**שלב 1 – טיפול בטקסט:**

2. שליפת טקסט בעל תוכן:

לקחת את הטקסטים הרלוונטיים עבורנו כך:  
כל עוד לא ראינו מישהו מדבר\* אז מדלג על הטקסטים, אחרת לקחנו את הטקסט.  
את הכתרות שיש באמצע (או אלה שאינם שייכם לאף דובר) שמנו אותם בדובר האחרון

דרך המימוש:

1. Class sentence ששומר את השם של הדובר ואת כל הטקסט שהוא אמר
2. Class protocol ששמור את השם של פרוטוקול ואת המספר וסוגו ורשימה מסוג Sentence .

שליפת שמות הדוברים:

1. אם זוהה דובר דרך הרשימה [דובר, אורח, יור, דובר-המשך, קריאות].
2. אם המפשט אינו מודגש וגם יש לו קו וגם התו האחרון הוא : או < או >.
3. אם המפשט הוא מודגש וגם יש לו קו וגם התו האחרון הוא : או < או > וגם הSTYLE הוא head or plain.
4. אם המפשט יש לו קו וגם התו האחרון הוא :.

סיבות להחלטות:

* ההחלטה בלקחת את הדברים הרלוונטיים (כלומר רק דיבורים):
  + כל מה שיש בתחילת הקובץ אינו רלוונטי עד שנראה מישהו מדבר ואז נתחיל לראות את הדיבורים וכל הטקסטים הרלוונטיים.
  + את מה שנמצא באמצע כלומר כתורת ודברים כאלה שלא רלוונטיים בחרנו שיהי בדובר האחרון משום ש אם אנו ננסה לנקות את אלה גם אנחנו עלולים לפספס דיבורים ודברים רלוונטיים ולכן הקורפוס עלול לקטון ולא ייצג את ב100% את הקבצים.
* ההחלטה בלקחת את הדוברים:
  + רוב פרוטוקולים שהם מסוג מליאה אנחנו נזהה את הדובר בלי בעיה מהרשימה שיש למעלה.
  + ושאר הפרוטוקולים אנחנו נזהה אותם כך: לאחר דגימה של כמות קבצים ראינו שרוב הדוברים לא מודגשים וגם יש להם קו ויש בסוף : או > לא כמו שאר המילים הלא רלוונטיים ולכן כך נוכל לזהות בקולת את השם של כל דובר כך.
  + ויש כמה מקרים חרגים כמו שהם מודגשים וגם יש להם קו ומסוג מסויים ולכן טפלנו במקרים כאלה.

1. ניקיון שמות:

לאחר שדגמנו כמות של קבצים ראינו שאם יש סוגרים "()" אזי זה אומר שם מפלגה ולכן אני מוחק אותה ולקוח כל מה שיש אחריה ואם אין אזי (בקובץ מסוג ptm) אזי יש תפקיד או שר או מזכיר ותפקדים מסוג זה אז רק לוקחים את שתי שימות האחרונים.

דרך המימוש:

* + רשימה קצרה של תפקדים אם מצאנו אחד מהתפקידים הללו בתוך השם שיש בידנו אזי לוקחים את שתי מילים האחרונות.
  + אם הקובץ מסוג PTM ואין לו סוגרים "(" אזי לוקחים את שתי המילים האחרונות (כי אז יש לו תפקד כמו שלב הקודם).

אחרת אנחנו לוקחים את כל המילים שיש לנו כשם, ונעשה להם ניקיון ע"י split() לפי ( כדי לזרוק את כל התפקיד שיש בתוך ה(...) וגם נעשה split() לפי < ועוד פעם לפי > כי יש שמות שמתילים ב < ו מסתיימים ב > ולכן כך אנחנו זרקנו את התפקדים שמתעטפים בסימונים כאלה, ואם יש מילה שיש בה " אז זהו קצור לתפקיד כמו ע"ד או יו"ר שזה בא בתחילת השם בה אנחנו עוצרים ומחזרים השם שיש בידינו (בלי tab or enter ).

