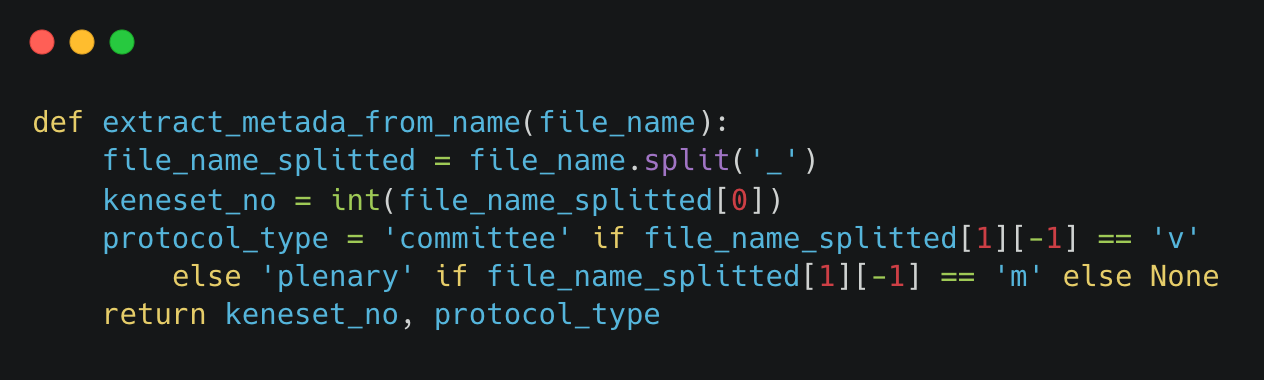
דו"ח תרגיל בית 1

קורפוסים

שלב 1

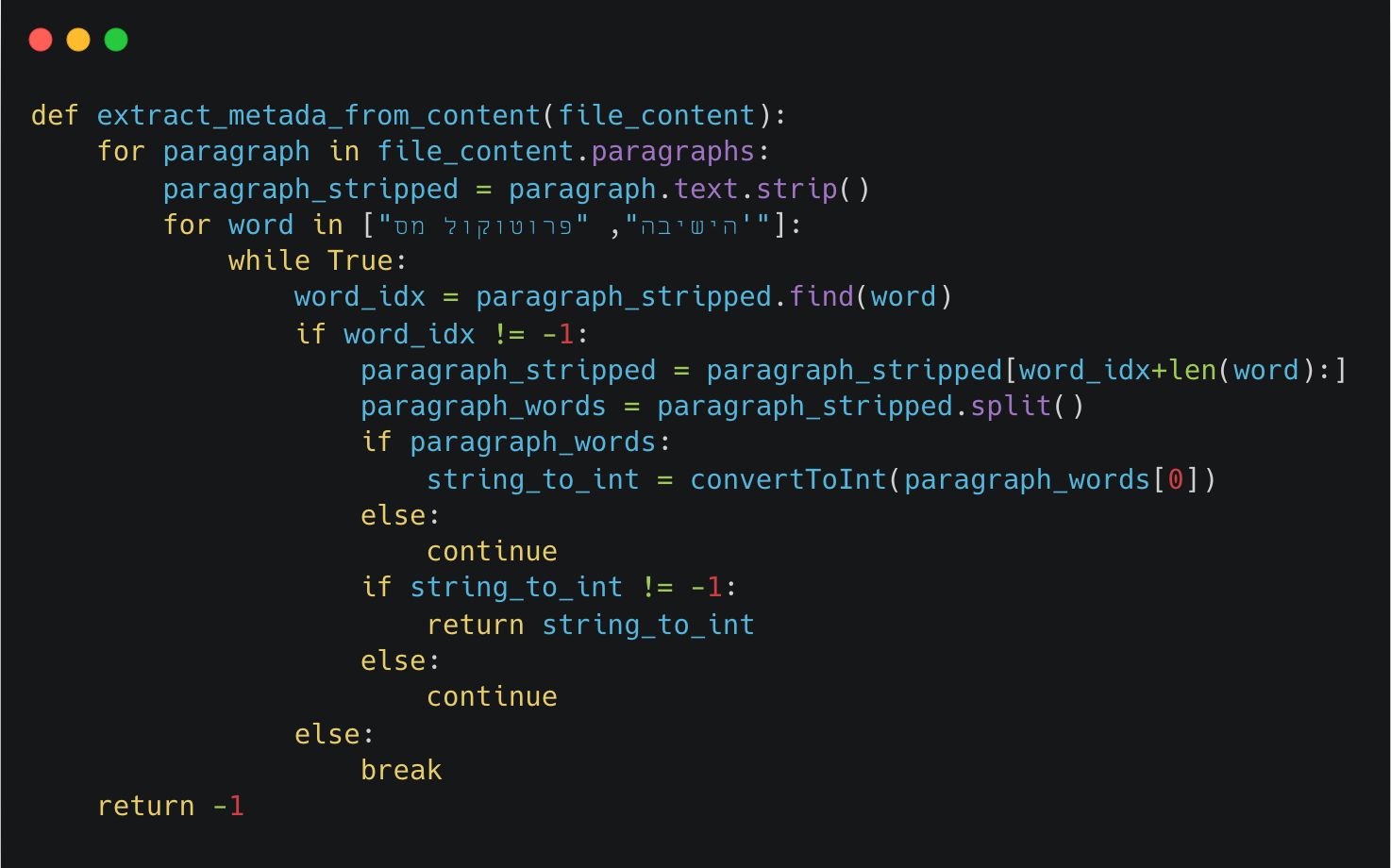
שלב 1.1:

