עיבוד שפה טבעית (20225211) 2024-2025 - תרגיל 4 חלק 2 (4 נק')

שלום לכולכן!

כעת נתבונן קצת יותר לעומק בשיטות ליצירת אוצרות מילים (vocabularies) מתחת לרמת המילה (subword). נאסוף קורפוס, נאמן טוקנייזר לעברית שמתאים ל-BERT, ונבחן את תכונותיו.

בסיום העבודה, הגישו **קובץ pdf** הכולל את כל הפלטים מכל התאים. דרך אפשרית אחת להשיג קובץ pdf היא על-ידי הדפסת הדף (מתוך colab, לא מהדפדפן) ולבחור "מדפסת" ישירה ל-pdf.

שימו לב שכל המידע הדרוש מופיע בתדפיס ה-pdf --- לא יתקבלו ערעורים על-בסיס חיתוך. אין לחרוג ממבנה התאים ואין לענות במקומות שאינם מוגדרים לכך.
אין למחוק הערות שמסמנות את מיקומי קטעי הקוד שלכם ואין לערוך קוד שנמצא מחוץ לתחומים המוגדרים. סעיפים שחורגים מהנחיה זו לא ייבדקו.

ניתן להגיש בצוותים של עד שלוש.ה סטודנטיות. לא יהיו הנחות לצוותים קטנים. אפשר לחלק את פתרון התרגיל בין חברי.ות הצוות כרצונכם. אין להסתכל על קוד של צוותים אחרים ואין להקריא קוד לצוותים אחרים.

אין להשתמש ברכיב ג'מיני Gemini או כל סייען מתקדם אחר.

מועד ההגשה הוא דקה לפני תחילת השיעור ב-13/1. לא יינתנו הארכות.

בהצלחה,

-- יובל

התקנות

דבר ראשון, נתקין את החבילות הנדרשות, כולן מבית האגינגפייס.

```
!pip install datasets -q
!pip install tokenizers -q
!pip install transformers -q
```

נייבא חבילות. את `os` נצטרך כדי לכתוב את הדאטאסט שלנו לטובת האימון, וכדי לכתוב את אוצר המילים של הטוקנייזר לאחר האימון כדי שיוכל להיטען לאובייקט הקורא אותו.

```
import os
import datasets
from tqdm.notebook import tqdm
```

החבילה `datasets` מכילה קורפוסים לא-מתויגים רבים, בדיוק מהסוג שאנחנו צריכים לטובת אימון טוקנייזר תת-מילי. נשתמש ב-OSCAR, פרויקט נקיון ותחזוקה של מידע ממאגר הגריפה הגדול `commoncrawl` התומך ב-166 שפות, וביניהן עברית.

נבקש ממנו את מחיצת האימון (היחידה שקיימת) לעברית ונספק את הפרמטר `streaming` כדי שנוכל לדגום חלק קטן ממנה בלי להוריד הכל לזכרון של קולאב (המון זמן). 200,000 מסמכים יהיו די והותר.

```
dataset = datasets.load_dataset(
    'oscar',
    'unshuffled_deduplicated_he',
    split='train',
    streaming=True,
    trust_remote_code=True)

sampled_ds = []
for i, doc in tqdm(enumerate(dataset)):
  if i > 200_000:
    break
  sampled_ds.append(doc)
```

```
0it [00:00, ?it/s]
```

Start coding or generate with AI.

