תרגיל בית 1 – עיבוד שפה טבעית – חלק רטוב

תיאור המשימה

בתרגיל בית זה תממשו מודל MEMM (כפי שנלמד בשבוע 4), המשמש לתיוג סדרתי של חלקי דיבר במשפט, תבצעו משימות עיבוד שפה על נתונים ותנתחו את טיב הצלחתכם.

סגל הקורס ממליץ להשתמש בשפת python, אולם תוכלו להשתמש בכל שפת תכנות שתחפצו. יש לקחת בחשבון כי נפח התרגיל נקבע על סמך ההנחה שאתם עובדים עם שפה עילית, המאפשרות גמישות בעבודה עם נתונים בצורה נוחה יותר.

נתונים:

הסבר על הקבצים המצורפים –

1. train.wtag – קובץ המכיל 5000 משפטים מתוייגים. עליכם להשתמש בקובץ זה בשלב האימון (הסבר בהמשך)
   דוגמא לפורמט של הנתונים – בתרגול 3 ראינו את הדוג' הבאה -
   the dog barks -> D N V STOP
   בפורמט הנוכחי הדוג' תראה כך:
   the_D dog_N barks_V ._.
   שימו לב כי כל המשפטים (או רובם) מסתיימים בנקודה ('.'), אולם לא כל נקודה סוגרת משפט.
   הסכימה (scheme) לפיה המשפטים מתוייגים נקראת Penn Treebank, וניתן למצוא עליה עוד אינפורמציה כאן.
2. test.wtag – קובץ המכיל 1000 משפטים מתוייגים, בפורמט זהה לפורמט של הקובץ הקודם.
3. comp.words – קובץ המכיל 1000 משפטים לא מתוייגים. המשפטים מופיעים בצורת הטבעית, לדוג':

the dog barks .

אימון (Train) :

כאמור את הערכת הפרמטרים תעשו על הקובץ train.wtag. אתם נדרשים לבנות שני מודלים:

1. מודל *בסיס*, המורכב מסט מאפיינים (Features) בסיסי – $f_{100}, f_{103}, f_{104}$ כפי שהוגדרו בהרצאות הוידאו [(Ratnaparkhi, 96)] ומצורפים לנוחיותכם בסוף מסמך זה.
2. מודל *מורכב*, הכולל את המאפיינים $f_{100} - f_{105}$, וכן מאפיינים התופסים מספרים ומילים המכילות אותיות גדולות (Capital letters). את אלו לא נגדיר כאן באופן מפורש, אתם רשאים להגדיר אותם כרצונכם.

בסיכום שתכינו (עוד על כך בהמשך) יש לציין את סוג המאפיינים בהם השתמשתם (למשל באופן בו מתוארת f_{100} בהרצאה) עבור כל אחד מן המודלים, ואת מספר המאפיינים שמכל סוג (לדוג' 217 מאפיינים מסוג מילה+תג). את המאפיינים שבחרתם להוסיף למודל המורכב יש להגדיר במפורש.

כל שיפור שהכנסתם למודל (לדוג' קיצוץ של מאפיינים ש"לא הופיעו מספיק") צריך להיות מוסבר היטב, כולל המוטיבציה לבצע אותו. אם השתמשתם בחבילת קוד חיצונית, יש לפרט במפורש היכן ולאיזו מטרה.

בנוסף, יש לפרט כמה זמן לקח לאמן כל מודל (וכן מפרט בסיסי של החומרה עליה הרצתם)

הסקה (Inference):

ההסקה תתבצע ע"י אלגוריתם ויטרבי כפי שנראה בהרצאה ובתרגול. יש לציין כל חריגה ושיפור שהכנסתם לאלגוריתם הבסיסי, את המוטיבציה לחריגה וכן את תרומתה.