שליפת הנתונים של מספר הכנסת וסוג הפרוטוקול ממומשת ע"י הפונקציה 'extract\_metada\_from\_name'. אופן העבודה שלה מתבסס על העובדה שלשמות קבצי הפרוטוקולים יש מבנה מסויים. הפונקציה מקבלת כקלט את שם הקובץ ומחלצת ממנו את הנתונים הדרושים ע"י לחלק אותו ל- 3 חלקים לפי ה- '\_', ומחזירה את החלק הראשון כמספר הכנסת והאות האחרונה בחלק השני כסוג הפרוטוקול.



שלב 1.2:

שליפת הנתון של מספר הישיבה מתממש ע"י הפונקציה 'extract\_metada\_from\_content'. המימוש שלה מתבסס על העובדה שהמספר של הפרוטוקול בא אחרי אחד משני צירופי המלים "הישיבה" או "פרוטוקול מס'". היא מקבלת כקלט את התוכן של הקובץ, ומתחילה בלעבור על כל המופעים של המילה "הישיבה" ובודקת אם המילה שמופיעה אחריה היא מספר, אם לא מצליחה למצוא כזה מופע, היא עושה אותו דבר עבור המופעים של הצירוף "פרוטוקול מס'". הפונקציה עוצרת בעת שהיא מוצאת מספר כמתואר למעלה ומחזירה אותו, או כאשר עוברת על כל המופעים של שני הצירופים ולא מוצאת, ואז מחזירה -1.



בדיקת האם המילה היא מספר ואת ערכה במידה שהיא מספר מתבצעת ע"י הקריאה לפונקציה 'convertToInt'. אשר מקבלת כקלט משתנה string, שיכול להיות מורכב מספרות (כמו 73) או מאותיות (כמו: שבעים ושלוש). היא ממירה אותו למקביל שלו ב- int, ואם לא מצליחה היא מניחה שזה לא מספר ומחזירה -1.