היעלה על הדעת שנוריד מידע ולא נתבונן בו? הבה נבחן את המסמכים האחרונים שהורדנו.

```
sampled_ds[-10:]
```

```
[{'id': 199991,
  'text': 'ולכבוד הסרט גוגל בנו משחק קצרצר של מלחמת הכוכבים הפועל "Star Wars The Force Awakens" בלילה האחרון החלו הקרנות הבכורה בארץ ובעולם של הסרט החדש Storm עליכם יהיה להכנס לקישור עם הקוד המיוחד שיוצג לכם באתר במהשב מהפלאפון ולאחר ביצוע כיול קצרצר ופשוט תוכלו להדוף יריות ולבסוף גם להלחם בn\.בתוך דפדפן הכרום תגיות: גוגל, דפדפן, כרוm,n\.לצערי המשקח מאוד מאוד קצר ומכיוון שמדובר במשחק דפדפן באיכות לא רעה ושליטה שניעשת ממכשיר אחר דרך הרשת יש מדי פעם קצת האטהn\.Trooper מלחמת הכוכבים, Chrome, Lightsaber, Star Wars, Storm Trooper, The Force Awakens', משחק, סרט},
 {'id': 199992,
  'text': 'התרגלנו לחשוב שיש לנו סוג של בעלות על האדם האהוב עלינו. לפעמים מספיק שנראה את הבן זוג שלנו מסתכל על אישה אחרת, והקנאה מתעוררת. האמירה הבאה אולי לרפואה המזרחית גישה מעניינת ואפקטיבית מאוד, לתופעות גיל המעבר. המחלקה לרפואהn\תפתיע אתכם: קנאה אינה שייכת לאהבה! אם תאמצו זאת כמוטו, חיי האהבה שלכם ישתנו לחלוטין סינית ויפנית במרפאות אלטרנטיבה מציעה טיפול ייחודי משולב לתופעות המטרידות של גיל המעבר: גלי חום, הזעות לילה, חרדה, מתח ואי שקט, דפיקות לב מואצות, הפרעות שינה ודיכאון. אנשי עסקים מצליחים שהכסף מצוי בכיסם מגיעים לפעמים לצומת דרכים. הכסף מסתבר לא עונה על הכל והתשובה עשויה להימצא בדווקא במקומות הרוחניים. על ידועונים\לאיזון ושקט נפשי הטבעn\הרוחניות החדשה של היום, טוענת שאין סתירה בין רוחניות וכסף. בהחלט אפשרי ליהנות משני העולמות. כיצד עבודה רוחנית יכולה לתמוך בחיבור לשפע חומרי\רוחניות ואנשי עסקים מנדלה היא יצירה שצומחתn\יודע לשחרר את הישן ולפנות מקום לחדש: העלים היבשים נושרים ועלים ירוקים חדשים יצמחו במקומם, בבוא העת. רק האדם לעתים נצמד לישן, מפחד משינויים מתוך מעגל פנימי, אשר גדלה ומתפתחת לעוד ועוד מעגלים סביבו, עד אין סוף. מנדלה היא מילה בסנסקריט שפירושה מעגל מקודש. היא מלווה דתות שונות ואף שבטים ותרבויות קדומות עוד מהתקופה הפרהיסטורית\n'שינויים מתרחשים כל הזמן ולכן עלינו לשנות את יחסנו אליהם, הן בתחומים מקצועיים והם בתחומים אישיים. כיצד לקבל את השינויים כחלק בלתי נפרד מהחיים\n'},
 {'id': 199993,
  'text': 'כולנו מנהלים את עצמינו במספר מעגלי חיים: קריירה , משפחה, חברה , תחומי עיניין אישיים . לא פעם אנו מוצאים את עצמינו "נקרעים" בין ה"עולמות" הנ"ל ולא פעם ( Well-being הרווחה האישיתn\.התוצאה היא הזנחה של חלק מתחומים, תשלום מחירים גבוהים בבריאות , בקשר עם הילדים , בקשר עם הבן זוג , ברמת הסיפוק אישי, או מקצועי ועוד במסגרתn\.מושגת תוך סיפוק הצרכים האישיים, המקצועיים והמשפתיים. סיפוק צרכים אלו מתאפשר ע"י ניהול אפקטיבי של המשימות וניתוב נכון של האנרגיות האישיות במעגלי חיים השונים ( 'מאנית\. האימון נפתח כלי התנהלות אפקטיביים לניהול המשימות ולניהול זמן , נאתר יחד את הדרך לאיזון האנרגיות'},
 {'id': 199994,
