

תרגיל 1 – קורפוסים – דו"ח מסכם

שלב 1 - טיפול בטקסט

מבוא

המטרה בתרגיל הייתה לבנות קורפוס בעברית מתוך כ-98 פרוטוקולים של הכנסת (קבצי וורד בפורמט docx). המערכת קוראת כל מסמך, מחלצת מטא-דאטה משם הקובץ, מזהה דוברים ואת הנאומים שלהם, מחלקת למשפטים, מבצעת ניקוי וטוקניזציה בסיסית ללא ספריות חיצוניות, ושומרת פלט לקובץ מסוג JSONL (שורה = משפט) בקידוד UTF8. הדגש היה דיוק בבחירת הטקסטים הרלוונטיים והימנעות מסינון יתר, תוך שמירה על עמידה מלאה בדרישות של התרגיל.

מטרות המטלה

- **חילוץ מטא-דאטה משם הקובץ:** מספר הכנסת (int) וסוג פרוטוקול (plenary/committee).
- **חילוץ מספר פרוטוקול:** זיהוי והחזרה into או 1- אם לא נמצא.
- **חילוץ יו"ר הישיבה:** איתור יו"ר כשם פרטי+משפחה נקי מתארים.
- **חילוץ דוברים וטקסטים:** שמירה על טקסט רלוונטי בלבד והימנעות מכותרות/תוכן עניינים.
- **חלוקה למשפטים וניקוי:** ללא ספריות חיצוניות, עם סף מינימלי של לפחות 4 טוקנים.
- **טוקניזציה פשוטה:** פיסוק כטוקנים נפרדים (עם מספר מקרים חריגים), רווח יחיד בין טוקנים.
- **פלט JSONL:** בפורמט קבוע, בקידוד UTF-8 ללא הדפסות למסך בכלל.

מתודולוגיה ויישום

(א) חילוץ מטא-דאטה משם הקובץ

- שם קובץ בפורמט: NN_ptm_ID.docx או NN_ptv_ID.docx
- ביטוי רגולרי: `(\d+)_pt([mv])(\d+)\.docx`
 - `kneset_number:int = NN`
 - `protocol_type="plenary" = m`
 - `protocol_type="committee" = v`
- אם הפורמט לא תואם - הקובץ מדולג באופן בטוח (ללא קריסה בעזרת try-except).

(ב) חילוץ מספר הפרוטוקול

- חיפוש במסמך כולו לפי תבניות נפוצות:
פרוטוקול מס, \d+, מספר ישיבה: \d+, ישיבה מס, \d+, פרוטוקול: \d+.
- אם נמצא אז מבוצעת המרה ל int אחרת 1-.

(ג) זיהוי יו"ר הישיבה

- יו"ר מוגדר רק כשנמצאת תגית דובר אמיתית שמכילה ביטוי יו"ר (למשל היו"ר / יושב-ראש וכו').
- מנגנון ניקוי שם (clean_name) מסיר תארים/תפקידים (למשל ח"כ/שר/ד"ר), סוגריים וסימני פיסוק מיותרים, ודוחה מחרוזות שלא מייצגות שם (כותרות, ציטוטים, התחלה במירכאות)
- אם לא זוהה יו"ר מפורשות אז אנחנו שומרים מחרוזת ריקה.

(ד) זיהוי דוברים והפרדה מטקסטי מערכת

- שורת דובר חוקית: מסתיימת בנקודתיים ונראית כשם לאחר ניקוי. כותרות/תוכן עניינים/הודעות מערכת מסוננים.
- הפונקציה looks_like_header מזהה ומסננת: "תוכן העניינים", "הצעות לסדר-היום", "תאריכים/שעות/מיקום (ירושלים, שעה...)", "מסמכים שהונחו...", "הודעת/סקירת...", "הצעת חוק...", "קריאה שנייה/שלישית" ועוד.
- לפני הופעת הדובר הראשון - לא מייחסים טקסט לאף אחד (נמנעים מהדבקת פתיח ליו"ר).
- לאחר מכן - רק טקסט שמופיע אחרי תגית דובר מיוחס לדובר הנוכחי, שורות ללא דובר נוכחי אינן משויכות.

(ה) חלוקה למשפטים

- פיצול על גבולות . ? ! ; (לא מתבצע פיצול שרואים נקודותיים ":").
- סינון רווחים וצמצום מקטעים ריקים.

(ו) ניקוי משפטים

- הסרה של רצפים לא תקינים: קווים תחתונים ---, תו נקודה תבליט •.
- יחס מינימלי של עברית במשפט ($hebrew_ratio \geq 0.25$) כדי למנוע סינון יתר ולשמר כמות נתונים.
- שמירה רק אם יש יותר מ 4 טוקנים.

(ז) טוקניזציה

- פיסוק נפרד כטוקנים (כללית).

- **חריגים שנשמרים כמילה אחת:**

- גרשיים בין אותיות עבריות (למשל ח"כ/יו"ר) נשמרים כמילה אחת באמצעות הגנה זמנית.

- מקף בתוך מילה עברית (למשל "יושב-ראש") נשמר.

- איחוד רווחים לרווח יחיד.

(ח) פלט JSONL

- כתיבה ל JSONL-בקידוד utf-8 ללא הדפסות למסך.

- שימוש ב `json.dumps(..., ensure_ascii=False)` לשמירת עברית תקינה.

- החזרת קודי יציאה ללא הדפסות.

- 1 - שימוש לא תקין, 2- תיקיית קלט לא קיימת, 3- שגיאת קריאה/סריקה, 4 - שגיאת כתיבה, 0 - הצלחה.

