

דו"ח חלק 1- מנוע חיפוש:

1. עיצוב תוכנה:

a. הסבר מפורט על אופן פעולה של התוכנית, המחלקות והשיטות של כל מחלקה.

התוכנית מחולקת:

Model: מכיל את הקוד ל-Read File, ל- parser ול- indexer

Package model.corpusDictionary

Class corpusDictionary – מחלקת סינגלטון המכילה את מבני הנתונים שכל parser מחזיק ובסופו של דבר מחזיקה במילון הסופי. המחלקה מכילה:

- TreeMap עבור המילים הבודדות המפורסרות.
 - TreeMap עבור כל הישגיות (מוחזק בנפרד מהמילים על מנת לבדוק האם ישות מופיעה יותר מפעם אחת בכל corpus)
 - HashMap עבור המידע על הטקסטים ב- corpus.
- הערה: עבור מבנה נתונים TreeMap מימשנו comparator חדש על מנת להתמודד עם המיון של האותיות הגדולות/ קטנות במילון.
- השיטות של המחלקה:

- addSesceDic - מוסיפה ישות למבנה הנתונים של היישות.
- addNumber - מוסיף term שעומד באחד מהחוקים של מספר ב- parser
- addWord - מוסיף term שעומד באחד החוקים של word ב- parser
- addInfoToArticleInfo - הפונקציה סופרת עבור כל מסמך את מספר המילים הייחודיות ואת מספר המופעים המקסימלי של המילה שמופיעה הכי הרבה במסמך.
- getDictionary - השיטה שולחת העתק של dictionary.
- buildDictionary - הפונקציה בונה את postingFile ועבור כל term רושם כמה פעמים היא מופיעה בכל corpus.
- getEssenceDic - השיטה מחזירה את מבנה הנתונים שמחזיק את כל היישגיות (כולל היישגיות שמופיעות רק פעם אחת בכל corpus)
- reset - השיטה מאפסת את כל מבני הנתונים שבזיכרון.
- Comparator - השיטה מממשת השוואה בשביל treeMap על מנת שנוכל להכניס כל term במילון עם אות גדולה/ קטנה באופן ממוין.

Package model.parser

Class Parse – מחלקת העל ששולטת ומפעילה על כל מילה את כלל החוקים המחלקה מקבלת מחרוזת text ומפרסרת אותו, בנוסף המחלקה קוראת את קובץ stopWords.

שיטות של המחלקה:

- initParse - השיטה קוראת מקובץ לתוך מבנה נתונים את כל ה- stopWords
- parseDoc - הפונקציה מקבלת מסמך ועבור כל term במסמך מעבירה אותה את רצף החוקים ובסופו של דבר מכניסה אותו ל- TreeMap עם או בלי stemming.
- deleteDelimiters - השיטה מקבלת מילה ומורידה את כל סימני הפיסוק שלא רלוונטים.
- checkStopWord - השיטה מקבלת מילה ומחזירה אמת אם המילה היא stopWords

Class RulesFactory - המחלקת מממשת את ה- Factory Design Pattern. אנחנו משתמשים בה במחלקה parser על מנת לקרוא לחוקים עבור כל מילה.

שיטות של המחלקה:

1. `getRuleChecker` - השיטה מחזירה אובייקט מסוג `Chok` חדש לפי המחרוזת שהיא מקבלת מה `parser`.

Package model.parser.ParserRule

ממשקים:

interface IRuleChecker - הממשק מגדיר פונקציה כללית אותה החוקים ב `package` יצטרכו לממש. כל `Chok` על פי ייעודו. השיטה אותה יצטרכו לממש - `roleChecker`.

המחלקות שקיימות ב `package` זה מממשות כל אחת `Chok` ספציפי מרצף החוקים עבור כל `term` ב `parser`. כל המחלקות יורשות ממחלקת העל `ARuleChecker`. להלן המחלקות:

1. **ARuleCheker** - מחלקה אבסטרקטית של `Chok` כללי המממשת את הממשק `IRuleChecker`. שיטות של המחלקה:

- `addDictionary` - השיטה מוסיפה `term` שעומד באחד מחוקי המספרים.
- `addDictionaryWord` - השיטה מוסיפה למילון `term` שעומד באחד מחוקי המילים.
- `addToEssenceDic` - השיטה מוסיפה למילון ביטוי חדש – אורך מקסימלי 3 מילים.