  'text': 'הנוסע המוסלמי אבן בטוטה (1304-1377) ביקר בטורקיהn\דף הבית >> הרצאות >> טורקיה >> סקר ארכיאולוגי\\היסטורי של אתרים מרכזיים יוונים ורומיים במחוזות טורקיה ב-1330 ותיאר אותה כאזור המופלא שבעולם, אכן מדינה מרתקת, ייחודית גן עדן ארכיאולוגי, נופים וחופים מהיפים בעולם. החל משנת 1998 לאחר לימודי הארכיאולוגיה ביקרתי, סיירתי האזור בו שוכנת כיום טורקיה המודרנית היה ערש הולדתן של תרבויות חשובות, בהן האימפריותn\וצילמתי יותר מ- 20 פעמים, היתה לי הזכות לצלם מאות אתרים ארכאולוגים ואלפי מנומנטים הביזאנטית והעות\'מאנית. בשל המיקום האסטרטגי של טורקיה, בנקודת המפגש בין אסיה ואירופה, התרבות הטורקית היא מיזוג מיוחד במינו של המסורת המערבית והמזרחית. יש לה גם חשיבות האימפריה המשמעותית הראשונה באזור הייתה האימפריה החתית שהתקיימה בין המאות ה-18 וה-13 לפנה"ס. הפריגים ירש את החתים עד שהקימרים\n.גיאו-פוליטית אסטרטגית משמעותית החריבו את ממלכתם במאה ה-7 לפנה"ס. במקביל, בין שנת 1000 לפנה"ס ועד למאה ה-6 לפנה"ס התקיימה ממלכת בחלקה המזרחי של אסיה הקטנה, עד שנכנעה למדי. את מקומה של פריגיה תפסו מספר ממלכות שהחשובות שבהן היו לידיה, קאריה וליקיה, כולן בחלקה המערבי של אסיה הקטנה, והיוונים החלו מיישבים אותה לקראת סוף האלף השני לפנה"ס. האזור כולו נכבש על ידי הממלכה האחמנית במאות ה-6 וה-5 לפנה"ס, ולאחר מכן על ידי אלכסנדר הגדול בשנת 334 לפנה"ס. בתקופה ההלניסטית חלקו מספר ממלכות את שטחה של אסיה הקטנה ובמאה ה-2 לפנה"ס החלו הרומאים בכיבושה והשלימו אותו בשנת 65 לפנה"ס לערדטורקיה היא רפובליקה דמוקרטית, חילונית, וחוקתית. מוסטפא כמאל אטאטורק בנה את המערכת הפוליטית שלה ב-1923, לאחר נפילת האימפריה העות\'מאנית. טורקיה נמנתה עם החברות המייסדות של האומות המאוחדות, ושל ארגון המדינות האסלאמיות, והצטרפה לנאטו ב-1952. מאז 2005, מנהלת טורקיה משא ומתן להצטרפות לאיחוד האירופי.'},
 {'id': 199995,
  'text': 'אם זה לגובה או אם זה למרחב, בכל סוג של פרויקט מתקן החשמל דורש שיטות תכנון מתקדמות אשר התפתחו עם השנים מליבה יציבה ומוכחת של ידע, ניסיון ועבודת צוות. שימוש בטכנולוגיות חדשות לצורך קביעת סטנדרטים חדשים במתקני החשמל אותם מתכנן המשרד ושילובם בחזית ההתפתחות הטכנולוגית העצומה שחלה בהקמת פרויקטים מורכבים בארץ ובעולם בשנים האחרונות.'},
 {'id': 199996,
  'text': 'אחד השיקולים החשובים ביותר בבחירת מקרר הוא בחירת הנפח המתאים. מקרר גדול מידי יגרום לבזבוז אנרגיה ומקרר קטן מידי יהיה מוגבל ביכולת האחסון, 500-600 ליטרים בנפח מתאים למשפחה הממוצעת. ישנם שני סוגים של מדידות נפח: *מדידות ישנות - מדידות אלו כוללות את כל הנפח של המקרר משמע ללא המגירות והמדפים - נפח ברוטו. *מדידות חדשות מקררים יכולים לבוא בתצורות ובצבעים שונים, האפשרויות בחירה באתר שלנו גדול מאוד וכאן תוכלו למצוא כל מקררn\.- מדידות אלו כוללות את המדפים והמגירות ולכן זהו הנפח נטו שתחפשו. ניתן להזמין מקררים עם ציפוי מזכוכית (בתוספת תשלום), במגוון צבעים עצום - לבן, כסוף, בז\', קרם שמנת, טיטניום, גימורי נירוסטה מכל מיני סוגים, לבן שחור ואפילו על ידי הממלכה האחמנית במאות ה-6 וה-5 לפנה"ס, ולאחר מכן על ידי אלכסנדר הגדול בשנת 334 לפנה"ס. בתקופה ההלניסטית חלקו מספר ממלכות את שטחה של אסיה הקטנה ובמאה ה-2 לפנה"ס החלו הרומאים בכיבושה והשלימו אותו בשנת 65 לפנה"ס לערדטורקיה היא רפובליקה דמוקרטית, חילונית, וחוקתית. מוסטפא כמאל אטאטורק בנה את המערכת הפוליטית שלה ב-1923, לאחר נפילת האימפריה העות\'מאנית. טורקיה נמנתה עם החברות המייסדות של האומות המאוחדות, ושל ארגון המדינות האסלאמיות, והצטרפה לנאטו ב-1952. מאז 2005, מנהלת טורקיה משא ומתן להצטרפות לאיחוד האירופי.'},
