# תרגיל בית 2

## מודלי שפה – n-grams

## מבוא

מודלי שפה הם מודלים סטטיסטיים על רצפי מילים. בהרצאה ראיתם מודלי n-grams, המחשבים את הסתברות הופעת משפט (צירוף של טוקנים) באמצעות שערוך הסתברותו בקורפוס. בתרגיל זה תתנסו בבניית מודלי שפה על קורפוס הכנסת.

לשם כך, תשתמשו בקובץ הJSONL שיצרתם בתרגיל 1.

## שלב 1 : בניית מודלי שפה

להזכירכם, בקובץ הJSONL יש שדה המסמן אם המשפט הגיע מפרוטוקול של ועדה או של מליאה.

עליכם לבנות את מודלי השפה הבאים :

1. מודל מבוסס Trigrams לוועדות – מודל זה יבנה בעזרת כל המשפטים המשוייכים לפרוטוקולים מסוג ועדה (ורק הם).
2. מודל מבוסס Trigrams למליאות - מודל זה יבנה בעזרת כל המשפטים המשוייכים לפרוטוקולים מסוג מליאה (ורק הם).

לשם כך, עליכם לבנות מחלקה בשם **Trigram_LM** שתתאים לכל אחד מהמודלים. על המחלקה לכלול את המתודות הבאות :

1. *calculate_prob_of_sentence*

פונקציה זו מחשבת את ההסתברות של משפט (צירוף של טוקנים). היא תעשה זאת בעזרת נוסחאת הTrigram שלמדתם לחישוב הסתברות לרצף מילים : $P(w_1, \dots, w_n) = \prod_{k=1}^{n} P(w_k | w_{k-2}, w_{k-1})$, כאשר ההסתברויות יחושבו בעזרת MLE עם אינטרפולציה ליניארית.

- <u>קלט :</u>
  i. מחרוזת שמהווה רצף של טוקנים (מופרדים ברווח)
- <u>פלט :</u>
  i. הפונקציה תחזיר מספר float שמייצג את לוג ההסתברות של המשפט.

2. *generate_next_token*

פונקציה זו מנבאת את הטוקן הבא בהנתן צירוף של טוקנים. היא תעשה זאת בעזרת נוסחאת הTrigram שלמדתם, לחישוב הטוקן הבא : $P(w_k | w_1, \dots, w_{k-1}) = P(w_k | w_{k-2}, w_{k-1})$, כאשר ההסתברויות יחושבו בעזרת MLE עם אינטרפולציה ליניארית.

- <u>קלט :</u>
  i. מחרוזת שמהווה רצף של טוקנים (מופרדים ברווח)
- <u>פלט :</u>
  i. הפונקציה תחזיר tuple המכיל במיקום הראשון את הטוקן עם הסבירות הכי גבוהה עפ"י המודל להיות הטוקן הבא במשפט (רצף הטוקנים שהתקבל), ובמיקום השני את ההסתברות שהתקבלה עבור אותו הטוקן. לדוגמה : ("hello", 0.72)

**<u>הנחיות נוספות:</u>**

1. עבור מימוש <u>אינטרפולציה ליניארית</u> :
   a. השתמשו במקדמים שתבחרו לנכון **(פרטו עליהם והסבירו בדו"ח).**
   b. עבור כל רכיב בנוסחת האינטרפולציה השתמשו בהחלקת לפלס.

2. שימו לב שבנוסחאות הTrigram הנ"ל, עבור כל הסתברות יש תלות בשני הטוקנים הקודמים, גם כאשר אין כאלו (כאשר הרצף קצר מדי או כאשר מחשבים את ההסתברות של הטוקנים בתחילת המשפט). לכן, עליכם להוסיף 2 טוקני "דמה" בתחילת **כל** משפט s_0, s_1. טוקנים אלו צריכים להיות כמימוש פנימי בלבד ולא שקופים למשתמש.