2. **ANumberRules** - מחלקה אבסטרקטית היורשת מהמחלקה הראשית `ARuleCheker` עבור חוקים המכילים מספרים.

3. **KiloMeterRepresentationRule** - המחלקה יורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת `Chok` שהוספנו עבור ביטויים המכילים מספר ואת המילה `kilo` או `meter` והופכת אותם ליישות יחידה `1 kg/1 m`.

4. **NumberRepresentation** - המחלקה יורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת את החוק עבור ביטויים (מספר ומילה אחריה) אשר מכילים את המילים `thousand/million/billion` והופכת אותם לפורמט אחיד - `number-k/m/b`.

5. **phoneNumberRules** - מחלקה היורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת `Chok` שהוספנו עבור ביטויים המכילים מספרי טלפון מהפורמט `xxxxxx (xxx) x` כאשר `x` הוא מספר. החוק דואג לשמור את מספר הטלפון כביטוי אחד.

6. **PercentageRepresentationRule** - המחלקה יורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת `Chok` עבור ביטויים המכילים אחוזים ומעביר אותם לפורמט אחיד - `X%` כאשר `X` מהווה מספר.

7. **PriceRepresentationRule** - המחלקה יורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת `Chok` עבור ביטויים המכילים מחירים ומעבירה אותם לפורמטים שהוגדרו.

8. **RangeRule** - המחלקה יורשת מהמחלקה `ANumberRules`. שיטה:

- `ruleChecker` - השיטה מממשת `Chok` עבור ביטויים הכוללים בתוכם `"-"` או עבור ביטויים `"Between x and x"` ושומרת אותם כ `term` יחיד.

9. **ADateRule** - מחלקה אבסטרקטית עבור חוקים המכילים תאריכים/חודשים. שיטות של המחלקה:

- `ChangeMonthToNumber` - השיטה מקבלת שם של חודש ומחזירה את המספר שהחודש מציין
- `isNumeric` - בודק אם המחרוזת שהתקבלה הוא מספר.

10. **DayMonthRule** - המחלקה יורשת מהמחלקה `ADatesRule`.

שיטה:

- `ruleChecker` - השיטה מעבירה את כל הפורמטים של תאריכים לפורמט אחיד `month-day`.

11. **MonthYearRule** - המחלקה יורשת מהמחלקה `ADateRule`.

שיטה של המחלקה:

- `ruleChecker` - השיטה מקבלת פורמטים שונים של חודשים ושנים והופכת אותם לפורמט אחיד `YYYY-MM`.

12. **AWprdsRule** - מחלקה אבסטרקטית עבור מילים רגילות במילון.

13. **ExpressionsRepresentationRule** - מחלקה יורשת מהמחלקה `AWprdsRule`. המחלקה

אחראית על הוספת ביטויים למילון.

שיטה:

- `ruleChecker` - השיטה מוסיפה `expression` למילון עבור ביטויים שכל המילים בהם מתחילים באות גדולה. השיטה מוסיפה ביטויים המכילים לכל היותר 3 מילים.

14. **SingleWordRule** - המחלקה יורשת מהמחלקה `AWprdsRule`.

שיטה:

- `ruleChecker` - השיטה מוסיפה למילון מילים בודדות שכוללת אותיות בשפה האנגלית ומוודא שהמילה אינה `stopWords`.

Package model.ReadFile

ReadFile - המחלקה אחראית לקרוא את הטקסטים מהקבצים `corpus`.

שיטות:

- `ReadFile` - השיטה מקבלת מחרוזת `path` וקוראת מהקבצים `corpus` את הטקסטים ומעבירה אותם ל`parser` ואז שולחת אותם ל`indexer`.
- `getNumOfArticle` - השיטה מחזירה את כמות המאמרים שהשיטה `ReadFile` קראה.