 {'id': 199995,
  'text': 'אם זה לגובה או אם זה למרחב, בכל סוג של פרויקט מתקן החשמל דורש שיטות תכנון מתקדמות אשר התפתחו עם השנים מליבה יציבה ומוכחת של ידע, ניסיון ועבודת צוות. שימוש בטכנולוגיות חדשות לצורך קביעת סטנדרטים חדשים במתקני החשמל אותם מתכנן המשרד ושילובם בחזית ההתפתחות הטכנולוגית העצומה שחלה בהקמת פרויקטים מורכבים בארץ ובעולם בשנים האחרונות.'},
 {'id': 199996,
  'text': 'מקררים יכולים לבוא בתצורות ובצבעים שונים, האפשרויות בחירה באתר שלנו גדול מאוד וכאן תוכלו למצוא כל מקררn\.- מדידות אלו כוללות את המדפים והמגירות ולכן זהו הנפח נטו שתחפשו. ניתן להזמין מקררים עם ציפוי מזכוכית (בתוספת תשלום), במגוון צבעים עצום - לבן, כסוף, בז\', קרם שמנת, טיטניום, גימורי נירוסטה מכל מיני סוגים, לבן שחור ואפילו בנפח מתאים למשפחה הממוצעת. ישנם שני סוגים של מדידות נפח: *מדידות ישנות - מדידות אלו כוללות את כל הנפח של המקרר משמע ללא המגירות והמדפים - נפח ברוטו. *מדידות חדשות מקרר חדש, עדיף לבחון את הדירוג האנרגטי שלו ולקנות מקרר אשר יצדיק את ההשקעה הכספית למשך שנים רבות. ככל שהמקרר יותר חסכוני בצריכת אנרגיהn\.אוויר קר ולח כאשר בוחרים מקרר חדש, עדיף לבחון את הדירוג האנרגטי שלו ולקנות מקרר אשר יצדיק את ההשקעה הכספית למשך שנים רבות. ככל שהמקרר יותר חסכוני בצריכת אנרגיהn\.אוויר קר ולח מייצרת אוויר קר ויבש בתוך המקרר על מנת של יצטבר קרח בדפנות, אוויר קר ויבש פוגע בצורה משמעותית יותר באוכל שבמקרר מאשר No-frostלטכנולוגיה זו יש גם חסרון: טכנולוגיית ה - No-frost. טכנולוגיה זו מייצרת קרח בדפנות הפנימיות של המקרר. קרח זה מצטבר עקב האוויר הקר והלח שהמקרר מייצר ועקב כך צריך להפסיק את פעולת המקרר אחת לחודש על מנת להפשיר את המקרר. טכנולוגיה זו מתקדמת יותר וקיימת בכל המקררים החדישים ביותר, לא צריך להפסיק את פעולת המקרר לעולם משום שלא מצטבר קרח בדפנות אך - No-frost הקרח המדובר. טכנולוגיית No-frost ישנם שני סוגי סוגי מקררים: טכנולוגיות קירור במקררים: טכנולוגייתn\!מקרר ויטרינה עם דלת כסופה - De-frost טכנולוגיה זו כבר ישנה ואט אט מוחלפת ע"י הטכנולוגיה החדישה יותר - No-frost. מקרר ריקוד גברים נעלי נוחות סוויד / PU $29.99 – 2018 6534228 שחור / סתיו נעליים ללא שרוכים לבן / אביב / חומר עליוn\: PU סוויד - זמש הוא החומרים הפופולריים ביותר במרחב ההנעלה.לא רק בגלל אופנה אלא גם מעשי.אתה אף פעם לא יכול לפספס נעלי זמש,'},
 {'id': 199998,
  'text': 'כדי לסייע לכם להתאים את השיש לבחירות השונות שצריך לעשות במסגרת תהליך של עיצוב מטבחים, נועד מאמר זה. השתמשו בו בתבונה והשיגו ממנו את המרב. אתם בהחלט דעו כי בכל הנוגע לבחירת שיש,n\.יכולים לקחת את מה שמועיל לכם ולהתעלם ממה שלא נראה לכם רלוונטי, אך דעו כי הבחירה הנכונה היום, היא שתאפשר לכם ליהנות מהשיש מחר ובכלל אין בחירה נכונה ובחירה לא נכונה, משום שהבחירה הינה אינדיווידואלית - מה שנכון לאחד לא נכון לאחר ולהיפך. הבחירה תלויה בין היתר באהבה שלכם לבישול ואפיה וכפועל יוצא תדירות נשאלת השאלה החשובה מתי נכון לבחור את השיש והאם לתקופה בה מתבצעת הבחירה יש משמעות מסוימת, התשובה על כדn\.השימוש במשטח, בטעם האישי שלכם מבחינה עיצובית ועוד
```

