מבוא לעיבוד שפה טבעית

מטלה 12 – פתרון

אייל טרבלסי, ת.ז. 200953859

רומן ציפרון, ת.ז. 306591645

שאלה 1

סעיף	פרמטר	קובץ אימון	קובץ גולד
1	מופעי יוניגרם של סגמנטים	127884	11282
2	סוגי יוניגרם של סגמנטים	15986	3171
3	מופעי סגמנט-תג (זהה ל-1 כי לכל מופע של סגמנט יש תג)	127884	11282
4	סוגי סגמנט-תג	18143	3424

5. מדד העמימות עבור קובץ האימון: 1.13493056424
   מדד העמימות עבור קובץ הגולד: 1.07978555661
6. מדד העמימות בקובץ האימון גבוה ממדד העמימות בקובץ הגולד. כיוון שלקובץ הגולד יש עמימות מסויימת, נקבל חוסר דיוק עבור מילים עמומות כי תמיד אותן המילים יקבלו את אותו התיוג במתייג הבסיסי. לעומת זאת, במתייג HMM אנחנו נצפה שהקשרן של המילים יעזרו באופן משמעותי להפגת העמימות, אך עדיין כנראה שתהיה פגיעה מסויימת בדיוק התיוג בגלל הפער בין כמות העמימות בקובץ האימון לבין הגולד.

שאלה 2

1. סכמת הפרמטרים במודל זה היא $P(t_i \mid w_i)$ – משמע הסתברות של תיוג בהנתן מילה.

2. נוסחת המשערך לפרמטר הינה

$$P(t_i \mid w_i) = \frac{\mathrm{Count}(w_i \text{ with tag } t_i)}{\mathrm{Count}(w_i)}$$

3. בהנתן כי מספר המשפטים בקורפוס האימון הוא N_1, מספר המשפטים בקורפוס הבדיקה הוא N_2 אורך כל משפט הוא $O(m)$, גודל קבוצת הסגמנטים הוא s וגודל קבוצת התגים הוא t
   - סיבוכיות האימון היא $O(N_1 m)$ שכן עוברים בצורה סדרתית על כל המשפטים, לכל משפט עוברים על כל המילים ולכל מילה שומרים במילון או בטבלת גיבוב את מספר הפעמים אותה המילה הופיעה עם התיוג הנתון – סך הכל $O(N_1 m)$. המילון המתקבל הוא בגודל s. לאחר מכן יש למצוא לכל סגמנט במילון מה היה התיוג הכי שכיח בשבילו, לכן לכל סגמנט נעבור על $O(t)$ תגים, לכן $O(s \cdot t)$ וסך הכל $O(N_1 m + s \cdot t)$, אך כיוון שבהכרח גודל המילון קטן מ-$N_1 m$, נקבל כי סיבוכיות האימון הכוללת היא $O(N_1 m)$.
   - סיבוכיות התיוג היא $O(N_2 m)$ שכן עוברים בצורה סדרתית על כל המשפטים, לכל משפט עוברים על כל המילים וכל מילה מתייגים בתיוג שמצאנו בשלב האימון.

4. ה-macro-avg הוא $A = 0.83061513916$ ו-$All = 0.11$.

שאלה 3

1. נסמן: הקלט הוא סדרת מילים $w_1, \dots, w_n$, הפלט הוא סדרת תיוגים $t_1, \dots t_n$, אוסף כל התיוגים האפשריים הוא T.
   פונקציית המטרה היא
$$f(w_1^n) = \text{argmax}_{\{t_1^n | t_i \in T\}} P(t_1^n | w_1^n)$$
   באמצעות נוסחת בייז נקבל
$$f(w_1^n) = \text{argmax}_{\{t_1^n | t_i \in T\}} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$
   בשלב זה ניתן להשמיט את המכנה שכן הוא אינו משפיע על המקסימיזציה. אז נקבל
$$f(w_1^n) = \text{argmax}_{\{t_1^n | t_i \in T\}} P(w_1^n | t_1^n) P(t_1^n)$$

   במודל מסדר ראשון נקרב את ההסתברויות באופן הבא
$$P(w_1^n | t_1^n) \approx \prod_{i=1}^{n} P(w_i \mid t_i)$$

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1})$$

עבור הנוסחה השניה נגדיר $t_0 =< S >$ על מנת לקבל נוסחה מוגדרת היטב.