Package model.Indexer

Indexer - המחלקה אחראית להעביר את כל ה `terms` שמתקבלים מה`parser` ולהכניס אותם לקובץ `posting` וממנו לייצר את קובץ ה-`dictionary`.

בסופו של תהליך ה-`indexer` ישמור שלושה קבצים:

- קובץ `posting` עבור `term` בפורמט הבא: `term$df,tf,listOfOcc#`
- `documentInfo.txt` - קובץ `posting` המכיל מידע על המסמכים בפורמט הבא:
`doc$max_tf | wordOfMaxtf | amount of unique words | Title |`
- `dictionary.txt` - מילון הנוצר מקבצי `posting` המילון מכיל עבור כל `terms` את הפרטים הבאים:
`term> The amount of documents is listed | point to posting file`
- `articleIndex.txt` - הקובץ נותן מספר בודד יחודי עבור כל `key` של מסמך.

השיטות של המחלקה:

- `createInvertedIndex` - השיטה מקבלת את מבני הנתונים שנוצרו ב`parser` המכילים את הנתונים אודות ה `terms` השונים ואודות המסמכים ומייצרת את קבצי `posting` של ה `term` וה- `documentInfo`
- `insertTextFirstIterate` - המחלקה מקבלת מהפונקציה `createInvertedIndex` את ה-`TreeMap` שנוצר ב`parser` עבור `terms` ומייצר עבורו עבור כל `terms` קובץ `posting` ע"פ הפורמט שהוצג למעלה.

- Merge - השיטה לוקחת שני קבצי posting שנוצרו באיטרציות של תהליך posting וממזגת אותם לקובץ אחד בעזרת הפונקציה mergePostingFile.
- mergePostingFile - השיטה מקבלת path לשני קבצי posting זמניים ומאחדת אותם באופן ממין לקובץ posting יחיד.
- saveDocInfo - השיטה מקבלת את ה-TreeMap מהparser ששומר את הנתונים עבור כל מסמך ומייצר קובץ posting המכיל את המידע עבור כל המסמכים עפ"י הפורמט שהוצג למעלה.
- CreateDictionary - השיטה מאחדת בשלב האחרון את המבנה נתונים של המילים הבודדות והמספרים עם מבנה הנתונים של הישויות שמופיעות יותר מפעם אחת בכל הcorpus.
- CreateDIC - השיטה מקבלת path עבור קובץ posting ומייצרת TreeMap עם הנתונים עבור כל term המעובדים מקובץ הposting לצורך יצירת המילון. השיטה מומשה לצורך הפעלת כפתור ה-Load בGUI
- saveDictionary - השיטה יוצרת קובץ dictionary.txt ורושמת אליו את כל הפרטים שקיימים בcorpusDictionary.
- saveEntities - השיטה מייצרת קובץ ישויות עבור כל הישויות שמצאנו בcorpus השיטה בסוף תהליך index בודקת האם ישויות מסויימת קיימת ביותר מקובץ אחד בcorpus ואם כן רושמת אותה לקובץ החדש.
- giveTermFromAllLine - השיטה מחזירה רק את הterm מהשורה עם כל הנתונים עליו.
- writeArticleIndex - השיטה מייצרת קובץ txt עבור מילון עם שם מסמך והkey הייחודי שאנחנו נותנים לו.

Package model.Stemmer

1. stemmer - מחלקה המשתמשת בporter stemmr עבור ה-terms.

Package Sample

1. Main - המחלקה יורשת מהמחלקה Application. המחלקה אחראית לייצר את ה-GUI עבור הפרוייקט. המחלקה מקבל scene מהמחלקה UserInteface ומוסיפה אותו לprimaryStage ומציגה אותו
2. UserInterface - המחלקה בונה את scenen שמכיל את כל הפונקציות של ה-GUI. המחלקה הינה מחלקת סינגלטון. שיטות של המחלקה:
 - getInstance - השיטה מחזירה את האובייקט.
 - BuildInterface - המחלקה מצרפת את כל האובייקטים הפונקציונליים של ה-GUI
 - addHBOXFirstOption - השיטה מצרפת ל-VBox את ה textFile וה-button לטעינת corpus.
 - AddVBOXSecOption - השיטה מצרפת ל-VBox את ה textFile וה-button לשמירת postingFile.
 - addCheckBox - השיטה מוסיפה את הפונקציונליות של אפשרות הבחירה בביצוע parser עם/ בלי stemming.
 - addFourButton - השיטה מוסיפה את כל הכפתורים לVBox.

