**עיבוד שפות טבעיות**

**תרגיל בית 2**

**שם: רנים אבראהים (212920896), אסיל נחאס (212245096)**

**מבוא:  
במהלך התרגיל התבקשנו לבנות וליישם מודלי שפה המבוססים על טריגרמות קורפוס שנלקח מפרוטוקולים של הכנסת. המטרה היא לחשב הסתברויות הופעת משפטים וטוקנים לפי מודל סטטיסטי המבוסס על תדירות הרצפים בקורפוס, וכן לאתר קולוקציות נפוצות ולבצע משימות השלמה וחיזוי טוקנים חסרים במשפט.**

**הקורפוס מחולק לשני סוגים: משפטים שמקורם בפרוטוקולים מסוג "ועדה" ומשפטים שמקורם בפרוטוקולים מסוג "מליאה". בהתאם לכך, נדרשו שני מודלי שפה נפרדים: אחד לוועדות ואחד למליאות.**

**שלב 1:**

**בניית מודלי השפה:  
בשלב זה התבקשנו לבנות מודלים המבוססים על טריגרמות. רעיון הטריגרמה הוא שחיזוי הטוקן הבא במשפט מתבסס על שני הטוקנים הקודמים לו.**

**Trigram\_LM : מחלקת  
בנינו מחלקה המאגדת את כל המבנים והפונקציות הדרושים למודל השפה: שמרנו במבנה נתונים את ספירות היוני-גראמס (טוקן יחיד), הבי-גראמס (זוג טוקנים רציפים) והטרי-גראמס (שלישיות טוקנים רציפות) עבור כל אחד משני סוגי הפרוטוקולים: ועדות ו- מליאות.**

**עקבנו אחר מספר הטוקנים הכולל וספירת כל מילה, כדי שנוכל לחשב הסתברויות בהמשך.**

**הוספנו שני טוקני דמה בתחילת כל משפט, וכן טוקן סיום על מנת לטפל במקרים של תחילת המשפט וסופו לפי הדרישות.**

**וגם הגדרנו את הלמדות שיהיו המשקל של כל n-gram כך שבמהלך חישוב ההסתברויות של ה- ngrams נרצה לתת יותר משקל ל- trigrams ביחס ל unigrams / bigrams כי רצף של 3 מילים יכול להגיד לנו יותר על המילה הבאה ברצף משתי מילים או ממילה אחת.**

**בנינו פונקציה שמחשבת את כמות הטוקנים וגם מצרפת להם טוקני הדמה הפונקציה גם מחשבת את כמות ה- ngrams. הפונקציה נקראת fit\_model\_to\_sentences.**

**במחלקה שלנו גם קיימות הפונקציות הבאות:**

**calculate\_prob\_of\_sentenceפונקציית:**

**פונקציה זו מקבלת משפט ומחזירה את לוג ההסתברות שלו לפי המודל.**

**הוספנו למשפט את הטוקנים הדמה בתחילתו ובסופו.**

**עברנו על כל טוקן במשפט (החל מהטוקן השלישי) וחישבנו את ההסתברות שלו בהתבסס על שני הטוקנים הקודמים.**

**ההסתברות חושבה בשיטת נראות מקסימלית עם החלקת לפלס כדי למנוע הסתברויות אפס, ועשינו את החישוב הזה בעזרת הפונקציה compute\_smoothed\_probability  
כך שהפונקציה בודקת על פי איזה ngram אנחנו מחשבים את ההסתברות ומציבה בנוסחה שלוקחת את הכמות של ה- ngrams ומוסיפה 1 ומחלקת בכמות הכללית ועוד גודל השפה.**

**השתמשנו באינטרפולציה ליניארית: שילבנו את הסתברות היוני-גראם, הבי-גראם והטרי-גראם תוך שימוש במשקלים שנבחרו. בתחילה תכננו משקלים שונים, אך על פי ההנחיות ניתן לבחור משקלים ולהסביר בדו"ח. בדוגמה שלנו בחרנו משקלים כך שהטריגרמה מקבלת משקל גדול (0.7), הביגרמה משקל קטן (0.01) והיוניגרמה משקל בינוני (0.29), במטרה להסתמך יותר על ההקשר הקצר (טוקן אחורה) והסיבה לכך היא כי לפעמים רוצים להסתמך על הקונטקסט ולכן נותנים משקל גבוה ל-trigrams ולפעמים רוצים להסתמך רק על מילה אחת כי אנחנו משתמשים במודל של מליאות על משפטים של וועדות בהמשך.**