מבחן (test) :

לכל אחד מן המודלים, יש לבצע הסקה (Inference) על הקובץ test.wtag, ולדווח את תוצאות הדיוק (accuracy) ברמת מילה, כפי שנעשה בהרצאה. התייחסו להבדל בביצועים בין המודלים השונים, והעלו מספר סיבות שעשויות לגרום להבדלים אלו.

הכינו ניתוח המכיל מטריצת בילבול בין 10 התגים עבורם המודל טועה הרבה, והציעו דרך לשפר את המודל(ים) על מנת להתמודד ישירות עם בילבול בין שני תגים.

בנוסף, אנא ציינו כמה זמן לקח לתייג את הקובץ לפי כל אחד מהמודלים (וכן מפרט בסיסי של החומרה עליה הרצתם את הקוד)

תחרות:

לכל אחד מן המודלים, יש לבצע הסקה (Inference) על הקובץ comp.words (אשר אינו כולל תיוגים), ולכתוב את תוצאות התיוג לתוך קובץ חדש בפורמט wtag (כמו קבצי האימון) (שמות הקבצים הרצויים מופיעים בהמשך).
לדוג', עבור המשפט:

the dog barks .

יש לבצע הסקה, שתיתן לכם את תוצאות התיוג. בהנחה שהתיוג שהתקבל הוא ". D N V", יש לכתוב עבור שורה זאת את השורה הבאה –

the_D dog_N barks_V ._.

שימו לב שהקבצים שאתם מגישים לא צריכים לכלול כוכביות וסימני סוף משפט, ושסדר המשפטים (הלא מתוייגים) בקובץ המקורי זהה לסדר המשפטים בקובץ הפלט.

יש לתאר במפורש מה עשיתם כדי לקבל את התוצאות שקיבלתם (שינויים שביצעתם בלמידה, בהסקה וכו').

בנוסף, אתם מתבקשים לכתוב: תחזית של אחוז הדיוק שאתם צופים לקבל, וכן להסביר מדוע עשוי להיות הבדל בין הדיוק על קובץ ה test. הסברים חכמים אף עשויים לקבל בונוס.

קוד חיצוני המותר לשימוש:

החבילות הסטנדרטיות בשפה בה בחרתם, וכן חבילות שעושות אופטימזציה על פעולות וקטוריות.

לדוג', בשפת python החבילות בהן מותר להשתמש הן scipy ו numpy בלבד.

בשלב האימון ניתן (ומומלץ, ואולי אפילו הכרחי) להשתמש בחבילה הממשת את אלגוריתם LBFGS עליו דיברנו בהרצאה, בתנאי שהיא עושה אופטימזציה על פונ' המטרה וגרדיאנט שאתם מספקים לה.

למען הסר ספק - **אסור** להשתמש ב:

1. חבילות הממשות אלגוריתם ויטרבי.
2. חבילות הממשות MEMM.
3. חבילות העושות עיבוד על טקסט – ספירת חזרות, uni\tri\bi gram וכו'.

הגשה:

קובץ zip בלבד, בשם HW1-Wet_123456789_987654321.zip (עבור שני סטודנטים שמספרי הזהות שלהם 123456789 ו 987654321). הקובץ הנ"ל יכלול:

1. קובץ הסברים וניתוח תוצאות, הכולל בין היתר:
   a. שמות המחברים ות"ז
   b. הערות על אימון המודל, לכל אחד מן המודלים (לפי הדגשים בסעיף "אימון")
   c. הערות על אלגוריתם ההסקה (לפי הדגשים בסעיף "הסקה")
   d. ניתוח תוצאות על קובץ המבחן (לפי הדגשים בסעיף "מבחן")
   e. סיכום שלכם על המשימה
   f. ניתוח של מה שעשיתם בקובץ התחרות (לפי ההסברים כפי שמפורטים בסעיף "תחרות")
   g. הסבר על חלוקת העבודה בין שני חברי הקבוצה – איזה חלק עשה\ביצע\מימש כל אחד
2. קבצי הקוד של התרגיל. על הקוד להיות מתועד וקריא. בנוסף, הקוד צריך מסוגל לרוץ על כל מכונה שהיא. אנא צרו ממשק פשוט להרצת התוכנית המייצרת את קבצי התחרות.
3. קבצי התחרות – על קבצי התוצאות להיות בפורמט wtag (כפי שמפורט בחלק "אימון"), הכולל את המילים והתגים. על מנת להימנע מאי נעימות בנוגע לציון, אנא ודאו כי אם משמיטים את הקו התחתי והתגים מקובץ זה מקבלים בדיוק את אותם משפטים כמו comp.words (לפי אותו סדר). חוסר התאמה פירושו ציון 0 בחלק הזה.
   על שמות הקבצים להיות – (123456789 הוא ת"ז של אחד הסטודנטים)
   a. comp_m1_123456789.wtag – קובץ wtag שאומן על ידי המודל הבסיסי.
   b. comp_m2_123456789.wtag – קובץ wtag שאומן על ידי המודל המורכב.

הציון על התחרות יתבסס על $\max\{accuracy(comp\_m1), accuracy(comp\_m2)\}$, כלומר נסתכל על התוצאה הטובה יותר מבין שתי ההגשות.

<u>על קבצי התחרות להיות ניתנים לשחזור (Reproducible).</u> הדרישה היא שניתן יהיה לקחת את הקוד שהגשתם ולבנות באמצעותו קובץ זהה לחלוטין לקובץ שהגשתם. במקרים חריגים בודק התרגילים יבדוק את ההתאמה הנ"ל.

בסה"כ קובץ ההגשה צריך להיראות כך:

```
HW1-Wet_123456789_987654321.zip/
    report (.pdf,.docx, etc.)
    Code_Directory/
        ...
    comp_m1_123456789.wtag/
    comp_m2_123456789.wtag/
```

העתקות:

בשל אופי המשימה והמורכבות שלה, קל לבדוק העתקות של קטעי קוד \ קבצים מלאים. למען הסר הספק אנו מדגישים כי אין להעביר קוד בין סטודנטים, בין אם להגשה ובין אם לא. אין לקחת קטעי קוד מוכנים מהאינטרנט, ובכלל אין להסתמך על שום מקור אחר לקוד מלבד פרי יצירכם והחבילות החיצוניות אשר צוינו בסעיף הרלוונטי.

The Full Set of Features in [(Ratnaparkhi, 96)]

- Word/tag features for all word/tag pairs, e.g.,

$$f_{100}(h,t) = \begin{cases} 1 & \text{if current word } w_i \text{ is } \texttt{base} \text{ and } t = \texttt{Vt} \\ 0 & \text{otherwise} \end{cases}$$

- Spelling features for all prefixes/suffixes of length ≤ 4, e.g.,

$$f_{101}(h,t) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in } \texttt{ing} \text{ and } t = \texttt{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{102}(h,t) = \begin{cases} 1 & \text{if current word } w_i \text{ starts with } \texttt{pre} \text{ and } t = \texttt{NN} \\ 0 & \text{otherwise} \end{cases}$$

The Full Set of Features in [(Ratnaparkhi, 96)]

- Contextual Features, e.g.,

$$f_{103}(h,t) = \begin{cases} 1 & \text{if } \langle t_{-2}, t_{-1}, t \rangle = \langle \text{DT, JJ, Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$f_{104}(h,t) = \begin{cases} 1 & \text{if } \langle t_{-1}, t \rangle = \langle \text{JJ, Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$f_{105}(h,t) = \begin{cases} 1 & \text{if } \langle t \rangle = \langle \text{Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$f_{106}(h,t) = \begin{cases} 1 & \text{if previous word } w_{i-1} = \textit{the} \text{ and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{107}(h,t) = \begin{cases} 1 & \text{if next word } w_{i+1} = \textit{the} \text{ and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$