b. אופן ההתמודדות עם מגבלת הזיכרון של המחשב והפעולות שנקטנו להביא לזמן ריצה מירבי. על מנת להתמודד עם מגבלת הזיכרון של המחשב אנחנו מבצעים את תהליך ה-parse ולאחר מכן את האינדוקס ויצירת קבצי posting עבור 8 תיקיות כל פעם. לאחר כל איטרציה כזאת אנחנו רושמים קובץ posting זמני למחשב, מאפסים את מבני הנתונים שלנו וכך מקלים בצורה משמעותית על הזיכרון ומתחילים איטרציה נוספת עבור 8 תיקיות חדשות. במהלך העבודה על הפרוייקט ביצענו בדיקות על מנת לבדוק מהי כמות התיקיות המיטבית עליה כדי לרוץ כדי למזער את זמן הריצה ובנוסף להכביד כמה שפחות ה-RAM וה-CPU. לאחר הבדיקות קיבלנו כי ריצה על 8 תיקיות מביאה לזמן ריצה מיטבי ולכן בחרנו במספר תיקיות זה. מבחינת הרצון להביא לזמן ריצה מרבי נקטו בפעולות הבאות:

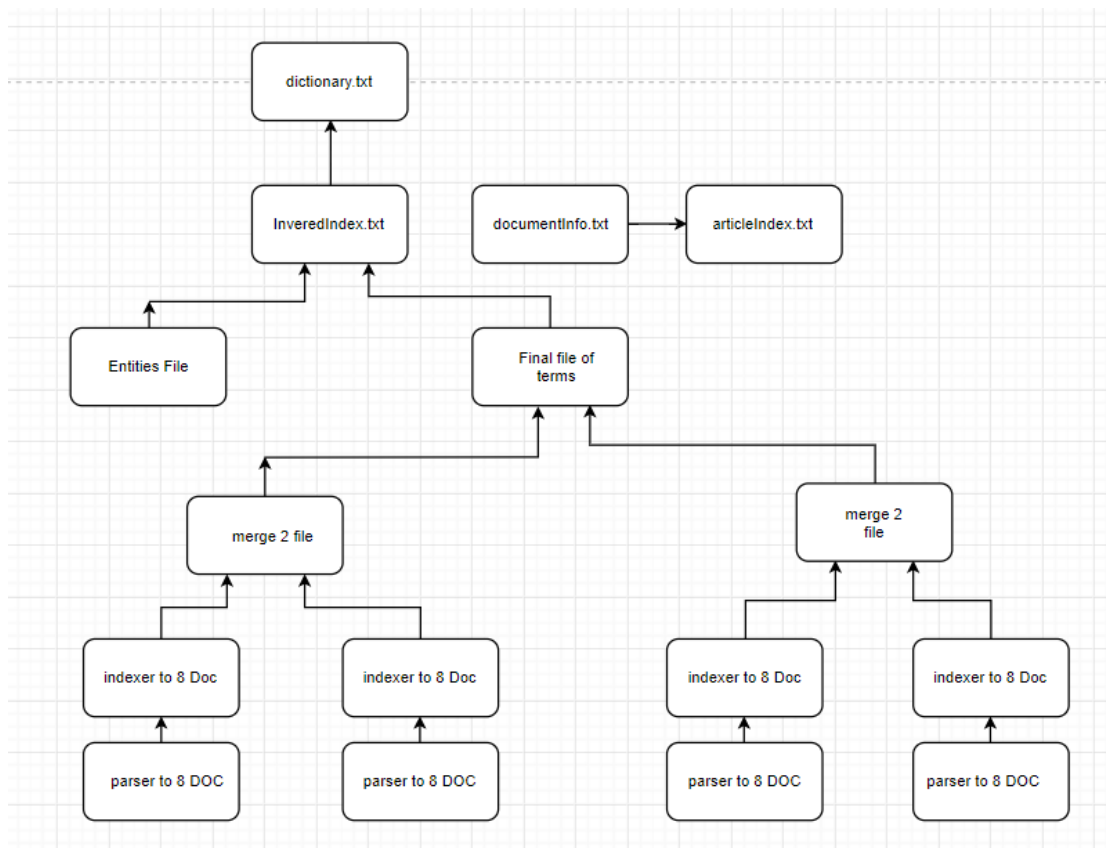
1. כלל הנתונים שלנו נשמרים כבר בתהליך ה-parse ב-mapTree דבר שגורם לכך שהם ממוינים לאורך כל תהליך ה-parse ואין לנו צורך להתעסק כלל עם מיון בזמן ה-indexer.
2. מיזוג קבצי ה-posting הזמניים שלנו מתבצע באופן אופטימלי (בדומה ל-moiltiway, ניתן לחשוב על כך גם כטורניר כדורגל עם רבע, חצי וגמר). לאחר בדיקה והרצת מקרים קטנים נוכחו לדעת כי באופן ביצוע מיזוג זה אנחנו מקטנים למינימום את הקריאה מהקבצים (בניגוד למיזוג של 2 קבצים ולאחר מכן מיזוגם עם הקובץ השני, דבר שגורם לקריאות רבות נוספות ומיותרות).
3. המיזוג שלנו מתבצע בזמן קריאה אופטימלי של $O(N+M)$ כאשר N ו-M הם האורכים של הקבצים הזמניים. המיזוג מתבצע בעזרת שני מצביעים, אחד עבור כל טקסט זמני שמתקדם כל פעם שנוסף term מהקובץ שלו לקובץ הממוין.
4. שמירת הנתונים על המסמכים מתבצע בעזרת מבנה נתונים linkedHashMap מה שמאפשר לנו לבצע עליו את כל הפעולות בזמן ממוצע של $O(1)$.
5. במהלך הפירסור אנחנו מונעים מעברים מיותרים על כל מילה.