לפי הסרט המוזכר במסמך 199,991, עושה רושם שהדאטאסט, או לפחות החלק הזה שלו, הוא מאמצע העשור הקודם.

שמירת דאטאסט מסודר למטרות אימון טוקנייזר

התא הבא מכין את הדאטא עבור מנגנון האימון שנכיר בחלק הבא.

```
CORPUS_DIR = './oscar_heb'

if not os.path.exists(CORPUS_DIR):
  os.mkdir(CORPUS_DIR)

text_data = []
file_count = 0

for sample in tqdm(sampled_ds):

    # remove newline characters from each sample as we need to use exclusively as seperators
    sample = sample['text'].replace('\n', '\s')
```

```
    text_data.append(sample)
    if len(text_data) == 5_000:

        # once we hit the 5K mark, save to file
        with open(f'{CORPUS_DIR}/text_{file_count}.txt', 'w', encoding='utf-8') as fp:
            fp.write('\n'.join(text_data))

        text_data = []
        file_count += 1

# after saving in 5K chunks, we may have leftover samples, we save those now too
with open(f'{CORPUS_DIR}/text_{file_count}.txt', 'w', encoding='utf-8') as fp:
    fp.write('\n'.join(text_data))
```

```
<>:12: SyntaxWarning: invalid escape sequence '\s'
<>:12: SyntaxWarning: invalid escape sequence '\s'
C:\Users\Benzu\AppData\Local\Temp\ipykernel_17860\3037718808.py:12: SyntaxWarning: invalid escape sequence '\s'
  sample = sample['text'].replace('\n', '\s')
  0%|          | 0/200001 [00:00<?, ?it/s]
```

נבדוק את תקינות פורמט השמירה.

```
from pathlib import Path

CORPUS_DIR = './oscar_heb'
paths = [str(x) for x in Path(CORPUS_DIR).glob('**/*.txt')]
paths[:5]
```

```
['oscar_heb\\text_0.txt',
 'oscar_heb\\text_1.txt',
 'oscar_heb\\text_10.txt',
 'oscar_heb\\text_11.txt',
 'oscar_heb\\text_12.txt']
```

אימון טוקנייזר של BERT מהדאטה שאספנו

הטוקנייזר של BERT נקרא וורדפיס (WordPiece). האלגוריתם לאימון שלו דומה מאוד ל-BPE, ואופן חלוקת המילים לתמניות (token) ("פיסות" ("pieces")) שלו חמדני. בוורדפיס מסמנים פיסה שאינה בתחילת מילה על-ידי התחילית ## .

נאתחל טוקנייזר ריק ונאמן אותו על הדאטא שלנו. באתחול, צריך לפרט אילו פעולות טרום-טיקנוז (pre-tokenization) אנחנו רוצים לעשות, ובאימון צריך לפרט את ההיפר-פרמטרים של אוצר המילים המבוקש (נבחר ב-3,000), השכיחות המינימלית של טוקנים שיצורפו לאוצר המילים, ומגבלה על גודל האלפבית ההתחלתי. שני הפרמטרים האחרונים של `train()` נועדו להתאים את אוצר המילים שניצור לפורמט המקובל של הרובוטריק BERT.

האימון צפוי לקחת כ-10 דקות.

```
from tokenizers import BertWordPieceTokenizer

# initialize
tokzr_trainer = BertWordPieceTokenizer(clean_text=True)

# and train
tokzr_trainer.train(files=paths, vocab_size=3_000, min_frequency=2,
                limit_alphabet=1_000, wordpieces_prefix='##',
                special_tokens=[
                    '[PAD]', '[UNK]', '[CLS]', '[SEP]', '[MASK]'])
```

נשמור את המודל...

```
MODEL_DIR = './wp-heb'

if not os.path.exists(MODEL_DIR):
  os.mkdir(MODEL_DIR)

tokzr_trainer.save_model(MODEL_DIR, 'wp-heb')
```

```
['./wp-heb\\wp-heb-vocab.txt']
```

... ונטען אותו באמצעות האובייקט המתאים.

