***תרגיל בית 1 :*** שם : ניב ספיר , ת.ז : 308315944

***מבנה של הקורפוס – חלק א' :***

\* המוטיבציה העיקרית מאחורי הפרדיגמה הזאת היא שאנחנו לא תמיד מסוגלים לנסח כהלכה את החוקים שפועלים בשפה טבעית אבל החוקים האלה באים לידי ביטוי בטקסט. קורפוסים משקפים ידע לשוני, גם אם הידע הוא לא מפורש הוא מובלע בתוך הקורפוסים. המהפכה פתחה 2 ענפים בלתי תלויים של מחבר: בלשנות מבוססת קורפוס: הבלשן יכול לחלץ את חוקי השפה מתוך טקסטים ולא מתוך משפט בודד. עד כמה תבנית אחת נפוצה בשפה. בניית מערכות שיכולות לאמוד באופן סטטיסטי פרמטרים מתוך קורפוס שיאפשרו לנו לבנות מערכות עיבוד שפות טבעיות שההיקף שלהם הרבה יותר רחב, שיש להם גישה לידע רחב. הקורפוס צריך להיות מייצג מכיוון שאנו אומדים פרמטרים מתוכו. מייצג משמע דומה לטקסטים אמיתיים שאיתם המערכת תצטרך לעבוד. לשם כך צריך לחשוב טוב איזה טקסטים רוצים לשלב בקורפוס ואיזה לא. צריך לחשוב איך ה data מיוצג )asci מתאים לאנגלית ולא לשפה אחרת), היום משתמשים ב Unicode , utf-16 ו . utf-8 ככל שהקורפוס יותר גדול עם יותר נתונים ככה הקורפוס טוב יותר – זה לא תמיד נכון לפעמים מידע יכול להרע.

\* למה נרצה לייצג קורפוס בקבצי xml ?

את הקורפוסים נשמור ב xml – משליטים מבנה על הטקסט. הM מייצג את המילה markup והכוונה לכך שיהיה ברור מהו המבנה של הטקסט , למשל היכן נגמר משפט, איפה נגמרת מילה ועוד.. נראה כי בקבצי xml נוכל לשמור עוד פרטים על ה דאטה באופן מסודר כמו pos ועוד..

\*בעיות תכנון :

הרבה פעמים בטקסט שמקבלים יש כבר תגים של פורמט או markup ,לדוגמא: אם קיבלנו מסמך בוורד יש תגים של b או i .אם קיבלנו מסמך מבית משפט יש הערות של מיקום ושעה שלא שייכים לטקסט ולכן צריך לזהות את הדברים הללו ולנקות אותם. יש המון לא-טקסט בתוך טקסט. למשל: אם הורדנו מאמר מהעיתון יש תמונות, מדיה ופרסומות. סריקה של ספרים ישנים מקבלים תמונה גרפית שצריך להפוך אותה לטקסט (OCR) והרבה פעמים בתהליך זה יש שגיאות, למשל י' במקום ו'. באנגלית יש Lowercase ו Uppercase , האם אות גדולה היא בשל התחלת משפט או שם פרטי. שמות פרטיים , יש המון דרכים שונות לכתוב שמות פרטים, איך יודעים שמתייחסים לאותו בנאדם?

***חלק ב' + ג' :***

לאחר קריאת והבנת קבצי ה xml בניתי את הקלאסים הרלוונטיים : token, sentence, corpus. על כל קלאס מוסבר בהרחבה (בתוך הקוד ע"י הערות) על בחירת השדות בכל הקלאס וייצוגם.

נראה כי חוק Zipf מתקיים על קבצי ה xml. אחרי קריאת הקבצים והכנסתם לקלאסים המתאימים, יצרתי מילון תדרויות של מילים אשר מאשש את חוק Zipf : 

נראה כי מילים כמו the,of וכו'.. מופיעות בתדירויות גבוהות (the מופיע 21,197 פעמים מתוך הקבצי xml שקראנו) כמו שחוק zipf טוען. לעומת מילים בתדרויות יותר נמוכות : 

***נקיון רעשים :***

1) כותרות : הכותרות היו מיוצגות ע"י '==' או יותר. כדי להבדיל בין הכותרות בתוך הקורפוס החלטתי להשאיר '=' אחד לפני ואחרי כל כותרת על מנת ליצור הפרדה בין כותרת לטקסט רגיל בתוך הקורפוס.