c. אופן שמירת קבצי ה-posting, סוג הקבצים, כמות הקבצים. מה מכיל כל קובץ. אנחנו שומרים בסך הכל 4 קבצים, כולל קובץ המילון:

- קובץ posting עבור term בפורמט הבא: term\$df,tf,listOfOcc#
 - documentInfo.txt - קובץ posting המכיל מידע על המסמכים בפורמט הבא: doc\$max_tf || wordOfMaxtf || amount of unique words || Title ||
 - dictionary.txt - מילון הנוצר מקבצי ה-posting המילון מכיל עבור כל terms את הפרטים הבאים:
- term> The amount of documents is listed | point to posting file
- articleIndex.txt - הקובץ נותן מספר בודד יחודי עבור כל key של מסמך.

כל הקבצים שנשמרים הם קבצי txt.

d. הסיבות לבחירת גודל קבוצת המסמכים החלקית והצגת תיעוד יצירת הקבצים ההופכיים (מילון וקובץ posting).
 במהלך העבודה על הפרוייקט ביצענו בדיקות על מנת לבדוק מהי כמות התיקיות המיטבית עליה כדי לרוץ כדי למזער את זמן הריצה ובנוסף להכביד כמה שפחות ה-RAM וה-CPU. לאחר הבדיקות קיבלנו כי ריצה על 8 תיקיות מביאה לזמן ריצה מיטבי ולכן בחרנו במספר תיקיות זה.



e. פרטי האינפורמציה הנוספים ששמרנו הם:

1. בקובץ documentInfo מלבד הכמות של המילה הכי נפוצה בטקסט שמרנו גם את המילה עצמה.
2. בקובץ documentInfo שמרנו את הכותרת של מסמך.
3. בקובץ dictionary שמרנו pointer עבור כל term לשורה שה-term מופיעה בה בקובץ ה-posting.
4. שמרנו קובץ נוסף: articleIndex בו נתנו לכל מסמך key ייחודי על מנת להקטין את הגודל שבו נשמר קובץ posting בדיסק.

f. שני החוקים הנוספים שהוספו ל-parser הם:

1. עבור 2 terms צמודים אשר מהווים מספר טלפון על פי הפורמט הבא: xxxxxx (xxx) , שמרנו אותם ביחד כמספר טלפון ולא כשני terms נפרדים.

PN: (714)	523-2070>1	958062
PN: (714)	524-2640>1	958063
PN: (714)	524-6951>1	958064
PN: (714)	524-7011>1	958065
PN: (714)	524-8408>1	958066
PN: (714)	525-3728>1	958067
PN: (714)	525-4567>2	958068
PN: (714)	525-7735>1	958069
PN: (714)	525-8464>1	958070
PN: (714)	526-1690>2	958071
PN: (714)	526-5071>1	958072
PN: (714)	526-8258>1	958073
PN: (714)	527-0680>1	958074
PN: (714)	527-1234>1	958075
PN: (714)	527-7107>1	958076
PN: (714)	527-7711>1	958077
PN: (714)	527-7727>1	958078
PN: (714)	528-0427>1	958079
PN: (714)	528-1171>1	958080
PN: (714)	528-1479>1	958081
PN: (714)	528-4254>1	958082
PN: (714)	529-0428>1	958083
PN: (714)	529-2233>1	958084
PN: (714)	529-2993>1	958085
PN: (714)	529-4589>1	958086
PN: (714)	530-0930>1	958087
PN: (714)	530-2318>1	958088
PN: (714)	530-5230>1	958089
PN: (714)	530-6111>1	958090
PN: (714)	530-8871>2	958091
PN: (714)	531-6157>1	958092
PN: (714)	531-9456>1	958093
PN: (714)	532-0383>1	958094
PN: (714)	532-0421>1	958095
PN: (714)	532-0629>1	958096
PN: (714)	532-3142>1	958097
PN: (714)	532-5800>1	958098
PN: (714)	533-2450>1	958099
PN: (714)	534-0250>1	958100
PN: (714)	534-0961>1	958101
PN: (714)	534-4555>1	958102
PN: (714)	534-6370>1	958103
PN: (714)	535-0645>1	958104
PN: (714)	535-1336>1	958105
PN: (714)	535-1552>1	958106
PN: (714)	535-2211>1	958107
PN: (714)	535-3059>1	958108
PN: (714)	535-3281>1	958109
PN: (714)	535-5694>2	958110
PN: (714)	535-9815>1	958111
PN: (714)	536-0202>1	958112
PN: (714)	536-1454>3	958113
PN: (714)	536-4702>1	958114

2. עבור כל מספר שמצורף אליו מיד אחרי המילה meter/kilo שמרנו אותם כterms נפרד זאת כדי להשאיר את הקשר הקיים בין המספר ליחידת משקל שבאה לתאר אותו.

```
dictionaryStem - Notepad
File Edit Format View Help
'Onekg>1|1
-3.4 Dollars>1|2
/kg>3|3
0>13702|4
0 Dollars>68|5
0 M Dollars>16|6
0%>100|7
0,6%>1|8
0.0%>84|9
0.00%>2|10
0.00002%>1|11
0.0001%>1|12
0.0002kg>1|13
0.0003%>1|14
0.0005%>3|15
0.0009%>1|16
0.001%>4|17
0.002%>3|18
0.002kg>1|19
0.003%>1|20
0.0033kg>1|21
0.004%>1|22
0.005%>4|23
0.005kg>1|24
0.006%>1|25
0.007%>3|26
0.008%>1|27
0.0089kg>1|28
0.01%>39|29
0.012%>1|30
0.01724%>1|31
0.02%>30|32
0.022%>2|33
0.025%>3|34
0.03%>24|35
0.0326kg>1|36
0.035%>1|37
0.035kg>2|38
0.036%>1|39
0.04%>17|40
0.04-0.05kg>2|41
0.042%>1|42
0.045%>1|43
0.05%>41|44
0.06%>10|45
0.0625%>1|46
0.07%>16|47
0.0721kg>1|48
0.076%>1|49
0.07kg>2|50
0.08%>35|51
0.083%>4|52
0.085%>1|53
```


g.

1. בקובץ dictionary שמרנו pointer עבור כל term לשורה שה term מופיעה בה בקובץ ה-posting.
2. שמרנו קובץ נוסף: articleIndex בו נתנו לכל מסמך key ייחודי על מנת להקטין את הגודל שבו נשמר קובץ posting בדיסק.

h. האם השתמשנו במהלך העבודה בקוד פתוח?

במהלך העבודה השתמשנו בקוד פתוח עבור המחלקה stemmer. המחלקה מקבלת term ומעבירה אותו סדרת פעולות עד שהיא מחזירה אותו בפורמט אחיד עבור כל ההטיות זכר/נקבה/יחיד/רבים שקיימות. השתמשנו במחלקה זו על מנת שנוכל לצמצם את המילון וקבצי posting שלנו זאת מכיוון שבעזרת המחלקה נוכל לקבל ביטוי אחד משותף לכמה וכמה מילים דומות שמופיעות במילון שהשוני בהם הוא רק בהטייה. בנוסף בשלב הבא נוכל לאחזר בצורה טובה יותר שאילות כיוון שלאחר שנעביר גם את השאילות דרך ה stemmer נוכל למצוא דמיון לא רק עבור מילה ספציפית שקיימת בשאילות אלא גם עבור כל ההטיות שלה.

i. לא קיים מידע נוסף.

2.

a. כמות הterms השונים במאגר לפני stemming:

1,471,281

b. כמות הterms השונים במאגר אחרי stemming:

1,333,390

c. כמות הterms השונים שהם מספרים שקיימים במאגר:

88,523

d. הדפסת רשימת 10 ה-terms השכיחים ביותר במאגר לפי סדר שכיחות:

עם stemming:

```
mr => 436631
cent => 321249
pounds => 307977
year => 303692
government => 289819
1.99K => 251742
people => 234830
state => 211692
time => 209705
market => 205249
```

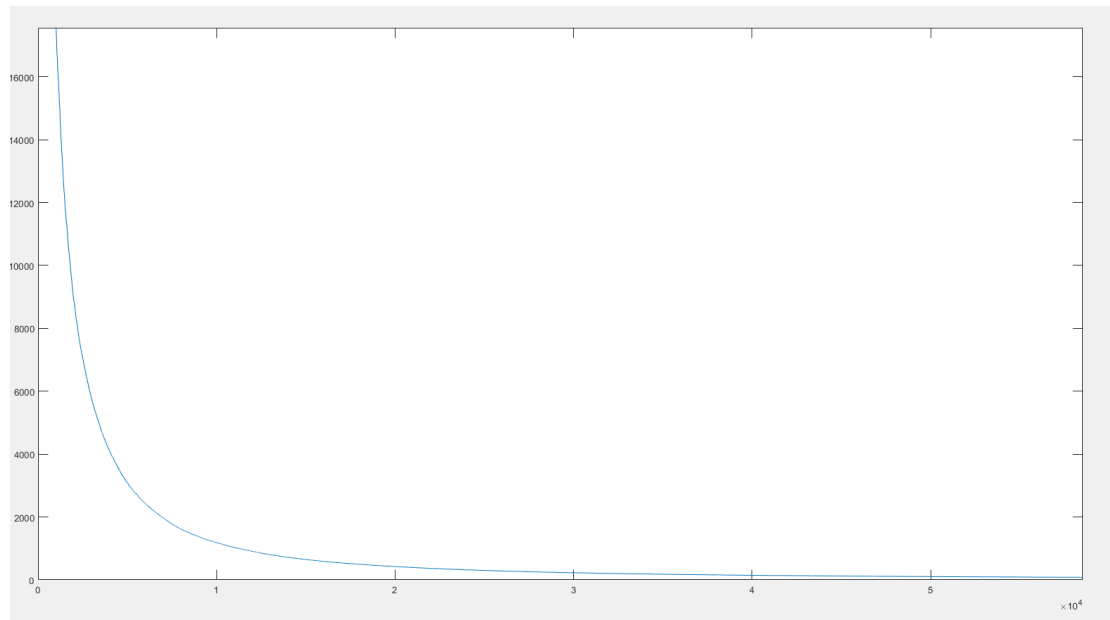
בלי stemming:

```
year => 526264
mr => 447009
govern => 356113
cent => 340095
state => 334167
compani => 333912
pound => 315552
market => 283058
peopl => 277920
time => 275287
1.99K => 251742
-----
```