> *מי שמעוניינת מוזמנת לשמור את הקבצים בנפרד מהמחברת ולטעון לפי הצורך בשלבי פתרון התרגיל כדי לחסוך את זמן האימון עם כל סשן. עם זאת, ההגשה הסופית חייבת להיות תוצר של הרצה אחת נקייה מההתחלה ועד הסוף.*

הבה נסתכל על כמה דוגמאות:

```
from transformers import BertTokenizer

MODEL_DIR = './wp-heb'
tokzr = BertTokenizer.from_pretrained(f'{MODEL_DIR}/wp-heb-vocab.txt')

print(f'There are {tokzr.vocab_size} tokens in our tokenizer\'s vocabulary.\n===')
print(tokzr.tokenize('hello world'))
print(tokzr.tokenize('שלום עולם'))
print(tokzr.tokenize('אם אין אני לי מי לי'))
```

```
c:\Users\Benzu\anaconda3\Lib\site-packages\transformers\tokenization_utils_base.py:1925: FutureWarning: Calling BertTokenizer.from_pretr
  warnings.warn(
There are 3000 tokens in our tokenizer's vocabulary.
===
['h', '##el', '##l', '##o', 'w', '##or', '##l', '##d']
['שלו', '##ם', 'עו', '##לם']
['אם', 'אין', 'אני', 'לי', 'מי', 'לי']
```

כמו שאנחנו רואים, הטוקנייזר שלנו בהחלט יודע לעבד עברית יותר טוב מאשר אנגלית, בהתאם לדאטא שקיבל לאימון.

סעיף 1 (0.5 נק'): הציעו הסבר מדוע בכל-זאת "שלום" מפריד את האות הסופית אך "עולם" מפריד שתי אותיות אחרונות. ניתן להוסיף תא קוד להנמקה.

הסבר אפשרי הינו ש'שלו' הינו נפוץ יחסית ולכן קיבל טוקן משל עצמו ,מאחר וזהו אלגוריתם גרידי הוא תחילית ארוכה יותר מ'של' ואילו 'עול' ככל הנראה לא נפוץ מספיק בשביל לקבל טוקן משל עצמו ולכן 'עו' קיבל את התחילית הארוכה ביותר (למטה ניתן לראות שעולים ועושים שניהם מתפרקים לטוקן 'עו')

```
print(tokzr.tokenize('שלום עולם'))

print(tokzr.tokenize('את לא כמו כולם'))
print(tokzr.tokenize('אין מקום בעולם בשבילם'))
print(tokzr.tokenize('סתם הלום'))

print(tokzr.tokenize('עולים לגובה עושים הרבה'))
print(tokzr.tokenize('שלחתי שליחים לשלום שלומית'))
```

```
['שלו', '##ם', 'עו', '##לם']
['את', 'לא', 'כמו', 'כול', '##ם']
['אין', 'מקום', 'בעולם', 'בש', '##ביל', '##ם']
['ס', '##תם', 'חל', '##ום']
['עו', '##לים', 'לג', '##ובה', 'עו', '##שים', 'הרבה']
['של', '##חת', '##י', 'שלי', '##חים', 'לש', '##לו', '##ם', 'שלו', '##מית']
```

כפי שאנו זוכרים מהשיעור, לרוב מאמנים טוקנייזר על אוצר מילים בסדר גודל של 30-50 אלף, ושלנו קטן פי עשרה. התוצאה היא הרבה פחות מילים וחלקי מילים שימושיים שזכו לקבל תמנית משלהן.

סעיף 2 (0.5 נק'): מצאו מילה (רצף אותיות עבריות ללא רווחים) תקינה בעברית שמתפצלת ליותר מ-6 תמניות.

```
### 3 ###
MY_WORD = 'האינציקלופדיה'
### # ###

toks = tokzr.tokenize(MY_WORD)
print(len(toks), toks)
```

```
7 ['האי', '##נ', '##צ', '##קל', '##ופ', '##די', '##ה']
```

כפי שהזכרנו בכיתה, אופן *הפענוח* (decoding, inference) של אלגוריתם וורדפיס הינו חמדני (greedy): בהינתן רצף תווים, וורדפיס יחפש את התמנית הארוכה ביותר שמתאימה לתחילתו, וימשיך מהנקודה בה נקטע.

