NLP – HW1

מגישים:

מוחמד חג'אזי – 213011026

מועאד עבד אללטיף - 209418896

שלב 1 – טיפול בטקסט

1) שליפת נתונים מתוך קבצי הפרוטוקולים:

```
def save_info_for_file(docx_files):
    files_list = []
    for docx_file in docx_files:
        parts = file_name_split(docx_file)
        if parts[1] == 'ptm':
            file_data = {'FileName ': docx_file, 'knessetNumber ': parts[0],
'CommitteeOrPlenary ': 'plenary'}
        else:
            file_data = {'FileName ': docx_file, 'knessetNumber ': parts[0],
'CommitteeOrPlenary ': 'committee'}

        files_list.append(file_data)
```

2) שליפת טקסט בעל תוכן:

בכל הפרוטוקולים שם הדובר מסתיים ב ״:״, ולכן התחלנו מנקודה זו. כמו כן, ברוב הוועדות/מליאות הדובר הראשון הוא ״היו״ר״ ולכן אנחנו רוצים להתחיל בשליפת הטקסט הנדרש כאשר מגיעים לפסקה(משפט) שמקיים את הנ״ל. יש כמה פרוטוקולים שהאופי שונה, כך שטיפלנו בהם בצורה קצת אחרת, למשל יש פרוטוקולים ששמות הדוברים נמצאים בין <<דובר>> ולכן המקרה הכללי לא עובד ולכן פיתחנו תנאים מיוחדים לזיהוי התוכן הנדרש בפרוטוקולים אלה.

בתוך הדיאלוג יש לפעמים תוכן שהוא לא שייך לאף אחד מהדוברים, והוא מופרד בשתי שורות ריקות מהתוכן ששייך לדובר שמלפניו ומאחרו. ולכן נעזרנו בדבר זה כדי לעצור אסיפת הטקסט של הדוברים.

לגבי שמות הדוברים, קודם כל ניקינו את החלק האחרון של המשפט(שם הדובר עם תוספות), כך שהוא מסתיים בסוגריים עם טקסט בתוכם ולכן הורדנו אותם. אחר כך, לקחנו את שתי המילים האחרונות שהם בעצם שם פרטי ושם משפחה של הדובר(חוץ ממקרים מיוחדים ובודדים השם מורכב מיותר משתי מילים).

3) חלוקה למשפטים:

הגדרנו Pattern מיוחד שמזהה אם קיים אחד מסימני הפיסוק, כך שהם מופיעות תמיד בסוף משפט

ולכן כשהוא רואה אותם בתוך פסקה הוא מפריד את המשפט מהפסקה עד הסימן שהוא ראה או את "---" שיש חלק מהמשפטים שמסתיימות בסיומת זו.

4) נקיון המשפטים:

בחלק הזה הגדרנו שלושה פאטרנס כך שהראשון עובד לזיהוי אותיות בעברית וספרות, במילים אחרות הוא מסנן מילים שלא רוצים אותם כפי שהוגדר(למשל מילים באנגלית), והשני עובד לזיהוי משפטים שיש בהם "--", אם הוא רואה אותם הוא זורק את כל המשפט. והשלישי עובר שורה חדשה, כך שהוא מוחק את השורות הריקות.

5) טוקניזציה:

אנחנו מטפלים בטוקנזציה בפונקציה word_tokenize הפונקציה עוברת על התווים (char by char) בלולאה :

טיפול בסימני פסוק:

לרוב התמודדנו עם סינמי פסוק כך שיהיו טוקנים נפרדים כמו : "-" במקרים של "ב–" , "ו-", גם לכל ",", ".", ":", "?", "!" וכו.. הפונקציה עושה אותם טוקנים נפרדים.

אבל הפונקציה מזהה פיסוקים בתוך מילים, ומטפלת במקרים חריגים כמו פיסוק בתוך מספר או בתוך מילה למשל:

התשנ"ג ,20 :14, 5,553,636 , 59%, 7076/04 במקרים אלה נתייחס לכל ה"ביטוי" כטוקן אחד.

בסופו הפונקציה תחזיר מערך של טוקנים.