שלב 1.3:

הדרוש בסעיף ממומש ע"י הפונקציה 'extract\_relevant\_text', אשר משתמשת גם בפונקציות עזר שונות. היא מקבלת כקלט קובץ ופרוטוקול (מופע של המחלקה Protocol), ומחלצת מהקובץ את המשפטים הרלוונטיים ושומרת אותם לתוך הפרוטוקול. היא מבצעת את העבודה באמצעות השלבים הבאים:

1. למצוא את הטקסט הרלוונטי, אשר הוגדר בנתוני השאלה כ- "הטקסט שנאמר על ידי דוברים בוועדה\מליאה". מהתסתכלות בקבצי הפרוטוקולים השונים ניתן לראות שהקט הרלוונטי נמצא ברצף, לכן נרצה למצוא את תחילת הקטע וסופו.
   1. **מציאת תחילת הקטע הרלוונטי** מתבצעת ע"י הקריאה לפונקציית העזר ' find\_starting\_relevant'. מהידע שלנו על אופן התנהלות הדיונים בכנסת הוועדה\המליאה מתחילה בדבריו של היו"ר של הכנסת או המ"מ שלו. דבר זה משתקף בקבצי הפרוטוקולים. כמו כן בעיון בקבצים השונים נוכל לשים לב שהתואר שלו תמיד נמצא לפני שמו, ו- ":" תמיד מופיעות אחרי שם כל דובר.. לכן המטלה הופכת להיות ללמצוא את המשפט הראשון שמתחיל באחד מהצירופים יו"ר\ היו"ר\ יו"ר הכנסת\ מ"מ היו"ר ונגמר ב- ":", זהו תחילת הקטע הרלוונטי.
   2. **מציאת סוף הקטע הרלוונטי** מתבצעת ע"י הקריאה לפונקציית העזר ' find\_last\_relevant'. ניתן לשים לב שכמעט כל דיון נגמר במשפט "הישיבה ננעלה בשעה .." או לחלופין "הטקס ננעל בשעה ..". משפט זה בא אחרי המשפטים שנאמרו ע"י הדוברים, וגם לא נאמר ע"י מישהו, לכן ניתן להניח שהוא המשפט הראשון הלא רלוונטי. כמו כן, ניתן לשים לב שבמקרים שהקובץ לא נגמר במשפט מהסוג הזה הדיון נמשך עד הסוף. על סמך הזה הפונקציה מאתר את סוף הקטע הרלוונטי ע"י להתחיל לחפש מהסוף את אחד מבין צירופי המלים "הישיבה ננעלה" או "הטקס ננעל", ואם לא מוצאת אותן אז היא מחזירה שהסוף של הקטע הוא בסוף הקובץ.
2. מיון המשפטים בקטע הרלוונטי לאחד מבין שני הסוגים: שם הדובר, משפט שהוא חלק מהנאום. זה מתבצע ע"י הפונקציה הראשית ('extract\_relevant\_text') אשר עוברת על הקטע שסומן כרלוונטי, היא מאתרת את המשפטים מסוג "שם הדובר" ע"י כך שהיא בודקת אם הוא עומד בשלושה תנאים: יש קו מתחתיו, מסתיים ב- ":", ומכיל לכל היותר 5 מילים אחרי הניקיון (על ניקיון השם מפורט אח"כ). אם שלושת התנאים מתקיימים, המשפט נחשב לכזה המכיל שם דובר, וכל המשפטים שמפרים אחד מהתנאים הללו ובאים אחריו משויכים אליו, עד למציאת שם דובר חדש. ברגע שמשפט של נאום נמצא הוא מטופל ע"י פונקציית העזר sentence\_handle, אליה מועבר המשפט, שם הדובר, והמופע של הפרוטוקול. על אופן עבודת הפונקציה מפורט יותר בלשבים 1.4-1.7.

* ניקוי השם: מתבצע ע"י הקריאה לפונקציה 'speakerClean'. מטרת הפונקציה היא "לנקות" את השם, כלומר להסיר כל דבר חוץ שמכיל השם מלבד השם הפרטי ושם המשפחה, ואולי גם שם האב. זה כולל תארים (כמו השר, היו"ר, ד"ר וכדו') או שם המפלגה או סימונים של <> שבהרבה פורמטים מופיעים לפני ואחרי השם. מימוש הפונקציה מתבסס על סקירת האופנים השונים שבהם יכול השם להופיע. למשל שם המפלגה תמיד בא בין סוגריים, לכן הפונקציה מאתרת סוגריים כאלה ואם מוצאת אותם מוחקת אותם מהשם ביחד עם מה שיש בתוכם. הפונקציה מנסה לאתר תארים שונים נפוצים ולמחוק אותם. בתארים של שר ושרה היא מניחה שיבוא אחריהם את שם המשרד של השר, לכן היא בודקת אם אחד משמות המשרדים מתוך רשימה מופיעים אם כן גם אותם מסירה.