סעיף 3 (1 נק'): ממשו את אלגוריתם הפענוח באופן נכון אך לא בהכרח יעיל, באמצעות אובייקט הקורא את אוצר המילים ופועל חמדנית. הקוד בתא השני יבדוק את נכונותו.

```
from typing import Callable, List

class GreedyTokenizerFromFile(Callable):
```

```
  def __init__(self, vocab_file: str) -> None:
    self.vocab = set()

    ### 5.1 (0.25 points) ###
    self.vocab |= set(open(vocab_file, 'r', encoding='utf-8').read().splitlines())

    print(f'initialized with {len(self.vocab)} pieces')

  def __call__(self, s: str) -> List:

    ### 5.2 (0.75 points, conditioned on test passing) ###
    tokens = []
    word = s
    while word != '##':
      for i in range(len(word), 0, -1):
        if word[:i] in self.vocab:
          tokens.append(word[:i])
          word = '##' + word[i:]
          break
        elif i == 1:
          tokens.append('[UNK]')
          return tokens

    return tokens


  ### you may define any additional functions below ###



##############################################
### Do not modify this cell when submitting ###
##############################################

from tokenizers.pre_tokenizers import BertPreTokenizer

greedy_tok = GreedyTokenizerFromFile(f'{MODEL_DIR}/wp-heb-vocab.txt')

# Bert's PreTokenizer handles punctuation. It's called implicitly by tokzr
pretok = BertPreTokenizer()

# our "test set" is a random document from the dataset
for w, char_span in tqdm(pretok.pre_tokenize_str(sampled_ds[2023]['text'])):

  # ground truth - the automatically-learned tokenization
  trained_tok = tokzr.tokenize(w)

  # preparation - BertTokenizer actually has a preprocessing submodule called
  #               "basic_tokenizer" that lowercases English and such
  grd_tok = greedy_tok(tokzr.basic_tokenizer.tokenize(w)[0])

  # Test - these asserts should not fail #
  assert trained_tok == grd_tok, f'{w}, {trained_tok}, {grd_tok}'
```

```
initialized with 3000 pieces
  0%|          | 0/25 [00:00<?, ?it/s]
```

משחק השבת

נניח שאנחנו מתכננים משימה שנוגעת לעיבוד טקסט ממסמכים הנוגעים לספורט. בואו נראה עד כמה המודל שלנו מותאם למילים מתוך התחום (domain) הספורטיבי:

```
' '.join(tokzr.tokenize('מכבי הפועל כדורגל כדורסל כדורעף טניס'))
```

```
'מכ ##בי הפ ##ועל כד ##ור ##גל כד ##ור ##ס ##ל כד ##ור ##ע ##ף ט ##ני ##ס'
```

נראה שהמצב די רע. ננסה לאמן מודל טוקניזציה שיטפל במילים האלה יותר טוב **רק באמצעות שינוי קורפוס האימון**.

סעיף 4 (2 נק', אין ניקוד חלקי): צרו דאטאסט חדש מתוך הקורפוס שכבר בידינו (אל תייבאו קורפוס חדש, אל תפצלו מסמכים, ואל תכתבו יותר מ-200,000 מסמכים לדיסק) ואמנו עליו מודל WordPiece עם אותם היפר-פרמטרים כמקודם, שמפצל לכל היותר מילה אחת מהמילים לעיל ליותר משתי תמניות, ולא מפצל בכלל לפחות שתיים מהמילים. הראו את הפלט. תעדו היטב באמצעות הערות את כל השינויים שערכתם ביחס לקוד המקורי.

