולכן נשמור בסה"כ: $T \times TE\ bytes$

<u>שמירת ה-Inverted Index:</u>

נניח ש:

- המספר המקסימלי של ביקורות נכנס ב-2 בתים (ולכן ההפרש המקסימלי בין 2 מספרי ביקורות ייכנס אף הוא ב-2 בתים לכל היותר).
- תדירות מילה ממוצעת בביקורת אחת נכנסת לכל היותר גם היא ב-2 בתים.

תחת ההנחות הללו, נצטרך לקודד לכל מילה מ-T המילים המופיעות בלקסיקון, רשימה של 2D מספרים שייצוג כל אחד מהם נעשה ב-2 בתים.
בשיטת ה-group varint מקצים בית נוסף לאחסון מספר הבתים הדרוש לכל אחת מארבעת המילים שמופיעות אחריו, לכן נוסיף $\left[\frac{2D}{4}\right]$ בתים שיחזיקו את התחיליות של כל 2D המספרים.
כלומר בסה"כ נשמור: $T \times 2D \times 2 + \left[\frac{2D}{4}\right] bytes$.

שמירת ה-Reviews:

- Helpfulness numerator:
  נשמור את השדה בתור short, ומכיוון שאנחנו שומרים כניסה אחת לכל ביקורת, נשמור סה"כ: $R \times 2\ bytes$.
- Helpfulness denominator:
  נשמור את השדה בתור short, ומכיוון שאנחנו שומרים כניסה אחת לכל ביקורת, נשמור סה"כ: $R \times 2\ bytes$.
- Review Length:
  נשמור את השדה בתור int, ומכיוון שאנחנו שומרים כניסה אחת לכל ביקורת, נשמור סה"כ: $R \times 4\ bytes$.
- Scores:
  נשמור את השדה בתור byte, ומכיוון שאנחנו שומרים כניסה אחת לכל ביקורת, נשמור סה"כ: $R\ bytes$.

סה"כ נשמור: $9R\ bytes$.

- Product Id:
  לפי הנתונים נניח שכל מוצר מורכב מ-10 אותיות, ונשמור כל אות בבית אחד. מכיוון שאנחנו שומרים P מזהי מוצרים שונים, נשמור סה"כ: $P \times 10\ bytes$.
- Product Id´s first Review Id:
  נשמור את השדה בתור int, ומכיוון שאנחנו שומרים כניסה אחת לכל ProductId, נשמור סה"כ: $P \times 4\ bytes$.

סה"כ נשמור: $14P\ bytes$.