עיבוד שפה טבעית - תרגיל בית 1

תיאור המשימה

בתרגיל בית זה תממשו מודל MEMM (כפי שנלמד בשבוע 3), המשמש לתיוג סדרתי של חלקי דיבר במשפט (Part of Speech Tagging), תבצעו משימות עיבוד שפה על נתונים אמיתיים ותנתחו את טיב הצלחתכם.

סגל הקורס ממליץ להשתמש בשפת Python 3, אולם תוכלו להשתמש בכל שפת תכנות שתרצו. יש לקחת בחשבון כי נפח התרגיל נקבע על סמך ההנחה שאתם עובדים עם שפה עילית, המאפשרת גמישות בעבודה עם נתונים בצורה נוחה יותר.

בתרגיל תידרשו לממש שני מודלים – מודל 1 (הגדול) ומודל 2 (הקטן), לטובת כל מודל מצורפים בקובץ `data.zip` קבצי הנתונים הרלוונטיים לאותו מודל בלבד.

לאורך התרגיל הכוונה במדד דיוק (Accuracy) היא לדיוק ברמת המילה. כלומר, שרשור כל הפרדיקציות של המודל שלכם והתיוגים האמיתיים על פני כל המשפטים, וחישוב דיוק בין שתי הרשימות.

הסבר על מבנה הציון בתרגיל:

- **45%** - מימוש מלא של מודל 1 (הגדול) כפי שיפורט בהמשך ואימונו על קובץ ה-`train1.wtag`, ועמידה בסף אחוז הדיוק (Accuracy) המינימלי של 90% על קובץ ה-`test1.wtag`.
- **20%** - תחרות מבוססת Accuracy בתיוג קובץ התחרות המתאים למודל 1 (הגדול) `comp1.words`
- **20%** - מימוש ואימון מודל 2 (הקטן) על קובץ ה-`train2.wtag`, תיוג קובץ התחרות המתאים למודל 2 (הקטן) `comp2.words` – **גם תחרות זו מבוססת Accuracy.**
- **15% - כתיבת דו"ח תמציתי (עד 3 עמודים) אשר יכלול את הסעיפים הנדרשים ועמידה בתנאי פורמט ההגשה (יפורטו בהמשך המסמך)**

נתונים:

הסבר על הקבצים המצורפים –

1. `train1.wtag` – קובץ המכיל 5000 משפטים מתויגים. עליכם להשתמש בקובץ זה בשלב האימון
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 מתויגים נקראת Penn Treebank, וניתן למצוא עליה עוד אינפורמציה כאן.
2. `test1.wtag` – קובץ המכיל 1000 משפטים מתויגים, בפורמט זהה לפורמט של הקובץ הקודם.
עליכם להשתמש בקובץ זה בשלב המבחן של מודל 1 (הגדול)
3. `comp1.words` – קובץ המכיל 1000 משפטים לא מתויגים. המשפטים מופיעים בצורתם הטבעית, לדוג':

the dog barks .

מודל 2 (הקטן):

1. `train2.wtag` – קובץ המכיל 250 משפטים מתויגים בפורמט זהה לפורמט קובץ האימון ממודל 1 (הגדול) המתואר לעיל. עליכם להשתמש בקובץ זה (`train2.wtag`) בשלב האימון של מודל 2 (הקטן).
2. `comp2.words` – קובץ המכיל 1000 משפטים לא מתויגים. המשפטים מופיעים בצורתם הטבעית, לדוג':

the dog barks .

אימון (Train) :

מודל 1 (הגדול):

כאמור, אמידת וקטור הפרמטרים תיעשה על הקובץ `train1.wtag`.

תידרשו לממש מודל המורכב מסט מאפיינים (Features$f_{100} - f_{107}$, כפי שהוגדרו בהרצאות הוידאו [(Ratnaparkhi, 96)] ומצורפים לנוחיותכם בסוף מסמך זה. בנוסף, אנא ממשו מאפיינים התופסים מספרים ומילים המכילות אותיות גדולות (Capital letters). את אלו לא נגדיר כאן באופן מפורש, אתם רשאים להגדיר אותם כרצונכם. כמו כן, אתם רשאים להגדיר מאפיינים נוספים כרצונכם, אך לא להחסיר מסט המאפיינים הנדרש.

מודל 2 (הקטן):

כאמור, אמידת וקטור הפרמטרים תיעשה על הקובץ `train2.wtag`.