צוותים שיפנו בשעות הקבלה שלפני תחילת השיעור ב-7/1 עם פתרון חלקי יוכלו לקבל רמז. לאחר מועד זה לא תינתן עזרה על הסעיף.

```
import re

def clean_text(text: str) -> str:
    """
    Cleans the input text by removing characters that are not Hebrew letters, digits, spaces, or specified punctuation/symbols.
    Args:
        text (str): The input text to be cleaned.
    Returns:
        str: The cleaned text with only allowed characters and no extra spaces.
    """

    hebrew_digits_punct_regex = re.compile(r"[^0-9א-ת\s.,!?;:'\"״־(){}[\]\-+*/@#$%^&*<>~|]")
    cleaned_text = hebrew_digits_punct_regex.sub("", text)
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()
    return cleaned_text

text = "45 (){*&^%$#@ (: האם זה עובד? ,123 זהו מבחן! שלוmabc <>"
cleaned_text = clean_text(text)
print(cleaned_text)
```

```
45 (){*&^%$#@ (: האם זה עובד? ,123 זהו מבחן! שלום <>
```

```
def download_corpus(keywords: List[str], n_docs: int) -> List[dict]:
    """
    Downloads a corpus of documents containing specified words.
    This is the ssame as before, just filtering the documents by specified keywords.
    Args:
        words (List[str]): A list of words to search for in the documents.
        n_docs (int): The number of documents to download.
    Returns:
        List[dict]: A list of dictionaries, each representing a document that contains any of the specified words.
    """
    i = 0
    sampled_sports_ds = []

    with tqdm(total=n_docs, desc="Finding sports documents") as pbar:
        for doc in dataset:
            if any([word in doc['text'] for word in keywords]):
                sampled_sports_ds.append(doc)
                i += 1
                pbar.update(1)
                if i >= n_docs:
                    break
    return sampled_sports_ds
```

```
import shutil

def save_clean_corpus(corpus: List[dict], directory: str) -> None:
    """
    Save a cleaned corpus to a specified directory in chunks of 5000 samples.

    Args:
        corpus (List[dict]): A list of dictionaries, each containing a 'text' key with the text sample.
        directory (str): The directory where the cleaned text files will be saved.
    Returns:
        None
    """

    if os.path.exists(directory):
        shutil.rmtree(directory)
    os.mkdir(directory)

    text_data = []
    file_count = 0

    for sample in tqdm(corpus, desc="Saving corpus"):

        # remove newline characters from each sample as we need to use exclusively as seperators
        sample = sample['text'].replace('\n', '\s')

        text_data.append(sample)
        if len(text_data) == 5_000:

            # once we hit the 5K mark, save to file
```

```
            with open(f'{directory}/text_{file_count}.txt', 'w', encoding='utf-8') as fp:
                fp.write(clean_text('\n'.join(text_data)))

            text_data = []
            file_count += 1

    # after saving in 5K chunks, we may have leftover samples, we save those now too
    with open(f'{directory}/text_{file_count}.txt', 'w', encoding='utf-8') as fp:
        fp.write(clean_text('\n'.join(text_data)))
```

```
<>:24: SyntaxWarning: invalid escape sequence '\s'
<>:24: SyntaxWarning: invalid escape sequence '\s'
C:\Users\Benzu\AppData\Local\Temp\ipykernel_17860\978491444.py:24: SyntaxWarning: invalid escape sequence '\s'
  sample = sample['text'].replace('\n', '\s')
```

```
SPORTS_WORDS = 'מכבי הפועל כדורגל כדורסל כדורעף טניס כדור ספורט'.split()
SPORTS_CORPUS_DIR = './sports_adjacent_ball'
SPORTS_MODEL_DIR = './sp-heb'
MODEL_NAME = 'sports_adjacent_ball-clean'


### 4 ###
# download the corpus with the sports keywords
sampled_sports_ds = download_corpus(SPORTS_WORDS, 50_000)

# clean and save the corpus
save_clean_corpus(sampled_sports_ds, SPORTS_CORPUS_DIR)

# All of this is the same as before, but with the new corpus
sports_tokzr = tokzr_sports_trainer = BertWordPieceTokenizer(clean_text=True)
sports_paths = sorted([str(x) for x in Path(SPORTS_CORPUS_DIR).glob('**/*.txt')])
tokzr_sports_trainer.train(files=sports_paths, vocab_size=3_000, min_frequency=2,
                limit_alphabet=1_000, wordpieces_prefix='##',
                special_tokens=[
                    '[PAD]', '[UNK]', '[CLS]', '[SEP]', '[MASK]'])
if not os.path.exists(MODEL_DIR):
  os.mkdir(MODEL_DIR)

tokzr_sports_trainer.save_model(MODEL_DIR, f'{MODEL_NAME}')
```

```
Finding sports documents:   0%|          | 0/50000 [00:00<?, ?it/s]
Saving corpus:   0%|          | 0/50000 [00:00<?, ?it/s]
['./wp-heb\\sports_adjacent_ball-clean-vocab.txt']
```

```
sports_tokzr = BertTokenizer.from_pretrained(f'{MODEL_DIR}/{MODEL_NAME}-vocab.txt')


### # ###

' '.join(sports_tokzr.tokenize('מכבי הפועל כדורגל כדורסל כדורעף טניס'))
```

```
c:\Users\Benzu\anaconda3\Lib\site-packages\transformers\tokenization_utils_base.py:1925: FutureWarning: Calling BertTokenizer.from_pretr
  warnings.warn(
'מכבי הפועל כדור ##גל כדור ##סל כדור ##ע ##ף ט ##ניס'
```