2. במודל זה קיימות שתי נוסחאות של פרמטרים. הסתברות של מילה בהנתן תג $P(w_i \mid t_i)$, והסתברות של תג בהנתן התג שמופיע לפניו $P(t_i \mid t_{i-1})$.

3. נוסחאות המשערכים הן

$$P(w_i \mid t_i) = \frac{\text{Count}(w_i \text{ with tag } t_i)}{\text{Count}(t_i)}$$

$$P(t_i \mid t_{i-1}) = \frac{\text{Count}(\text{tag } t_{i-1} \text{ followed by tag } t_i)}{\text{Count}(t_{i-1})}$$

4. בהנתן כי מספר המשפטים בקורפוס האימון הוא N_1, מספר המשפטים בקורפוס הבדיקה הוא N_2 אורך כל משפט הוא $O(m)$, גודל קבוצת הסגמנטים הוא s וגודל קבוצת התגים הוא t מתקיים כי סיבוכיות האימון היא $O(N_1 m + s \cdot t + t^2)$ שכן עוברים בצורה סדרתית על כל המשפטים, לכל משפט עוברים על כל המילים ולכל מילה שומרים במילון או בטבלת גיבוב את מספר הפעמים אותה המילה הופיעה עם התיוג הנתון ושומרים במילון אחר את מספר הפעמים אשר כל רצף של שני תגים הופיע, סך הכל $O(N_1 m)$. לאחר מכן לכל סגמנט במילון הראשון מוצאים את התג הכי שכיח עבורו, סך הכל $O(s \cdot t)$ וכן עוברים על כל רצף של שני תגים במילון השני, סך הכל $O(t^2)$. כיוון שגודל שני המילונים חסום על ידי $N_1 m$ (שכן מספר המילים ומספר הרצפים של שני תגים יהיה לכל היותר כמספר כלל המילים בקורפוס), נקבל כי סיבוכיות האימון הכוללת היא $O(N_1 m)$.

5. בהנתן אותן ההנחות כמו בסעיף הקודם, סיבוכיות התיוג היא $O(N_2 m t^2)$ שכן לכל משפט אנו מריצים אלגוריתם ויטרבי שהסיבוכיות שלו היא $O(mt^2)$.

6. ה-macro-avg הוא $A = 0.348697039532$ ו-$All = 0.134$

7. הוספנו החלקה פשוטה למילים שלא נראו: הנחנו שכמות המילים שאינן נראו שווה לכמות המילים שנראו פעם אחת בלבד. עבור כל מילה באימון שנראתה פעם אחת הוספנו מופע אחד למילה UNK עם התג הנתון.
בשלב התיוג, עבור כל מילה שלא נראתה באימון אנו מחפשים במקומה את המילה UNK עבור חלק הדיבר הרלוונטי באלגוריתם ויטרבי. החלקה זו נותנת את ה-macro-avg: $A = 0.892306328665$ ו-$All = 0.268$ – שיפור ניכר לתוצאה ללא החלקה.

8. הוספנו החלקה פשוטה למעברים שלא נראו: הנחנו שכמות המעברים שאינם נראו שווה לכמות המעברים שנראו פעם אחת בלבד. עבור כל מעבר באימון שנראה פעם אחת בלבד הוספנו מופע אחד למעבר מהתג UNK לתג הנתון אליו עוברים במעבר הנוכחי. בשלב התיוג, במהלך אלגוריתם ויטרבי אם אנחנו מכפילים בהסתברות למעבר שלו נתקלנו בו, נחליף הסתברות זו במעבר מהתג UNK לתג הנבדק באיטרציה הנוכחית. החלקה זו נותנת את ה-macro-avg: $A = 0.900904095019$ ו-$All = 0.26$– נותן שיפור מינורי ב-A וגורע באופן לא משמעותי מערך הפרמטר All.

<u>שאלה 4</u>

1. מטריצת הבילבול מצורפת בקובץ exps/confusion.matrix. שלושת השגיאות הנפוצות ביותר היו:
   81 times NN was tagged as VB
   71 times NN was tagged as NNT
   64 times NNT was tagged as NN
2. התיוג הנכון של המשפט הוא
   yyDOT NN IN NN H AT VB yyCM NN VB JJ NN
   אישה נעלה נעלה נעלה, נעלה את הדלת לפני בעלה.