בעיות יכולת להיות שהיה קשה להגבר עליהן :

* + יש בעיה בקובץ מסוג ptm כאשר יש כמה סוגים של תפקדים ולא נוכל להכליל את זה כי זה יוכל להשתנות ולכן יכול להיות יש שם מלא מאורך 3 או יותר ויש לו תפקיד (למשל סגן של מישהו..) ולכן לפי השיטה נלקח רק את השניים האחרונים (ואז נפספס את המילה הראשונה את קיימת בשם).
  + יש עוד בעיות בדובר כמו תשובה של "שם של הדובר" לשר המפשטים או דברים מסוג הזה לא נוכל לזהות איפה השם נמצא במדויק.
  + יש דוברים שאין להם בסוף :

1. בעיות לפני:

* אנחנו עלולים לחשוב שהמפלגה או שם של תפקיד שיכול לחזור על עצמו הרבה פעמים היא מילה חשובה וקיימת בכל מקום.
* כך אנחנו נזכיר שם של מישהו באופן ספציפי מאוד.

בעיות אחרי:

* אנחנו עלולים לפספס חלק מהשם עצמו כי אנחנו לא יכולים לנקות באופן מושלם.
* עלולים לאבד מידע חשוב עבור השימות האלה שהם יכולים גם להיות דרך של זיהוי.
* לא נוכל להגיד מה הדובר יגיד או באיזה נושאים הוא ידבר.

(הערה: לא הבנתי את השאלה באופן מדויק אבל כתבתי למעלה את מה שהבנתי ומעל גם כתבתי את הבעיות של הנקיון...).

3. חלוקה למשפטים:

הפרדת משפטים הייתה לפי ., !, ?, : ו; כלומר אם אנו ראים אחד מהם אזי זה סימן לסוף משפט.

* בשפה הערבית נקודה היא רק לסוף משפט.
* ! הוא גם סימן לסוף משפט.
* ? הוא גם סימן לסוף משפט.
* : הוא בא לפני ציטוט או דיבור ישיר, לפני פירוט שקדם לו ביטוי מכליל.
* ; בין משפטים שמבחינה תחבירית הם עצמאיים.

ההחלטות:

* אנו יודעים ש אם יש נקודה אזי זהו סוף משפט חוץ מהמקרה כאשר יש הנקודה בין שני מספרים או 1. אזי לא מפרדים אותם לפי נקודה.
* ! הוא סימן לסוף משפט לפי השפה גם ? .
* לגבי : מה שבא אחריו זה ציטוט אל פירוט לכן הוא מעיד על סוף משפט.

4. ניקיון המשפטים:  
הניקיון היה לפי ביטוים רגולריים:

* ואז אם יש משפט יש בה מילה בשפה זרה אזי מוחקים אותה.
* משפטים שמכילים תווים או מספרים ואין בהם אותיות.

בדיקות ייחודיות:

* משפטים שנקטעים באמצע ע"י - - - .
* מפשט הוא לפחות 4.

איך התמודדנו בלנקות משפטים:

* כל משפט שיש בו --- ברצף (אחרי שחלפנו את הרווחים במחרוזת ריקה) אזי מוחקים אותה כי שלושה הקווים האלה יכולים להיות באמצע משפט ולכן גם משפט זה אינו חוקי.
* לגבי מילה זרה אנחנו בדקנו את כל המשפט ע"י ביטוי רגולרי וגם לגבי תווים לא חוקים (כלומר מחרוזת שלא מכילה אותיות בעברית).
* הפרדנו כל משפט ע"י רווחים ובדקתי אם יש פחות מ 4 מילים ואז מחקתי.

5. טוקניזציה:

כל מילה היא token בחרנו בדרך הזו כי:

* כך נדע שכל מילה יש לה משמעות פני עצמה.
* כל צורה של מילה תהיה במאגר שלנו כך גם נרוויח את הצורות המורפולוגיות שלה.
* זה יעזור לנו אם מילה נכתבה בצורה לא נכונה אז נוכל לזהות את זה בקלות.

ראשי תיבות: החלטנו להשאיר אותה כמו שהיא.

* אחרת לא יהי משמעות בכלל אם נפריד בניהם.

קיצורים: ניקח אות כל הקיצור עם הסימן ' כtoken .

* אחרת לא היה משמעות בכלל אם נפריד בניהם.

Hyphenation: אם היה בסוף מילה אז מחקנו את ה – וחיברנו את המילה אחרת אנחנו נחליף את – ברווח.

* אם הוא מחבר מילה אז אנחנו חיברנו את המילה ומחקנו אותה ולכן לא צריך אותו.
* אם הוא מחבר שני מילים לא צריך אותו משום שעדיין אנחנו שומרים על המשמעות של המילים שהיו מחוברות.