אתגרים ושיקולים

- **שמות שאינם דוברים (איכות הנתונים):** ביטויים כמו "מסקנות הוועדה", "מנהלות הוועדה", "רשמת פרלמנטרית" הופיעו בתגיות של דוברים. פתרון: רשימת חריגים + סינון כותרות לפני שיוך לדובר.

- **סינון לא אגרסיבי מדי:** הוגדר סף עברית מתון (0.25) ושמירה על ספרות/פיסוק, כדי לשמור על כמות משמעותית של משפטים.

- **דיוק ביו"ר:** יו"ר נקבע רק מתוך תגית דובר אמיתית עם ציון יו"ר, ולא מסריקות כלליות, מונע טעויות כמו קטעי פתיחים שנקלטו כשם יו"ר.

- **טיפוסים נכונים:** `protocol_number` | `knesset_number` נשמרים כמספרים או כ-1 אם לא נמצאו.

- **יציבות והימנעות מקריסות:** סביב כל שלב יש `try/except` כדי למנוע קריסה של התהליך על קובץ בעייתי.

מבנה הפלט ודוגמה

שדות הפלט לכל משפט:

`protocol_name, kneset_number, protocol_type, protocol_number, protocol_chairmain, speaker_name, sentence_text`

דוגמה (מהרצת בדיקה):

{"protocol_name":"13_ptm_532058.docx","knesset_number":13,"protocol_type":"plenary","protocol_number":-1,"protocol_chairmain":"ש' וייס","speaker_name":"ש' חברי הכנסת , אנחנו פותחים את ישיבת הכנסת , יום שלישי , כ"ז" , "sentence_text":"כ"ז" , "וייס"}
בכסליו התשנ"ג , 22 בדצמבר 1992

שלב 2 - שאלות

שאלה 1

אני חושב שזה באמת עניין של trade-off מצד אחד , פיצול המורפמות יכול לעזור למודל להבין טוב יותר את המבנה הדקדוקי - כי אז הוא רואה בבירור את ההבדל בין החלקים התפקודיים (כמו "ו" או "כש") לבין השורש של המילה עצמה. בנוסף, זה יוצר איזשהו סטנדרט - אותו שורש תמיד יופיע באותה צורה, לא משנה אילו תחיליות מחוברות אליו.

מצד שני, יש בעיה ממשית: כשמפרקים מילה למורפמות, אפשר לאבד את המשמעות השלמה שלה. ניקח לדוגמה ביטוי כמו "וכשיבואו" - זה לא סתם רצף של מורפמות, זה ביטוי שלם שמתאר מצב ספציפי. וגם מבחינה טכנית, כל פיצול כזה פירושו יותר טוקנים, מה שמסרבל את העיבוד.

בהתחשב בדרישות של התרגיל, נראה לי שבחירה לא לפצל היא די סבירה - היא חוסכת סיבוכים מיותרים ושומרת על זרימה טבעית יותר.

שאלה 2

נניח שהייתי מחליט לפצל את המילה "לכשתצטרכו" הייתי מחלק אותה באופן הבא:

- ל - מורפמת יחס
- כש - מורפמת זמן
- תצטרכו - הפועל עצמו

למה ככה? כי כל אחד מהחלקים האלה עובד אחרת בתוך המשפט. "ל" ו-"כש" זה בעצם כלים דקדוקיים שיכולים להתחבר למון מילים שונות, ואילו "תצטרכו" זה הליבה - הפעולה שבאמת קורית. ככה המודל יכול "ללמוד" את המורפמות התפקודיות פעם אחת ולהשתמש בהן בכל מיני הקשרים.

שאלה 3

להחליטה לשמור כל משפט בנפרד עם כל הפרטים שלו, יש כמה יתרונות ברורים:

קודם כל, זה נוח מאוד לעבודה מדויקת. רוצים לבדוק מה דובר מסוים אמר על נושא מסוים? פשוט מחפשים לפי השדות הרלוונטיים. גם מבחינת ניתוח - אפשר לרוץ על המשפטים אחד אחד, לסווג אותם, לבדוק טון, מה שרוצים.

אבל יש גם מחיר: הקשר בין המשפטים לא ממש נשמר. משפט שני עשוי להיות תשובה ישירה למשפט ראשון, אבל אם הם מאוחסנים כרשומות נפרדות, הקשר הזה לא אוטומטי. בנוסף, כל

משפט נושא איתו את כל המטא-דאטה (מספר פרוטוקול, דובר וכו'), מה שיוצר המון מידע כפול וחזרתיות במערכת.

לעומת זאת:

- אם שומרים את כל הפרוטוקול ביחד, יש את ההקשר המלא אבל קשה יותר לבודד משפטים בודדים.
- אם מקבצים לפי דובר, רואים את כל מה שאדם מסוים אמר אבל מאבדים את הסדר הכרונולוגי של הדיון.

שאלה 4

הייתי נשאר עם גישת המשפט הבודד, אבל הייתי עושה לה שדרוג קטן.

הרעיון: לשמור כל משפט בנפרד (כי זה נותן את הגמישות הכי גדולה למשימות שונות), אבל להוסיף שדות שישמרו את ההקשר:

- **קישור למשפט הקודם** - ככה אפשר תמיד לגשת אחורה ולראות את הרצף.
- **תגיות תוכן** - למשל, תקציר קצר של מה הפרוטוקול עוסק בו או נושא הדיון הספציפי. בעצם זו גישה היברידית: מקבלים את היכולת לעבוד ברמת פירוט של משפט בודד, אבל לא מוותרים על האופציה להבין את התמונה הרחבה כשצריך.