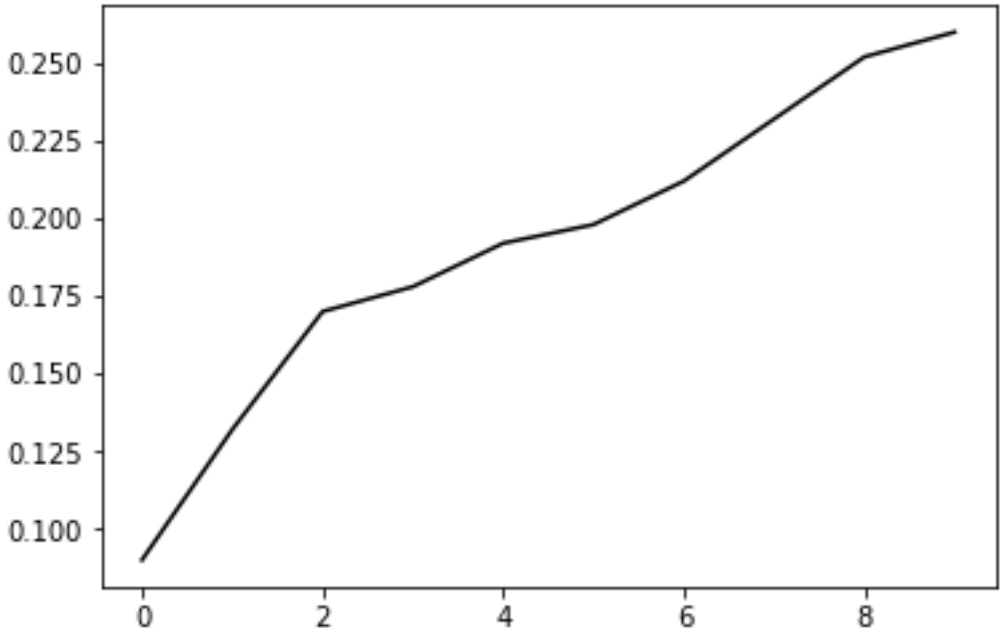
הרצנו על משפט זה את מתייג מסדר ראשון עם החלקות אשר אומן על הקובץ heb-pos.train. התיוג שהתקבל הוא:
yyDOT NN IN NN H AT VB yyCM NNP NNP NNP NN

המתייג צדק בכל המילים פרט סדרת המופעים של "נעלה" לפני הפסיק הראשון. הדיוק הוא 75% שכן תוייגו נכון 9 מילים מתוך 12.

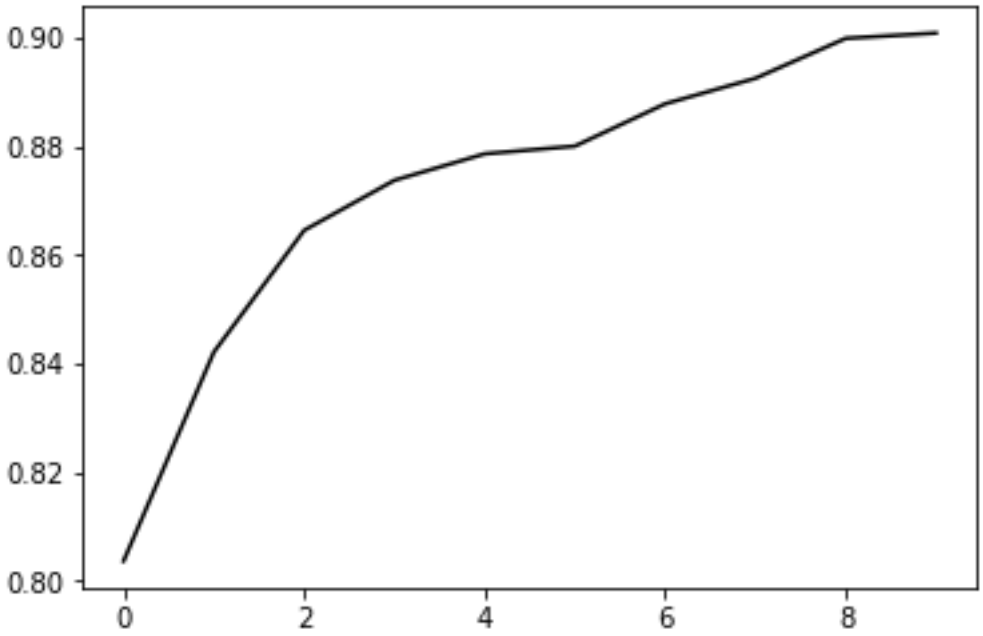
מבדיקה עולה כי המילה "נעלה" כלל לא הופיעה בקורפוס האימון, ומכאן ניתן להסיק שהמופע האחרון של המילה תוייג נכון בזכות החלקה של transitions שכן כנראה קיימת מובהקות להופעת פועל לאחר פסיק.