## שלב 2 : קולוקציות

1. ממשו פונקציה בשם *get_k_n_t_collocations*

פונקציה זו מחזירה את k הקולוקציות באורך n הכי נפוצות בקורפוס מסויים, עפ"י מדד מסויים, ממויינות בסדר יורד (מהכי נפוצה לפחות) ואשר מופיעות בקורפוס לפחות t פעמים.

- <u>קלט</u>:
  i. k – מספר הקולוקציות הרצוי
  ii. n – אורך הקולוקציות הרצוי
  iii. t – threshold, סף הקובע את מספר המופעים המינימלי של כל קולוקציה בקורפוס
  iv. corpus – dataframe של Pandas המכיל את רשומות הקורפוס הרלוונטיות (מליאות או וועדות)
  v. type- מחרוזת המייצגת את סוג המדד: "frequency" או "tfidf" , כאשר "frequency" מייצג את תדירות המחרוזות בקורפוס ו"tfidf" מייצג את מדד הtf-idf שלהן.
- מדד tf-idf מחושב ע"י הנוסחה:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

כאשר:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$idf(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

$t$-מייצג מונח (קולוקציה באורך $n$)

$d$- מייצג פרוטוקול

$f_{t,d}$- מספר הפעמים שמונח $t$ מופיע בפרוטוקול $d$

למידע והסבר נוסף ראו פה: *wikipedia-tf-idf*

2. הדפיסו לקובץ את 10 הקולוקציות באורך 2 הכי נפוצות בכל אחד מהקורפוסים (מליאות וועדות בנפרד), עם סף t=5 לפי כל אחד מהמדדים.
3. הדפיסו לקובץ את 10 הקולוקציות באורך 3 הכי נפוצות בכל אחד מהקורפוסים (מליאות וועדות בנפרד), עם סף t=5, לפי כל אחד מהמדדים.
4. הדפיסו לקובץ את 10 הקולוקציות באורך 4 הכי נפוצות בכל אחד מהקורפוסים (מליאות וועדות בנפרד), עם סף t=5, לפי כל אחד מהמדדים.

הערות:

1. פונקציה זו יכולה להיות כחלק מהמחלקה שבניתם בסעיף קודם, או בנפרד, להחלטתכם.
2. על הפלט להיות מודפס לקובץ <u>אחד</u> בשם **knesset_collocations.txt** בפורמט הבא (ללא הסימנים <>):

Two-gram collocations:

Frequency:

Committee corpus:

<collocation number 1>

<…>
<collocation number 10>
<empty line>
Plenary corpus:
<collocation number 1>
<…>
<collocation number 10>
<empty line>
TF-IDF:
Committee corpus:
<collocation number 1>
<…>
<collocation number 10>
<empty line>
Plenary corpus:
<collocation number 1>
<…>
<collocation number 10>
<empty line>
Three-gram collocations:
Frequency:
Committee corpus:
<and so on>
Plenary corpus
<and so on>
<empty line>
Tf-IDF:
<and so on>
Four-gram collocations:
<and so on>

## שלב 3 – יישום מודלי השפה

1. ממשו פונקציה בשם *mask_tokens_in_sentences*

פונקציה זו תקבל רשימה של משפטים ותחליף אחוז מסויים מהטוקנים בכל משפט בטוקן "[*]" .

- <u>קלט:</u>
  i. sentences - רשימה של מחרוזות. כל מחרוזת היא משפט.
  ii. x – אחוז הטוקנים שיש למסך בכל משפט.
- <u>פלט:</u>
  רשימה של מחרוזות התואמת לרשימה שהתקבלה בקלט (באותו סדר), בהם x אחוז מהטוקנים בכל משפט הוחלפו בטוקן המיוחד [*]

2. בחרו 10 משפטים באופן רנדומלי מתוך קורפוס **הוועדות** ומסכו 10% מהטוקנים בהם בעזרת הפונקציה מסעיף 2.

a. הדפיסו את המשפטים המקוריים לקובץ בשם **original_sampled_sents.txt** , כך שכל משפט יהיה בשורה נפרדת, ללא שורות רווח.
b. הדפיסו את המשפטים הממוסכים לקובץ בשם **masked_sampled_sents.txt** , כך שכל משפט ממוסך יהיה בשורה נפרדת, ללא שורת רווח, באותו סדר כמו בקובץ המשפטים המקוריים.