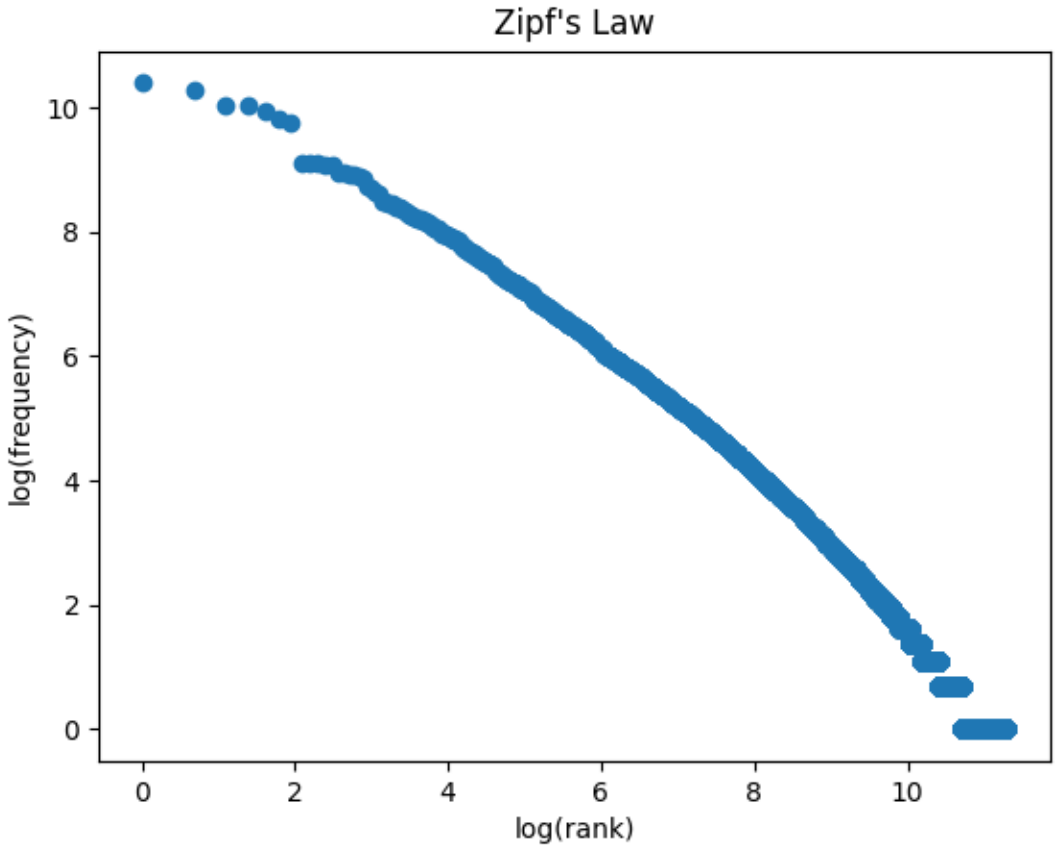
שלב 2 – מימוש חוק zipf

2) הגרף מייצג את לוג הדרגה מול לוג התדירות של הטוקינים(נתייחס רק למילים), דרגה של מילה זה הדירוג של מילה ביחס למספר ההופעות שלה: דרגה 1 אומרת שהמילה הזו היא המילה שיש לה הכי הרבה הופעות הטקסט וכו..
התדירות זה כמה פעמים כל מילה הופיעה, לכן הגרף מייצג את הקשר הלוגרתמי בין דרגות המלים לתדירות שלהן.

3) כן הגרף תואם את הצפיות שלנו, כי לפי החוק הגרף יהי כמעט קו ישר – והגרף שלנו מקיים את החוק והוא קרוב מאוד לקו ישר.

4) אם היינו מקטינים את גודל הקורפוס יתכן שנקבל יותר אי-סדר בגרף אבל לרוב עדיין נקבל משהוא שדומה לקו ישר במיוחד באמצע הגרף, אם הקורפוס גדול יותר אז הגרף יצטבר יותר לקו ישר.

5) תמונה של הplot:



6) כן המלים תומאות את הצפיות שלנו, המלים הכי נפוצות הן לרוב מילות קשר,המילה "הכנסת" כמובן שהיא תהיה בינם. לגבי המלים הלא נפוצות שמופיעות פעם אחת הן מלים ששומעים אותם פעם בשנה 😂

```
Top 10 words with highest frequency:
   32521 את
   29425 לא
   23010 של
  22793 אני
   20644 על
   18489 זה
17440 הכנסת
   9017  גם
  8918  חבר
  8869  הוא

Bottom 10 words with lowest frequency:
      1 שמורשת
     1 שנואמים
     1 מתפקודו
      1 בהשתקת
     1 ברודנות
   1 סירקולציה
 1 שהסירקולציה
1 מההתייחסויות
  1 ומהתפטרותו
       1 נתלית
```

עוד החלטות שקיבלנו במהלך העבודה על התרגיל:

בכתיבה לקובץ CSV היה לנו משהן מוז שבכתיה היה שיכפול ל סימן " , אחרי כמה נסיונות לתקן השכפול הזה היה מקורו שהסימן הזה הוא ה –qoutechar של הקובץ CSV ,והיה אפשר לשנות אותו לכן בחרנו ב $ להיות ה-qoutechar , וגם היה צריך להודיע לפונקציה כשקוראים את קובץ ה-CSV .
שורת הכתיבה:

```
df.to_csv(output_csv_path, index=False, encoding='utf-8',
quoting=csv.QUOTE_MINIMAL,quotechar="$")
```

שורת הקראיה :

```
df = pd.read_csv('output_data.csv', quotechar="$")
```