הדפסת רשימת 10 ה-terms הכי פחות שכיחים (לפני stemming)

```
8850 M Dollars => 1
LAMIDAT => 1
926.52 => 1
Cumming-Bruce => 1
chemical-metallurgical => 1
Franco-Senegalese => 1
433400 M Dollars => 1
ghannuj => 1
Tory-packed => 1
golovnyy => 1
```









e. הצגת המילים הייחודיות במאגר על גרף Zipf's Law. הסבירו האם העקומה אכן דומה למה שאמור להיות. (הגרף בוצע בעזרת תוכנת MATLAB)



f. הדפסת רשימת הterms במסמך שמספרו FBIS3-3366 ממזין, לפני stemming עם תדירות עבור כל מילה במסמך.

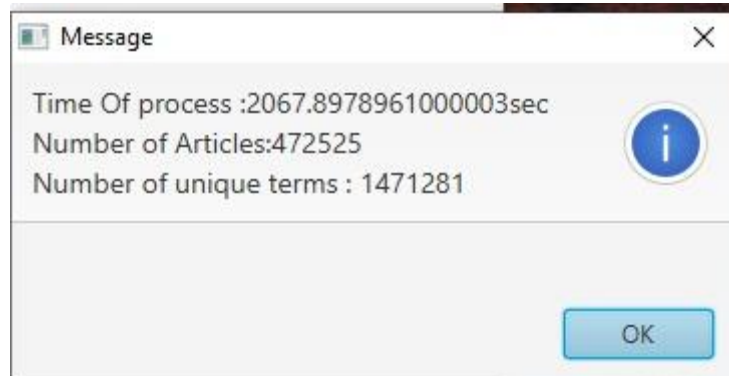
```
CHINESE => 5
COMMITTEE => 5
NATIONAL => 4
CONFERENCE => 4
PEOPLE'S => 4
POLITICAL => 4
CONSULTATIVE => 4
CPPCC => 4
EIGHTH => 3
CHARTER => 3
amended => 3
SESSION => 3
19 => 2
ADOPTED => 2
effect => 1
decided => 1
proposed => 1
RESOLUTION => 1
STANDING => 1
TEXT => 1
TYPE => 1
XINHUA => 1
BFN => 1
MAR => 1
ARTICLE => 1
BEIJING => 1
1994-03 => 1
LANGUAGE => 1
|
```

g. הציגו את גודל ה-posting – נפח האחסון בKB עבור stemming וללא.

Name	Date modified	Type	Size
 DocumentInfo	12/18/2019 6:15 PM	Text Docu...	25,962 KB
 DocumentInfoStem	12/18/2019 6:53 PM	Text Docu...	25,703 KB
 articleIndex	12/18/2019 6:15 PM	Text Docu...	9,271 KB
 articleIndexStem	12/18/2019 6:53 PM	Text Docu...	9,271 KB
 dictionary	12/18/2019 6:17 PM	Text Docu...	33,097 KB
 dictionaryStem	12/18/2019 6:55 PM	Text Docu...	30,263 KB
 InvertedIndex	12/18/2019 6:17 PM	Text Docu...	633,061 KB
 InvertedIndexStem	12/18/2019 6:54 PM	Text Docu...	588,642 KB

h. הציגו את משך הזמן שלוקח לבנות את האינדקס על קבצי הCorpus.

בלי stemming: 34 דק'



עם stemming: כ-32 דק'.