3. חזו את הטוקנים הממוסכים בעזרת מודל השפה של ה**מליאות** וחשבו את ההסתברות לכל אחד מהמשפטים (אחרי שהשלמתם את החוסרים) בעזרת כל אחד משני מודלי השפה שבניתם.
   a. הדפיסו את התוצאות לקובץ בשם **sampled_sents_results.txt** באופן הבא:

original_sentence: <first original sentence as appeared in original_sampled_sents.txt file>
masked_sentence: <first masked sentence as appeared in masked_sampled_sents.txt file>
plenary_sentence: <the sentence with the generated tokens as was produced by the plenary LM>
plenary_tokens: <A list with the generated tokens, separated by a comma (“,”): token1,token2…>
probability of plenary sentence in plenary corpus: <log probability of the plenary sentence>
probability of plenary sentence in committee corpus: <log probability of the plenary sentence>
original_sentence: <second original sentence as appeared in original_sampled_sents.txt file>
<and so on>

הערה: ההדפסה של ההסתברויות צריכות להיות עם דיוק של 2 ספרות בלבד אחרי הנקודה.

4. חשבו את הperplexity הממוצע של מודל **המליאות** עבור כל הטוקנים הממוסכים בכל המשפטים מסעיף 3. לשם כך, עבור כל משפט, השתמשו בנוסחת הperplexity לTrigram , כפי שלמדתם בהרצאה, וקחו בחשבון רק את האינדקסים של הטוקנים הממוסכים, במקום את כל הטוקנים במשפט. גם פה ההסתברויות יחושבו בעזרת MLE עם אינטרפולציה ליניארית.
   a. הדפיסו את התוצאה לקובץ בשם: **perplexity_result.txt** ובו רק תוצאת החישוב עם דיוק של 2 ספרות בלבד אחרי הנקודה.
   b. כתבו בדו״ח את **התוצאה** שקיבלתם ואת **הנוסחה** שהשתמשתם בה.

## שלב 4 – שאלות סיכום

1. האם הקולוקציות הנפוצות ביותר בכל קורפוס, על פי מדד התדירות, יכולות לספר לנו משהו על התוכן והנושאים בהם הקורפוס עוסק? האם הופתעתם מהתוצאות שהתקבלו או שהן תאמו לציפיות שלכם? הסבירו.
2. ענו על שאלה 1, הפעם עבור מדד tf-idf.
3. האם ראיתם הבדלים בולטים בין הקולוקציות של שני המדדים הנ״ל? בין אם כן ובין אם לא הסבירו מדוע.
4. האם הגדלה או הקטנה של הסף t הייתה משפיעה על הקולוקציות שהדפסתם בשלב 2 (סעיפים 2-4)? הסבירו עבור כל אחד מהסעיפים, המדדים, ואופן השינוי (הגדלה/הקטנה).
5. האם קיבלתם משפטים הגיוניים בשלב 3 סעיף 3?
6. עד כמה ההשלמות של המודל בשלב 3 סעיף 3 קרובות למילים החסרות האמיתיות? פרטו.
7. הסבירו את המשמעות של נוסחת הperplexity שהשתמשתם בה ושל התוצאה שהתקבלה (בשלב 3 סעיף 4).
8. בהמשך לשאלה הקודמת, האם ביצועי המודל על פי מדד זה היו טובים על המשפטים שבחרתם? מדוע לדעתכם?

