מוחמד עראבי 206985533

ג'ון פייר חדאד 316379999

דו"ח

תרגיל בית 2

שלב 1:

בנינו המחלקה Trigram_LM שבתוכה המשפטים וגם trigrams,bigrams,unigrams ועוד שני משתנים vocabulary and num all tokens שנשמור עם המודל בלי צורך לחשב בכל פעם, ומתודה התחלתית build_the_model() כדי להגדיר ולבנות המודל.

בשביל לבנות או להגדיר trigrams,bigrams,unigrams בצורה מתוחמת ויותר נוחה כי זה מבוסס ספירה השתמשנו ב-defaultdict ו Counter לפי הצורך (גם בשאר הפונקציות הגדרנו מילונים מסוג הזה) וגם השתמשנו בו ובשאר הפונקציות ב-set() כאשר רצינו להגדיר משהו ייחודי.

בנינו המתודה calculate_prob_of_sentence(self, sentence, smoothing_type) שתחשב לנו לוג ההסתברות של המשפט לפי סוג המדד שתקבל כקלט, כך השתמשנו ב- trigrams,bigrams,unigrams שהגדרנו להמודל. ולגבי החישוב לפי המדד Linear בחרנו בגנות 0.7 , 0.2 , 0.1 כך 0.7 בשביל trigram ו-0.2 בשביל bigrams, הבחירה הייתה כך ה-trigram יקבל המשקל הכי גדול כי בעצם המודל והעבודה מבוסס trigram, ואז משקבל יותר קטן ל-bigrams שגדול ממשקל ה-unigram כדי יהיה לו יותר חשיבות.

גם בנינו המתודה generate_next_token(self, sentence) שמחזירה התוקן הכי סביר להמשך המשפט שנקבל כקלט לפי ההסתברות של המשפט עם הטוקן הזה כלומר קוראת לפונקציה calculate_prob_of_sentence(self, sentence, smoothing_type) בשביל חישוב ההסתברות.

במתודות בשלב הזה גם הוספנו את שני טוקני דמה <s_0><s_1> לפי המבוקש ולפי הצורך.

שלב 2:

בנינו הפונקציה get_k_n_collocations(self, k, n, copus, measure_type) כמתודה למחלקה שבנינו, כך בהתחלה ספרנו בתוך מלון חדש שהגדרנו את הקולוקציות בגודל n, ואז עבור המדד frequency פשוט נבחר הקולוקציות עם המספר ספירה המקסימאלי, ובשביל המדד tfidf השתמשנו בנוסחה : tf הוא מספר הפעמים ה-term מופיע במשפט חלקי מספר המילים שיש במשפט

ו-idf הוא מספר המשפטים בקורפוס חלקי מספר המשפטים ש-term מופיע בהם.

ובשני מקים אחרי שנעשה מיון יורד (reverse=True) פשוט נחזיר ה-k הראשוניים ברשימה.

ובשביל הכתיבה לקובץ כתבנו פונקציה

write_colocations_file(output_path, file_name, committee_model, plenary_model)

שמקבלת נתיב לקובץ שאלו נכתוב והשם של הקובץ וגם שני המודילים ובתוך הפונקציה נקראה למתודה

get_k_n_collocations(self, k, n, copus, measure_type) ונדפיס לפי המבוקש.

שלב 3:

בשביל השלב הזה הגדרנו שתי פונקציות :

- complate_masked_sentences(model, sentences) שמקבלת מודל והמשפטים שקראנו מתוך הקובץ masked_sentences.txt כך עבור כל משפט בכל פעם נמצא [*] נקרא לפונקציה get_next_token ונתן לה המשפט עד לפני [*] שראינו והיא תחזיר לנו הטוקן הכי סביר וככה נמלא החסר במשפט. והפונקציה הזאת תחזיר לנו בסוף רשימה של המשפטים אחרי שהשלמנו וגם רשימה של הטוקנים שקבלנו במהלך התהליך בשביל ההדפסה אחר כך.

- committee_or_plenary(committee_probability, plenary_probability) שפשוט תקבל ההסתברויות של המשפטים תחזיר ותבחר מה הסוג הכי סביר מבין השנים שהמשפט יהיה שייך לו, פשוט לפי מי ההסתברות שלו יותר גדולה.

לגבי הקריאות החישוב של ההסתברויות כתבנו ב-main, ואחרי הקריאה של הפונקציות האילו ב-main ואחרי שנקבל חישובים ותוצאות יש קטע קוד ב-main שכותב התוצאות לקובץ sentences_results.txt לפי הסדר המבוקש.

שלב 4 :

1. כן שמנו לב להבדל משמעותי בין שני המודלים כך קבלנו לרוב תוצאות שונות וזה בגלל התוכן השונה בכל אחד מהקורפוסים הוועדה ומליאה, כלומר יש שונות בנושאים בין כל אחד מהסוגים וגם צירוף מילים שונה בשני הקורפוסים וגם שונות במספר ובמקום הופעת מילים וגם בכלל הופעות של מילים או צירופים בקורפוס שלא יופיעו בקורפוס השני, וגם יתכן גודל הקורפוס השונה השפיע על התוצאות.

2. לגבי התוצאות של הקולוקציות תאמו לציפיות שלנו כך באמת קבלנו קולוקציות שנראות הגיוני שיהיו נפוצות בכללי וגם בפרט היה קולוקציות שצפינו לראות בגלל הנושא של הקורפוסים וההופעה שלהם הרבה באופן כללי. עכשיו לגבי האם הקולוקציות הנפוצות יכולות לסבר משהו : לגבי הקולוקציות הנפוצות בקורפוס הוועדה כמעט לא מראים לנו שום דבר על התוכן אלא רק עבור gram-4 יכול לזה יראה לנו שזאת וועדה, לגבי הקולוקציות הנפוצות בקורפוס המליאה זה יותר עוזר כך ניתן לראות עבורם שזה קשור לישיבה בכנסת.

3. עבור המדד tfidf באמת לא צפינו בכלל לקבל קולוקציות האלו, וכן יש קולוקציות שמראות שזה קשור לוועדה או ישיבה אבל לא מראה ממש שזה קשור לכנסת.

4. כן ניתן לראות בקלות שיש הבדל משמעותי בין הקולוקציות שקבלנו עבור כל מדד, וזה כי מדד ה-frequency מתחשב רק במספר ההופעה של הקולוקציה אבל המדד tfidf בחישוב גם נותן משקל למשמעות הקולוקציה והיחס של הקולוקציה במשפט ובמשפטים בכללי.

5. כן קבלנו משפטים הגיוניים אבל גם לפעמים קבלנו משפט לא הוגן כי יתכן שהצירוף של שני מילים לפני החסר [*] לא נמצא בכלל בתוך המשפטים בקורפוס, (אז בחישוב לפי המדד Linear אם אין צירוף trigram אז החישוב של trigram התאפס ואז עכשיו כאילו החישוב לפי bigrams וגם יתכן ה-bigrams התאפס ואז החישוב יהיה לפי ה-unigrams ואז הדבר הזה יתכן לרוב לתת תוצאה לא מתאימה או לא הוגנת למשפט שלנו) או המילה הבאה הכי נפוצה לא מתאימה להמשך המשפט שלנו.

6. לפי ההערכה שלנו נקבל תוצאות יותר טובות כי עכשיו נתייחס לצירוף 3 מילים קודם אז הגיוני לקבל תוצאה יותר טובה, במקביל גם ניתן לקבל משפטים לא הגיוניים בגלל הסיבות שהזכרנו קודם אבל בכללי זה ישפר התוצאות שנקבל.