ציטוט: את ה סימן הוא token בפני עצמו וכל מש שיש בפנים נפריד לפי הכללים שנ"ל.

* אנחנו מעוניינים במילים שיש בציטוט וכפי שהברנו קודם כל מילה היא token מהסיבות שנ"ל
* כל מה שיש בתוך הציטוט הוא אוסף של מילים ולכן אין הבדל בינו לבין דיבור רגיל זה לא יפריע לנו אלא להיפך.

סימנים ( ) : כל אחד מהם הוא token בפני עצמו כי הוא מכיל מילים ואלה אנחנו רוצים להפריד לפי הכללים שנ"ל.

* משום שיש בתוך ה ( ) מילים ואנחנו לא רוצים ש ( ) יפריע לנו אזי אנחנו נטפל בה כמקרה חריג.

לגבי סימני פיסוק " : ", " ; ", " , ", " . ", " ! " ו" ? ":

כפי הוחלט בתרגיל בית הם מקרים חרגים וכל סימן הוא token .

**שלב 2 – מימוש חוק zipf :**

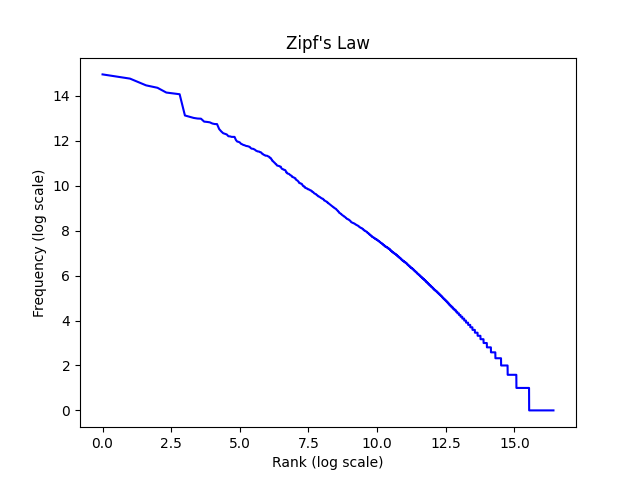
2. הגרף מציג לנו איך המילים מתנהגים בקורפוס ובפרט מה הקשר בין שכיחות המילה לבין הדרגה שלה בסדר לפי שכחיות. גרף כזה חשוב מאוד לתפיסת תופעות בשפה הטבעית.

3. כן, לפי חוק ZIP יש יחס קבוע בין log(frequency) ו log(rank), וזה מה שיצא לנו.

4. ההתנהגות של הגרף לא אמורה להשתנות מדי אם אנו מגדילים או מקטנים את הקורפוס, כך שחוק ZIP מתקיים בכל קורפוס לא משנה מה גודל הקורפוס. כדי ש ההתנהגות של הגרף תשתנה בצורה משמעותית נחפש מאגר שמייצג הרבה קיצורים ומילים שונות כמו ב RETURES .

אבל, אם הקורפוס יהיה מאוד קטן למשל, ה"מדרגות" בגרף יהיו יותר גדולות כך שבקורפוס קטן תופעה נדירה בשפה יש לה משקל יותר גדול.

5.



6.

כן, המילים עם התדירות הכי גבוהה , רובם מילים מאוד נפוצות בשפה העברית. עבור המילים "חבר" , "הכנסת" גם היה צפוי שיהיו עם תדירות גבוהה בגלל שהמאגר שלנו לקוח מפרוטוקולים של מליאות וועדות הכנסת. אותו דבר לגבי המילים עם תדירות הנמוכה ביותר אילו מילות לא היינו מצפים שיהיה להם תדירות גבוהה בשיחות של חברי הכנסת.

|  |  |
| --- | --- |
| top 10:  ('את', 31682)  ('לא', 27947)  ('של', 22648)  ('אני', 21615)  ('על', 18149)  ('הכנסת', 17467)  ('זה', 17401)  ('חבר', 8997)  ('גם', 8603)  ('הוא', 8323) | lowest 10:  ('שנתאם', 1)  ('טעימה', 1)  ('נצלול', 1)  ('משויפים', 1)  ('דילגנו', 1)  ('מתאוששת', 1)  ('מעוקל', 1)  ('הטלה', 1)  ('319טז', 1)  ('מגובש', 1) |