## הערות כלליות

1. אתם יכולים לעבוד בכל סביבת עבודה שנוחה לכם, אך הפתרון ייבדק בסביבת windows עם python 3.9 ועליכם לדאוג שהוא ירוץ בהצלחה בסביבה זו.
2. על הקוד שלכם להיות מסוגל להתמודד עם שגיאות עבור כל שלב בתהליך ולא לקרוס. השתמשו ב-Try Except blocks לפי הצורך.
3. שימו לב, בבדיקת תרגילי הבית בקורס ניתן משקל גדול מהניקוד הן על **הדו"ח**, ההסברים והידע שהפגנתם בחומר הנלמד והן על **הקוד**, אופן המימוש, יעילותו, קריאותו ועמידותו. בפרט, הרבה מהבדיקות הן אוטומטיות ולכן עליכם להקפיד על קוד תקין שרץ ללא שגיאות ועל עמידה **מדוייקת** בפלט הנדרש וביתר ההנחיות.
4. ניתן לשאול שאלות על התרגיל בפורום המיועד במודל. למעט מקרים אישיים מיוחדים, אין לשלוח שאלות הקשורות לתרגיל הבית במייל.
5. על אחריותכם לעקוב אחר הודעות הקורס במודל (בלוח ההודעות ובפורום) ולהיות מעודכנים במידה ויהיו שינויים בהנחיות.

## ספריות מותרות לשימוש

אתם יכולים להשתמש בpandas ובכל ספריה **סטנדרטית** של python.

אתם יכולים לחפש שם של ספריה ב https://docs.python.org/3/library/index.html על מנת לבדוק אם זו ספריה סטנדרטית. לא יהיה מענה על שאלות לגבי שימוש בספריות ספציפיות.

- למען הסר ספק, **json** היא ספרייה סטנדרטית של python.
- מומלץ להשתמש עבור כל פרוייקט בסביבה וירטואלית virtual environment חדשה משלו על מנת להיות בטוחים שאתם משתמשים רק בספריות מותרות ולמנוע קונפליקטים עם ספריות קודמות שהתקנתם בעבר. ראו מצגת על כך במודל.

## אופן ההגשה

1. ההגשה היא בזוגות בלבד.
2. עליכם להגיש קובץ zip בשם hw2_<id1>_<id2>.zip (כאשר <id1>,<id2> הם מספרי תעודות הזהות של הסטודנט הראשון והשני בהתאמה), המכיל את הקבצים הבאים:
   a. קובץ python בשם **knesset_language_models.py** המכיל את כל הקוד הנדרש כדי לממש את שלבים 1-3.
      i. - הקלט לקובץ הוא נתיב לקובץ ה-jsonl של הקורפוס שלכם ונתיב לתיקייה לשמירת קבצי הפלט.
      - הפלט יהיה שמירה של **כל** קבצי הפלט כפי שתוארו בשלבים 2,3 לתיקיית הפלט. (שימו לב שבשלב ההגשה התוכנית לא אמורה להדפיס שום דבר למסך).
      ii. על הקובץ לרוץ תחת הפקודה (ללא הסימונים <>):

```
python knesset_language_models.py <path/to/corpus_file_name.jsonl> <path/to/output_dir>
```

   b. קובץ text בשם **knesset_collocations.txt** כפי שתואר בשלב 2.
   c. קובץ בשם **original_sampled_sents.txt** כפי שתואר בשלב 3.
   d. קובץ בשם **masked_sampled_sents.txt** כפי שתואר בשלב 3.
   e. קובץ בשם **sampled_sents_results.txt** כפי שתואר בשלב 3.
   f. קובץ בשם **perplexity_result.txt** כפי שתואר בשלב 3.

g. קובץ ה**jsonl** של הקורפוס שלכם שיצרתם בתרגיל הקודם.
h. קובץ PDF בשם **<id1>_<id2>_hw2_report.pdf** ובו דו״ח המפרט על הקוד, על ההחלטות שקיבלתם במהלך העבודה על התרגיל ומענה על השאלות בשלב 4.
<u>אל תשכחו לציין בתחילת הדו״ח את שמותיכם **בעברית** ותעודות הזהות שלכם.</u>

יש להקפיד על עבודה עצמית, צוות הקורס יתייחס בחומרה להעתקות או שיתופי קוד, כמו גם שימוש בכלי AI דוגמת chatGPT.

ניתן לשאול שאלות על התרגיל בפורום הייעודי לכך במודל.

יש להגיש את התרגיל עד לתאריך 16.12.24 בשעה 23:59.

בהצלחה!