**לבסוף סכמנו את לוג ההסתברויות של כל הטוקנים במשפט וקיבלנו לוג הסתברות כוללת למשפט.**

**generate\_next\_tokenפונקציית:**

**פונקציה זו מקבלת צירוף של טוקנים וחוזה את הטוקן הבא.**

**בדקנו אם צריך להוסיף טוקני דמה ובחישוב של ההסתברות לטוקן שרוצים לבחור אותו התעלמנו מטוקני הדמה**

**גם פה חישבנו הסתברות לכל מילה במילון לפי טריגרמה, בי-גראם ויוני-גראם בתוספת החלקת לפלס שמחשבים בעזרת הפונקציה compute\_smoothed\_probability שהזכרנו קודם.**

**בחישוב ההסתברות המקדמים שבחרנו הם אותם מקדמים שהגדרנו במחלקה lambda\_trigram = 0.7) (Lambda\_bigram = 0.01, lambda\_unigram = 0.29 כך שנתנו משקל יותר ל- trigrams כדי לקבל תשובה מדויקת יותר.**

**לאחר חישוב ההסתברויות עבור כל טוקן, בחרנו את הטוקן בעל ההסתברות הגבוהה ביותר והחזרנו אותו יחד עם לוג ההסתברות שלו.**

**שלב 2:**

**קולוקציות:**

**בשלב זה התבקשנו להחזיר את 10 הקולוקציות הנפוצות ביותר באורכי 2,3,4 בכל אחד משני סוגי הקורפוסים (ועדות ומליאות), לפי שני מדדים: (frequency) ו-TF-IDF.**

**לכן כדי לעשות כך נצטרך לממש פונקציה שמחזירה לנו את k הקולוקציות באורך n הכי נפוצות בקורפוס על ידי מדד מסוים. לכן מימשנו את הפונקציה הבאה:**

**get\_k\_n\_t\_collocationsפונקציית:**

**חילקנו את המשפטים לפי סוג הפרוטוקול (committee, plenary) :**

**יצרנו n-grams מהמשפטים לדוגמה: (2-grams, 3-grams, 4-grams)**

**אם המדד הוא "frequency", סיננו רק את ה- ngrams שמופיעות לפחות t פעמים, מיינו אותם לפי התדירות בסדר יורד ולקחנו את ה- k המובילים.**

**אם המדד הוא "tfidf", חישבנו TF-IDF עבור כל ngram, סיננו לפי הסף t, מיינו לפי ערך TF-IDF, ולקחנו את k הגבוהות ביותר.**

**את התוצאות הדפסנו לקובץ knesset\_collocations.txt בפורמט הנדרש.**

**תוצאות לדוגמה מתוך הקובץ :knesset\_collocations.txt**

**ניתן לראות בקובץ כי הקולוקציות הנפוצות ביותר (במדד תדירות) כוללות המילים "אני" "זה" וכן מילים מפתח כמו "חבר הכנסת", "היושב - ראש", "אדוני". זה מצביע על כך שבקורפוסים יש הרבה תבניות טקסטואליות שחוזרות (כמו קריאות ביניים, שמות דוברים, סמלי פורמליות).**

**במדד ה-TF-IDF ניתן לראות הבדלים גדולים: הקולוקציות עם ערכי TF-IDF גבוהים הן עדיין קולוקציות חוזרות, אך המיקום שלהן מבחינת TF-IDF מעיד על כך שהן "ייחודיות" יחסית בתת-קורפוס מסוים.**

**בפונקציה get\_k\_n\_t\_collocations השתמשנו בפונקציות עזר כמו calculate\_ngram\_frequencies שמחשבת את כמות ה- ngrams במשפטים, ואנחנו נשתמש בערך הזה כדי לחשב כדי לחשב את ערך ה tfidf, ולחישוב ערך ה- tfidf נשתמש גם בפונקצית עזר compute\_tfidf שמחשבת הערך על ידי הצבת הארגומנטים המועברים לפונקציה כמו ngram, ngram\_frequencies, document\_frequencies, num\_documents בנוסחת המדד ולאחר החישוב הפונקציה מחזירה את התוצאה.**

**שלב 3:**

**יישום מודלי השפה:**

**mask\_tokens\_in\_sentences :פונקציית**

**בחרנו אקראית 10 משפטים מתוך קורפוס הוועדות, כל משפט באורך של לפחות 5 טוקנים. מסכנו 10% מהטוקנים בכל משפט (לפחות טוקן אחד) בעזרת "[\*]". את המשפטים המקוריים כתבנו לקובץ original\_sampled\_sents.txt ואת הממוסכים כתבנו לקובץ masked\_sampled\_sents.txt.**

**ניבוי הטוקנים החסרים (generate\_results) :  
לאחר שמסכנו את הטוקנים, השתמשנו במודל המליאה (plenary) לחזות מה הטוקן החסר.  
למשל, אם המשפט היה "אדוני [\*] אתה שאלת , אני אענה*", המודל ניסה לנבא את המילה במקום "[\*]*" בקובץ התוצאות sampled\_sents\_results.txt ניתן לראות שחזה ",", כנראה כי "אדוני , אתה שאלת , אני אענה" הייתה אופציה שהמודל העריך כבעלת הסתברות גבוהה יחסית.**

**כמו כן, חישבנו את ההסתברות של המשפט המשוחזר בכל אחד משני המודלים (מליאה וועדה) והדפסנו את התוצאות. ניתן לראות שהסתברויות הלוג משתנות, ולעיתים המשפט מתאים יותר למודל המליאה ולעיתים למודל הוועדה.**

**חישוב Perplexity:  
חישבנו את ה- Perplexity עבור הטוקנים הממוסכים בלבד Perplexity .הוא מדד לאיכות המודל: ככל שהוא נמוך יותר, כך המודל מתאים יותר לנתונים.  
בקובץ perplexity\_result.txt התקבל ערך גדול מאוד (12890.42). משמעות הדבר היא שהמודל, במימוש הנוכחי לא מנבא היטב את הטוקנים הממוסכים. זה צפוי, שכן המודל הוא די בסיסי. ראינו עם הרבה ניסויים גם שככל אנחנו משנים את הלמדות ונותנים משקל יותר גבוה ל trigrams קיבלנו perplexity יותר טובה אבל הגענו למצב שבו הלמדות החלו להשפיע באופן רע על ה perplexity.**

**הנוסחה שהשתמשנו בה היא: 2^H כך ש: H = 1/N \* log(P(W1,…,Wn).**

**הסבר על הבחירות והמשקלות:  
בחרנו משקלים (λ1 = 0.7, λ2 = 0.01, λ3 = 0.29) המעדיפים באופן כבד את הטריגרמה, בתקווה שהקונטקסט הרחב יותר יסייע לחיזוי מדויק.**

**גם ראינו שכאשר השתמשנו בלמדות שנותנות העדפה פחות ל trigrams קיבלנו ערך perplexity גדול יותר ממה שקיבלנו עם הבחירה של lambda\_trigrams = 0.7 וגם כאשר נסינו לתת משקל יותר גבוה, כלומר lambda\_trigrams = 0.99 מצאנו שגם ערך ה perplexity עלה, לכן מפני המשקלים שבחרנו נתנו התוצאה הכי טובה דרך trial and error, החלטנו לקחת אותם להיות הבחירה שלנו.**

**סיכום:  
בתרגיל זה בנינו מודלי שפה טריגרמיים עבור פרוטוקולים של הכנסת, למדנו להעריך הסתברויות למשפטים ולטוקנים, ולאתר קולוקציות נפוצות לפי תדירות או TF-IDF. ביצענו גם משימות מסכה וניבוי טוקנים חסרים, וחישבנו Perplexity כדי להעריך את ביצועי המודל. התוצאות מראות כי למרות שהמודל יכול לנבא טוקנים ולהציג הסתברויות, ההוספה של ענישות וחלוקת משקלים קיצונית ייתכן ופגעה באיכות החיזוי. בנוסף, הקולוקציות הנפוצות מראות את המבנה החוזר והפורמלי של הפרוטוקולים. כנראה שמודל מתוחכם יותר, או שימוש במשקלים מאוזנים יותר, עשויים לשפר את ביצועי המודל ולהפחית את ה-Perplexity.**