שלב 1.4:

המשפטים מחולקים ע"י [. ! ?]. זה מתבצע ע"י הפונקציה 'sentence\_handle', היא עוברת על כל אות במשפט שאותר כמשפט נאום, ובודקת עבור כל אות אם הוא אחד משלושת הסימנים, אם כן הוא חותך את מה שלפניו מזה שאחריו.

שלב 1.5:

המשפטים מנוקים אחרי השלב שתואר בסעיף הקודם (1.4), הם נשלחים לפונקציה ' sentence\_validity'. שאחראית לכך, ופוסלת אותם אם הם עונים לאחד משלושת הדברים הבאים: משפט שלא מכיל אף אות בעברית, משפט שמכיל לפחות אות אחת באנגלית, משפט חתוך (כזה שמכיל ---). אם המשפט לא עונה לאחד משלושת הדברים הוא מצליח במבחן "החוקיות".

שלב 1.6:

השלב מתבצע ע"י הפונקציה 'sentence\_tokenize'. הטוקנים מופרדים ע"י כל אחד מבין הרווח והסימנים והבאים [, ; : ( )]. לסימן ["] שיכול לבוא כמפריד (למשל, סיום ציטוט) או כחלק מטוקין (למשל כחלק מקיצור כמו ת"א), הוחלט לבדוק את המקרה שלו לפי ההקשר שהוא בא, אם שכניו מימין ומשמאל הם אותיות אז הוא נחשב כחלק מהטוקין, ואם לפחות אחד משכניו הוא רווח או תחילת\סיום משפט אז הוא נחשב למפריד. כל המפרידים מלבד הרווח נשמרים כטוקנים. הפונקציה מחזירה משפט שמופרד ע"י רווח בין כל שני טוקנים.

שלב 1.7:

השלב מטופל ע"י הפונקציה 'sentence\_tokenize'. אחרי שהפונקציה מבצעת טוקניזציה וכאשר הטוקנים נמצאים בתוך רשימה, היא בודקת את אורכה ואם היא לפחות אחת היא מאחדת את הטוקנים למשפט, אחרת פוסלת את המשפט.

שלב 1.8:

הפונקציה jsonl\_make נוצרה והיא אחראית על המרת הרשימה protocols לפורמט של jsonl. היא פותחת את קובץ הפלט במצב כתיבה עם קידוד UTF-8. ועוברת על כל המופעים של protocol, כאשר בכל אחד מהם היא עוברת על המשפטים (ששמורים בו כ- attribute של רשימה). היא יוצרת עבור כל משפט שורה בקובץ ה- jsonl עם התכונות שהוזכרו בהוראות.

שלב 2

שאלה 2.1: מצד אחד, פיצול מוספיות חשוב בתהליך של עיבוד שפות על מנת להיות אפשר שלזהות ששתי מילות הן זהות גם אם אחת מהם באה עם מוספית. מצד שני, בניגוד לאנגלית ולכמה מילות חיבור בעברית (כמו "וגם"), בעברית קשה לפעמים לזהות אם אות היא מוספית או שהיא חלק מהמילה, דוגמה קלאסית היא האות "ש" במילה "שבתה" שהיא בעלת שתי משמעויות, ורק באחת מהם האות "ש" היא מוספית. מכאן שפיצול כזה יכול לפגוע במקום להועיל.

שאלה 2.2: ניתן לפצל את המילה ל- 4 טוקנים. "ו", "כש", "יבוא", "ו". כאשר "ו" הראשון מציין חיבור, "כש" מציין תיאור זמן, "יבוא" היא המילה המרכזית בזמן עתיד, ו- "ו" האחרונה מציינת ריבוי. הפיצול הזה מאפשר לזהות את מרכיבי המילה בצורה ברורה, כמו כן עוזרת בהבנת ההקשר.

שאלה 2.3: