**חלק א': יצירה של מודל Word2Vec ואימונו על קורפוס הכנסת:**

1. היכנסו לתיעוד המודל, הבינו מה המשמעות של כל ארגומנט שנתנו לו ( min\_count, vector\_size, window ) והסבירו מה היתרונות והחסרונות של הערכים שנבחרו עבור השימוש שלנו. האם הייתם בוחרים ערכים אחרים?
   * יתרונות:
     1. Min\_count : המשמעות שלה היא שאנחנו לא מתייחסים למילים שהתדירות שלהם פחות מ min\_count (לפי המקרה שלנו אנחנו מתייחסים לכל מילה) כך אנחנו לא מאבדים שום מידע.
     2. Vector\_size : אנחנו מספקים המימד של הוקטור, ולכן לכל מילה יש את הווקטור המיצג שלה המימד שלו יהי לפי ה vector\_size (במקרה שלנו זה 100) כך נשמור על סיבוכיות טובה וגם הייצוג הוא גם טוב עבור מילה כלשהי.
     3. window : אנחנו מספקים למודל עד כמה יסתכל קדימה או אחורה מהמילה שהוא נמצאה בה (במקרה שלנו זה אנחנו נסתכל מקסימום 5 מילים קדימה ו5 מילים אחורה) הפרמטר הזה עוזר לנו לבדוק את ההקשר בין המילים והוא עושה לנו איזון בין הקשר רחוק להקשר קרוב.
   * חסרונות:
     1. בגלל שיש לנו קורפוס שהוא קטן וגם כאשר אנו לוקחים min\_count = 1 אזי זה עלול לצור overfitting .
     2. משום שהשתמשנו ב min\_count = 1 אזי זה עלול להוסיף רעש.
     3. יש לא מעט משפטים שהם ארוכים וזה יוצר תלות בין מילים במרחק גדול 5.
   * הייתי משנה את גודל ה window כי בקורפוס שלנו יש הרבה משפטים שהם ממש ארוכים ולכן הייתי מגדיל את window ומשאר את השאר כי גם כך הקורפוס שלנו ממש קטן ולכן אם אנחנו נחליט להגדיל את ה min\_count שזה עלול לפגוע ולהקטין עוד יותר ואת גודל הווקטור הוא בסדר כך גם אנחנו נשמור על סיבוכיות טובה.
2. הסבירו מה הבעיות שיכולות לעלות משימוש במודל הנ"ל שאומן על הקורפוס שלנו. התייחסו בתשובתכם לאופן יצירת הקורפוס, לגודל שלו ולשימושים פוטנציאלים של המודל.
   * הגודל של הקורפוס ממש קטן ולכן הוא ממש לא מכיל הרבה נושאים והוא ממוקד רק על דברים שנאמרו בכנסת.
   * גודל המודל הוא קטן והוא ממוקד רק בנושאים שנאמרו בכנסת ולכן כך אנחנו נעשה overfitting והוא יתאים רק לנושאים מסוג זה.
   * כך גם אנחנו לא למדנו הרבה על מילים נדירות ולא יהיו לנו הרבה מילים נרדפות כך שכל קטגוריה תהיה ממש רלוונטית.

השתמשנו בפונקציה tokenization(sentence) שמקבלת כפרמטר משפט כלשהו:

* עברנו על טוקן (שהוא מילה):
  + עברנו על כל אות בטוקן:
    - אם יש אות בעברית אז זה טוקן חוקי.
  + אם אין אותיות בעברית בטוקן אז לא מכנסים לרשימת הטוקנים.

השתמשנו בפונקציה get\_excel שמקבלת כפרמטר את הנתיב לקובץ הקורפוס כדי שאוכל לשלוף את המידע ומופרד לפי מה שמבוקש:

* קראית את קובץ ה excel שהוא הקורפוס
* עברנו על שורות של הקובץ:
  + בחרנו בעמוד של ה sentence\_text .
  + אם הוא committee אזי מוספים לרשימה של ה committee אחרת ל plenary .
* עברנו על כל רשימה בנפרד:
  + עשינו טוקניזציה (באמצעות tokenization(sentence) ) למשפט והכנסנו לרשימה חדש.

מחזרים את שלוש הרשימות.

השתמשנו ב word2vec כדי לצור את המודל עם הפרמטרים המתאימים.

ההחלטה לגבי ה טוקניזציה:

* אם אנחנו מוחקים גם את תווים " או ' אז היינו מקבלים דברים לא מובנים ולכן כדי לקבל את המילה טובה גם אם זה קיצר או רשאי תיבות זה נשאר יותר טוב.
* כך גם יעזור לקבל יותר מילים וגם לסווג מילים שיש להם הקשר למשל למילה רמטכ"ל אז כך המילה הזו עוזרת לנו לסווג יותר מילים בצורה יותר טובה.

**חלק ב׳: דמיון בין מילים:**

1. האם המילים הכי קרובות שקיבלתם בסעיף א' תואמות את הציפיות שלכם? הסבירו. גם אם תאמו לציפיות וגם אם לא, נסו להסביר מדוע זה עבד או לא עבד טוב.
   * הם תאמו את הציפיות שלנו, כי קבלנו מילים שהם קרובים או באים באותם הקשרים או נושאים שהמילה באה ולכן גם קבלנו ציונים שהם קרובים למילה.
   * קבלנו גם מילים שהם קרובים לאותה מילה ויש להם אותה שורש ולכן גם זה מתאים לצפיות שלנו.
   * יש מילים שיש להם יותר ממשמעות אחד למשל שלום שזה יכול להיות שהיא ברכה שמקובל להגיד בתחילת בפגישה או במובן מדיני שהוא מצב של יחסים דיפלומטיים בין מדינות ולכן זה יכול להשפיע מאוד על ציונים ועל התוצאות ובכל זאת נקבל דברים קרובים בנושאים אלה.
2. אם ניקח שתי מילים שנחשבות להפכים ( antonyms ), למשל "אהבה" ו"שנאה", או "קל" ו"כבד". האם היינו מצפים שהמרחק בין שני וקטורי המילים שלהן יהיה קצר או ארוך?
   * קצר, משום שאהבה ושנאה הם באותה קטגוריה שזה רגש וממש הם באים באותם הקשרים ואותו משפטים ולכן כפי שראינו בהרצאה שזה יכול להוביל לדמיון גודל.
3. מצאו זוג מילים שנחשבות להפכים ( antonyms ) הקיימות בקורפוס שלנו ובידקו את המרחק ביניהן. האם הציפייה שלכם מסעיף 2 מתקיימת עבורן עם המודל שבניתם?
   * כן, כפי שנ"ל הם יכולים להיות באותה קטגוריה וגם ההקשר במפשט זהה ויהי תקין אם נחליף אחד עם השני
     1. אחרי שהפעלנו את similarity בין המילה שנאה ואהבה נקבל 0.94741905 שזה גבוהה.
4. האם המשפטים הכי קרובים בסעיף ג' תאמו לציפיות שלכם? הסבירו. גם אם תאמו לציפיות וגם אם לא, נסו להסביר מדוע זה עבד או לא עבד טוב.
   * כן זה התאים לציפיות שלנו כי ממש החליף את המפשטים כמצופי וראינו את החולשות ואת החוזקות (את הבחירה של המפשטים ועוד הסבר מעמיק נמצא למטה) כפי שמצופה:
     1. כפי שראינו יש משפטים שהוא החליף ממש באופן מושלם והחליף כל מילה או פשוט בחיר במפשט שהוא מסודר אחרת.
     2. וראינו גם משפטים שהוא החילף במפשט שממש זהים אבל הוא החילף במילה בהפך שלה ולכן קבלנו משפט שהוא הפוך המשפט המקורי מבחינת משמעות.
     3. יש משפטים הארוכים במיוחד ויש בהם מילים נדירות הוא החליף במשפט שלא היה ממש קשור במשמעות אלו.

השתמשנו בפונקציה similarity\_between\_words(model, save\_output) שמקבלת כפרמטרים את המודל ועוד נתיב איפה לשמור את הפלט:

* עברנו על 5 המילים:
  + והשתמשנו ב model.wv.most\_similar(word, topn=5) כדי לקבל את 5 המילים הכי קרובות ושמנו אותם כפי שמבוקש בקובץ.

השתמשנו בפונקציה sentence\_embeddings(word\_vectors, corpus) שמקבלת שני פרמטרים שזה ה model.wv ואת הקורפוס שמכיל את כל המפשטים באופן מסודר.

* אנחנו עוברים על כל משפט בתוך הקורפוס:
  + עוברים על כל הטוקנים בתוך המפשט:
    - בודקים את המילה נמצאת במודל אם כן אז מוספים אחד ל k ומוספים את הווקטור ל summing .
  + בודקים אם k שונה מ 0 אז מוספים את התוצאה לרשימה.

השתמשנו בפונקציה closest\_sentences(embeddings, save\_output) שמקבלת שני פרמטרים שזה ה embeddings של כל המפשטים שיש בקורפוס:

* בחרנו 10 משפטים שמייצגים את החלשות החזקות של המודל.
* עברנו על כל משפט:
  + עברנו על כל המשפטים שיש בקורפוס:
    - בדקנו לפני ה embeddings ובעזרת cosine\_similarity ולקחנו את המשפט שיש לנו ציון הכי גבוה.

**הבחירה של המשפטים:**

בחירה שלנו הייתי ממש כדי לייצג את החולשות ואת החזקות של המודל ולא באופן רנדומלי:

* החזקות:
  + אם יש לנו משפט מסוים ויש עוד משפט שיש לו אותו מילים ויש לו אותה משמעות אזי הזוג הזה מקבל ציון ממש גבוהה וגם הוא שמור על הקוהרנטיות ועל התחבירית ועל התוכן.
  + גם כאשר יש לנו משפט ונוכל להחליף את המילים שלו במילים אחרים שהם באותה קטגוריה או שהמרחק בניהם ממש קצר לכן נקבל משפט דומה וגם יש לו אותה משמעות.
  + ויש כאלה שנוכל להחליף את כל המשפט וגם לקבל במילים קרובות וגם נקבל משפט שיש לו אותה משמעות.
* חלשות:
  + יש משפטים שמצאנו משפט אחר שיש לו רק מילה אחת שונה והיא במרחק קצר ממשפט מקורי אבל יש לה משמעות הופכית .
  + יש משפטים שנקבל מהמפשטים המקוריים שהם מכילים את המפשטים המקוריים ויקבלו ציון ממש גבוהה כי יש להם את כל המילים ויותר.

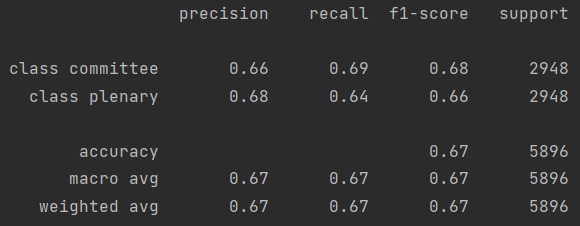
השתמשנו ב replace\_red\_words(model, save\_output) שמקבלת שני פרמטרים שהם המודל ונתיב כדי לשמור את הקובץ ששומר את המשפטים לאחר ההחלפה:

* עברנו על כל משפט ומצאנו את המילים שיכולו להחליף את המילה האדומה כך:
  + "ברוכים הבאים , הכנסו בבקשה לחדר ." :
    - כדי שנוכל להחליף את "לחדר" במילה קרובה החלטתי להשתמש במילה קרובה אליה שהיא "בית" ואז קבלתי מילה קרובה ל "בית" זה "לבית" ואז המפשט המתקבל הוא: "ברוכים הבאים , הכנסו בבקשה לבית .", הבחירה של המילה "בית" כי היא ממש קרובה למשמעות וממש נוכל להחליף אותה במשפט ונקבל דבר שהוא תקין מכל הבחינות.
  + "אני מוכנה להאריך את ההסכם באותם תנאים." :
    - כדי שנוכל להחליף את "מוכנה" שמתי בתוך positive=['מוכנה', 'אישה'] ובתוך negative=['גבר'] ואז כך קבלתי יכולה
    - הבחירה שלנו הייתה באותה מילה והוספנו אישה כדי שנוכל להתמקד בפעולות של אישה וגם ללא הפעולות של גבר.
    - כדי שנוכל להחליף את "ההסכם" לא היה לנו אפשרות להשתמש באותה מילה אלא במילה נרדפת וקרובה ל "הסכם" שזו "הסדר" ואז קבלנו ההסדר.
  + "בוקר טוב , אני פותח את הישיבה ." :
    - כדי שנוכל להחליף את "טוב" החלטתי לקחת מילה קרובה אליה שזו "מדהים" ואז קבלנו מבורך שזה מילה קרובה לטוב וגם המפשט שקבלנו הוא תקין מכל הבחינות.
    - כדי שנוכל להחליף את "פותח" החלטתי לקחת מילה קרובה במשמעות במשפט שהיא "מתחיל" ואז קבלתי גם את המילה "ממשיך" שזה גם קרובה ומתאימה למשפט.
  + "שלום , הערב התבשרנו שחברינו היקר לא ימשיך איתנו בשנה הבאה ." :
    - כדי שנוכל להחליף את "שלום" בחרנו את המילה "חברים" כי היא קרובה אליה בהקשר ובמיקום של המילה בתוך המשפט שנוכל לשים "חברים" במקומה ולכן בחרנו ב"חברים" כדי שנקבל "שימו" שהיא גם מילה מתאימה לתחילת המפשט.
    - כדי שנוכל להחליף את "היקר" החלטתי לקחת את המילה קרובה במשמעות אליה שהיא "הטוב" ואז קבלתי "הגדול שזו מילה מתאימה למשפט.
    - החלפנו את "בשנה" ל "לשנה" שזו מילה ממש קרובה למילה המקורית וגם מתאימה למשפט.

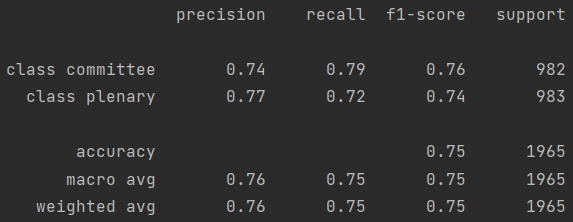
**חלק ג׳: סיווג:**

1. האם עבור אותם פרמטרים ותנאים שהשתמשתם בהם בתרגיל 3 (צ'אנק בגודל ,5 אותה כמות שכנים , שיטת חלוקה וכו') קיבלתם תוצאות טובות יותר או פחות עבור וקטור המאפיינים הנ"ל?
   * פחות טובות.
2. עבור התשובה שעניתם בסעיף ,1 הסבירו מדוע לדעתכם זה קרה.
   * אנחנו משתמשים בווקטורים שהם הממוצע של מילה בתוך הצ'אנק ולכן יכול להיות שווקטור של מילה שהיא לא כל כך רלוונטית הוא גודל ומשפיע יותר מווקטור של מילה שהיא מאוד חשובה ולכן כך אנחנו נאבד קצת את החשיבות של המילה הזו.
   * אם יש מילה אחת שהיא מופיעה הרבה ב committee ולא plenary אנחנו נעשה ממוצע ולכן כך נאבד את המילה שהיא ממש תכונה חשובה לסווג.
   * גם אנו לא מחשבים או מתייחסים למילים שהם לא במודל OOV .
   * מילים נדירות הולכות לאיבוד או לא משפעיות כי אנחנו ממש לוקחים את הממוצע כך אם יש לנו מילה נדירה שהיא שייכת אחד משני ה classes אזי היא תקבל ציון ממש קטן או זניח.
3. עבור איזה גודל צ'אנק קיבלתם תוצאות יותר טובות? האם זה נכון גם לגבי וקטורי המאפיינים שהשתמשתם בהם בתרגיל 3? הסבירו.
   * עבור צ'אנק בגודל 5, כן גם כך קבלנו בתרגיל בית 3.
     1. הסיבה היא שצ'אנק בגודל 5 יהיו תכונות יותר שיהוו מאפיינים משמעותיים למסווג וכך לגבי הצ'אנקים שיצרנו הווקטור יקבל ערכים יותר מתאימים לקטגוריה המתאימה.
     2. כאשר ה צ'אנק מכילה יותר ממשפט אחד אזי זה יקטן את הרעש ואת המשקל של מילים שהם לא רלוונטיים.
     3. ואת המילה שמהוו תכונה עבור class מסוים היא תחזור על עצמה במשפטים אחרים ואז תקבל ציון יותר טוב.