האינפורמציה החסרה במודל היא אוצר מילים ותיוגים אפשריים למילים אלו. זאת אומרת לדעתנו הסיכוי לתייג נכון את כל המשפט היה עולה אם המילה "נעלה" הייתה מופיעה בקורפוס האימון עם כל חלקי הדיבר איתם המילה נראתה במשפט הנ"ל.

3. גרף ערך ה-ALL:



גרף ערך ה-A:



ניתן לראות תפוקה שולית פוחתת. משמע שאם נאמן על פני קורפוס גדול יותר אז הדיוק ישתפר, אך נזדקק ליותר ויותר (באופן אקספוננציאלי) טקסט מתוייג על מנת לשפר את איכות המתייג.

6. תיאור הקוד: תוכנית האימון (train.py) מקבלת קובץ אימון וקוראת לפונקציה utils.analyzeFileFull אשר עוברת על הקובץ הנתון שורה-שורה, מחשבת ומחזירה את המשתנים הבאים:

segmentTagsDict – מילון אשר ממפה כל סגמנט הנצפה בקורפוס למילון בו המפתחות הן חלקי הדיבר שנצפו למילה הנתונה והערכים הם מספר הפעמים שכל חלק דיבר נצפה. מילון זה משמש ליצירת קובץ ה-lex.

unigramDict – מילון הממפה תג לכמות הפעמים שתג זה נצפה בקורפוס (כולל הסימנים של תחילת וסוף משפט). מילון זה משמש לבניית חלק ה-1-grams בקובץ ה-gram.

bigramDict – מילון זה ממפה זוגות של סגמנטים לכמות הפעמים שכל זוג נצפה בקורפוס בסדר הנתון. מילון זה משמש לבניית חלק ה--2 grams בקובץ ה-gram.

unigramCount – מספר המופעים הכולל של סגמנטים בקורפוס (כולל חזרות). מספר זה משמש לחישוב ההסתברויות בחלק ה-1-grams בקובץ ה-gram.

בהנתן ארבעת המשתנים הנ"ל אנו בונים את קובץ ה-lex (וקובץ ה-gram עבור model=2).
עבור מודל 1, קובץ ה-lex הנוצר מכיל בכל שורה סגמנט ואת התג הכי נפוץ בשבילו. עבור מודל 2 קובץ ה-lex הוא בפורמט הדרוש לפי הנספח ובשני קבצי הפלט כל הערכים מנורמלים כך שסכום ההסתברויות של הערכים הרלוונטיים (לדוגמא כל ההסתברויות שמותנות על אותו המאורע) יהיה 1 ונקבל התפלגות חוקית.
אם מצויין פרמטר ההחלקה, אז אנחנו מוסיפים לקבצי הפלט גם סימני UNK כפי שמוסבר בשאלה 3 סעיפים 7,8.

בחלק ה-decode אנו קוראים את קבצי ה-lex (וה-gram עבור מודל 2).
אם אנחנו במודל 1 אז אנו יוצרים מילון אשר ממפה מילה לתג הכי שכיח בשבילה, ואז עוברים על קובץ ה-test ומתייגים כל מילה לפי המילון. אם המילה לא נמצאה במילון אז נתייג אותה עם NPP.