ככה נראה כותרת בקורפוס אחרי ניקיון : 

בלי השארת '=' אחד מכל צד הכותרות היו יכולות להיראות כמו משפטים חסרי הקשר לטקסט בתוך הקורפוס ולכם החלטתי להשאיר '=' אחד מכל צד.

אחרי כל כותרת בקורפוס ירדתי פעמיים שורה על מנת להדגיש את חשיבות הכותרת. כותרות יוצאת מין הכלל הן כותרות ריקות (כותרות ללא טקסט אחרי הכותרת או כותרות משניות). כותרת כאלה ללא תוכן אחרי , או מספר כותרות רצף ייוצגו גם עם '=' אחד מכל צד תתבצע בקורפוס רק ירידת שורה אחת בין כל כותרת (ולא שתיים להבדיל מכותרת רגילה).

2) ניקיון של נוטציות מספרים ליד הערכים שלהם.

3)יותר מרווח אחד הפכתי לרווח אחד. למשל tab ועוד..

4)החלפתי שורות רווח (שניים או יותר) לשורת רווח אחד.

***חלוקה למשפטים :***

אחרי הניקיון נפרק את הטקסט למשפטים. החלטתי לחלק את הטקסט למשפטים לפי ה delimiters הבאים : נקודה '.' , סימן קריאה '!' , סימן שאלה '?'.

הטקסט מחולק כך :

***\****כותרת : כותרת היא משפט בפני עצמו.

***\****נקודות רצופות : במקרה זה כמו למשל (ועוד.....) נרד שורה רק בסוף רצף הנקודות. כלומר משפט יסתיים רק בסוף הרצף.

***\****ציטוט : בסוף ציטוט אם אנחנו בסוף משפט ע"י אחד התנאים שבחרתי למעלה אזי נרד שורה וניקח את זה כמשפט.

נשים לב למקרים מיוחדים בהם אין משפט חדש וצריך להיזהר מהם :

1. כאשר יש נקודה בין מספרים . למשל נקודה עשרונית.
2. בקיצורים של אותיות באנגלית למשל e.g. ועוד..
3. קיצורי מילים , למשל Mr. ועוד..

הערות חשובות : אל נקודה-פסיק (';') ונקודתיים (':') התייחסתי כמו אל פסיק. כלומר הם לא מהוות סיום משפט וירידת שורה. לדעתי ברוב המקרים הסימנים האלה משמשים יותר למטרת עצירה קצרה ולא למטרת סיום משפט.

***טוקניזציה:***

לאחר פירוק הטקסט למשפטים, חילקתי כל משפט ל tokens לפי ההחלטות הבאות:

עבור כל אחד מהתווים הבאים: \)| \(|;| \}| \{| \ ]|\ [|>|<| \+| \^| \ %|\#|=|"|\|( כאשר הם מופיעים ליד מספר או מילה כלשהי, הפרדתי ל tokens נפרדים.

את הטוקנים '.' ',' ':' '/' ' \' הפרדתי לטוקנים נפרדים כאשר לא באו בין שתי ספרות:

\* נקודה יכולה להיות בין שתי ספרות, למשל נקודה עשרונית, תאריך וכו''.

\* נקודתיים יכולים להוות יחס בין מספרים.

'/' '\ ' \*יכולים להיות חלק מתאריך.

עבור פסיק, כאשר יש לפניו בדיוק שלוש ספרות הספקתי שהוא חלק ממספר ואין צורך להפריד אותו לטוקנים שונים, למשל במקרה הבא: 111,111,1 .

גרשיים (") החלטתי לפרק לטוקן. כלומר מקרים בהם יש מילים בעברית כמו בס''ד, אזי זה יתפרק ל בס " ד.

גרש החלטתי לא לפרק לטוקן נפרד. בהרבה שפות לטיניות הוא מופיע כחלק מהמילה ולא מסמל קיצור. גם בעברית למשל יש מילים רבות כמו צ'יפס, מאסז ' וכו' שהוא חלק מהמילה עצמה והורדתו יכולה להשפיע על משמעות המילה.

מקף (' -') החלטתי לא לפרק ל tokens נפרדים. ההפך הרבה פעמים '–' מהווה חיבור בין מילים אשר יוצרות טוקן אחד. למשל בית-חולים.

לסיכום כל משפט הורכב מהרבה טוקנים לפי התנאים למעלה נשים לב שכל מילה היא טוקן אלה אם כן המילים מחוברות כמו הדוגמא למעלה : בית-חלים אשר מהווה טוקן אחד.