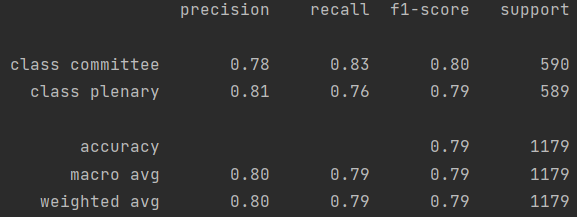
צ'אנק בגודל 1:



צ'אנק בגודל 3:



צ'אנק בגודל 5:



השתמשנו בפונקציה down\_sampling(c\_chunks, p\_chunks) שמקבלת שני פרמטרים את משפטים בנפרד של הplenary ושל הcommittee :

* אנחנו מחזרים את # הצנקים שיהיו שווים ל plenary ול committee .

השתמשנו בפונקציה make\_chunks(committee, plenary, chunk\_size) שמקבלת שלושה פרמטרים את כל המפשטים של committee וגם ה plenary וגם את גודל של כל צאנק :

* עוברים על כל רשימה:
  + מכנסים chunk\_size משפטים בתוך רשימה חדשה מתאימה שמכילה את כל צנקים המתאמים .
  + משתמשים בפונקציה שנ"ל down\_sampling כדי שנאזין את ה chunk ים.

השתמשנו בפונקציה classification\_knn(word\_vectors, plenary, committee, chunk\_size) שמקבלת ארבעה פרמטרים שהם המודל ( mode.wv ) וכל המפשטים שהם plenary וכל המשפטים שהם committee וגודל הצאנק :

* משתמשים ב make\_chunks כדי לייצר את ה צאנקים.
* משתמשים ב sentence\_embeddings שנ"ל עבור כל רשימה של צאנקים .
* מייצרים לאבלים בהתאמה.
* ואז משתמשים ב train\_test\_split כדי לקבל קבוצה אימון וקבוצה בדיקה.
* אימנו את המודל שהוא knn ואז עשינו חיזוי ואז הדפסנו את ה classification\_report .

**חלק ד': שימוש במודלי שפה גדולים:**

1. האם קיבלתם משפטים הגיוניים? מבחינת התוכן, קוהרנטיות ומבחינה תחבירית.
   * כן, בכל הבחינות קבלנו דברים טובים יותר אבל יש כמה משפטים שלא הושלמו בצורה מושלמת.
2. השוו את התוצאות שקיבלתם עכשיו לאלו שקיבלתם בתרגיל בית .2 האם יש שיפור בתוצאות לדעתכם?
   * כן, כי יש משפטים בתרגיל בית 2 אנחנו השלמנו עם סמני פיסוק במקומות לא נכונים וגם מבחינת התוכן וקוהרנטיות ומבחינה תחבירית ממש קבלנו דברים שהם יותר טובים.
3. האם יש משפטים שעבורם המודל עבד פחות טוב? אם כן, הסבירו מה לדעתכם הסיבה לכך. אם לא, האם לדעתכם הוא יעבוד בצורה מושלמת על כל משפט מתוך קורפוס הכנסת? הסבירו.
   * כן, יש משפטים שהשלם באופן לא קוהרנטית וגם יש מפשטים שיש להם בעיה בתוכן, סיבות:
     1. הפגת עמימות למשל במפשט : "בכל העולם מדברים באנגלית." מדברים לא כל כך מתאימה בה היה יותר טוב אם הינו משלמים ב "מתקשרים".
     2. המודל יכול להשלים את המשפט מבחינה תחבירית טוב אבל מבחינה הגיון וקוהרנטיות לא השלים אותה בצורה טובה.
   * הוא יעבוד בצורה טובה על רוב המשפטים כפי שקבלנו בתוצאות קודמות אבל כמו קודם יש כאלה שהוא לא ישלם בצורה טובה מבחינה קוהרנטית וגם מבחינה תחבירית.

השתמשנו בפונקציה fill\_masked(masked\_sentences\_path, output\_path) שמקבלת שני פרמטרים את הקובץ שמכיל את המפשטים שצריך להשלים ועוד נתיב כדי לשמור את התוצאה:

* עוברים על כל משפט:
  + מחלפים את [\*] ב [MASK] ואז מפעילים את המודל על כל משפט והוא מחזר לי את כל המילים החסרים.
  + עובר על כל המילים שהמודל החזיר ומשלים את המשפט.