עבור מודל 2 אנו בונים את המילון emissionProbabilityDict מתוך קובץ ה-lex, ואת המילון transitionProbabilityDict מתוך קובץ ה-gram.
לאחר מכן אנו עוברים על קובץ ה-test שורה-שורה, וכל פעם שנגמר משפט אנו קוראים לאלגוריתם ויטרבי על המשפט הנתון. האלגוריתם ממומש בפונקציה utils.viterbi אשר מקבלת את המשפט הנתון, קבוצת המצבים האפשרים (התגים) ושני המילונים שיצרנו ומפעילה את אלגוריתם ויטרבי כפי שנלמד בכיתה. עבור כל מילה או מעבר שלא נמצאו במילון המתאים, האלגוריתם מחפש את המילה UNK או המעבר מ-UNK לתג במילון המתאים, אם גם הם לא נמצאו, סביר להניח כי לא הפעלנו החלקה ובמקרה זה ההסתברות שתוחזר תהיה אפס.
במהלך ביצוע האלגוריתם אם לאחר שסיימנו לסרוק את המשפט הנתון קיבלנו כי כל ההסתברויות במצב האחרון הן 0 (או $-\infty$ עבור הסתברות לוגריתמית), אז נתייג ב-NNP את כל המילים עד למילה הכי מאוחרת במשפט עבורה יש הסתברות שאינה אפס, ומכאן נמשיך את ה-backtracking של האלגוריתם כרגיל, כך נמנע מלתייג משפט שלם ב-NNP כאשר יש בו מעבר אחד שאינו ידוע ונקבל לפחות דיוק חלקי.

רכיב ה-evaluation (evaluate.py) סורק את קבצי ה-test המתוייג וה-gold ולכל סגמנט בודק את נכונות התיוג שלו מול קובץ הגולד. בסופו של דבר מתקבל כפלט הדיוק של כל משפט והדיוק הכללי (macro-avg) כדרוש בהגדרות.

הקושי העיקרי במשימת התיוג הוא שדרוש קורפוס גדול מאוד על מנת לדייק כמה שיותר בתהליך התיוג. כיוון שעברית היא שפה עשירה במובן המורפולוגי (זאת אומרת, קיימות הרבה דרכים לגזור מילים בהתאם להקשרן), יש לצפות לנפח מסויים של עמימות מורפולוגית בקורפוסים. כמו כן, כיוון שסדר המילים בעברית אינו מאוד קשיח אז הגיוני להניח שנתקשה לתייג נכונה אם נסתמך באופן משמעותי על הסתברויות מעברים מתג לתג.

על מנת לשפר את איכות המתייג, כדאי לנסות ליישם מתייג עם סדר הגבוה יותר מכפי שיושם בתרגיל, לדוגמא מתייג מסדר שני. סביר להניח כי מהסתכלות של שתי מילים אחורה ולא אחת נוכל ללמוד הרבה יותר על ההקשר של המילה ולהסיק מסקנות יותר נכונות.

מסיבות אלו (פוטנציאל רב לעמימות והצורך בניתוח מסדר גבוה יותר) נרצה קורפוסים גדולים ועשירים מבחינת אוצר מילים ותחביר. נרצה שמילים בעלות עמימות מורפולוגית יופיעו על כל משמעותן כך שנוכל ללמוד משמעויות אלו על פי ההקשר, וגם שיהיו בקורפוס כמה שיותר סדרות שונות של k (לדוגמא 3 עבור מתייג מסדר שני) מילים כך שנלמד מהן הקשרים בצורה נכונה.

יתכן כי להגדיל את הסדר של המודל (מספר התגים שזוכרים אחורה) יכול להגדיל משמעותית את דיוק התיוג אך יש להזהר מ-overfitting שכן מקורפוס נתון נלמד רק את סדרות התיוגים שיש בו.

דבר נוסף שניתן לעשות על מנת לשפר את הדיוק הוא לשפר את איכות ההחלקה. בפרוייקט זה מימשנו החלקה נאיבית ביותר אשר הביאה לשיפור ניכר בתוצאה, אך וודאי ניתן לשפר את התוצאה עוד עם החלקה מתוחכמת יותר.

לסיכום, על מנת לשפר את איכות התיוג כדאי לעשות את הדברים הבאים:

- לספק קורפוסים מתוייגים גדולים לאימון, עדיף ממקורות שונים עם הקשרים, אוצר מילים ורמות שפה (לדוגמא שפה גבוהה, שפה עיתונאית, שפה יומיומית) שונים.
- לנסות לשנות את המודל לסדר שני או שלישי.
- לשפר את ההחלקה.