במודל זה, אתם רשאים להגדיר ולממש מאפיינים כרצונכם, כלומר ניתן להשתמש בכל קומבינציה של מאפיינים שמימשתם עבור מודל 1 (אך לשים לב שהאימון מתבצע על הקובץ המתאים) וכאלו שאתם מוסיפים עבור מודל 2.

בדו"ח שתכינו (עוד על כך בהמשך) יש לציין את סוג המאפיינים בהם השתמשתם (למשל באופן בו מתוארת f_{100} בהרצאה) בכל מודל, ואת מספר המאפיינים מכל סוג (לדוג' 217 מאפיינים מסוג מילה+תג).
את המאפיינים שבחרתם להוסיף לכל מודל יש להגדיר במפורש.

<u>יש לציין ולהסביר כל שיפור שהכנסתם למודלים הסופיים (לדוג' קיצוץ של מאפיינים ש"לא הופיעו מספיק")</u>, כולל המוטיבציה לבצע אותו.

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

הסקה (Inference):

עבור שני המודלים, ההסקה תתבצע ע"י אלגוריתם Viterbi כפי שנראה בהרצאה ובתרגול.
<u>יש לציין כל חריגה</u> ושיפור שהכנסתם לאלגוריתם הבסיסי, את המוטיבציה לחריגה וכן את תרומתה.

מבחן (Test):

עבור מודל 1 (הגדול) יש לבצע הסקה (Inference) על הקובץ `test.wtag`, ולדווח את תוצאות הדיוק(Accuracy) ברמת מילה.

הכינו ניתוח המכיל Confusion Matrix בין 10 התגים עבורם המודל טועה הכי הרבה, והציעו דרך לשפר את המודל על מנת להתמודד ישירות עם בילבול בין שני תיוגים נבחרים מתוך הטבלה.

עבור מודל 2 (הקטן) לא ניתן קובץ מבחן ייעודי, יש לחשוב כיצד להתמודד עם בעיות בחינת ביצועי המודל ומיעוט דוגמאות האימון ולתאר את דרך התמודדותכם בדו"ח בקצרה.

תחרויות:

לכל אחד מן המודלים, יש לבצע הסקה (Inference) על קובץ התחרות המתאים (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 סימני כוכביות ו STOP, 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 התחרות וה-test. הסברים חכמים אף עשויים לקבל בונוס.

סביבת עבודה:

על הפרויקט לרוץ על המכונה שניתנה לכם בסביבה TBD.

הקוד שסופק לכם

הקוד שסופק לכם מכיל את הקבצים הבאים:

preprocessing.py – קובץ המכיל את יצירת הפיצרים וההיסטוריות. ניתן לכם קוד הממשמש את קבוצת הפיצרים 100, אתם נדרשים לממש את יתר הפיצרים כפי שתואר בפרקים של מודלים 1 ו2.

Optimization.py – קובץ המכיל את הנדרש לאופטימיזציה. את קובץ זה אתם לא נדרשים לשנות, אך יכולים לעשות זאת אם תרצו.

Inference.py – קובץ בו אתם נדרשים לממש את הפונקציה memm_viterbi וכל פונקציית עזר שתרצו, הממשמשת את אלגוריתם ויטרבי כפי שתואר בפרק inference.

main.py – קובץ המריץ את התכנית אותה כתבתם ומוציא כפלט את קובץ התיוגים.

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. (מותקנות בסביבה הסטנדרטית המצורפת המוזכרת מעלה)

בשלב האימון ניתן (מומלץ, ואולי אפילו הכרחי) להשתמש בחבילה הממשת את אלגוריתם LBFGS (לדוגמא ב-scipy), בתנאי שהיא עושה אופטימזציה על פונ' המטרה וגרדיאנט שאתם מספקים לה.

למען הסר ספק - **אסור** להשתמש ב:

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

הגשה:

קובץ zip בלבד, בשם HW1_123456789_987654321.zip (עבור שני סטודנטים שמספרי הזהות שלהם 123456789 ו 987654321). הקובץ הנ"ל יכלול:

1. **דו"ח קצר** (עד 3 עמודים בשם report_123456789_987654321.pdf) המכיל הסברים תמציתיים, דיווח וניתוח תוצאות, הכולל:
   a. שמות המחברים ות"ז
   b. אימון - דיווח אחוז דיוק (Accuracy) והערות על אימון כל מודל (לפי הדגשים בסעיף "אימון")
   c. הסקה - הערות על אלגוריתם ההסקה (לפי הדגשים בסעיף "הסקה")
   d. מבחן - דיווח אחוז דיוק (Accuracy) ו-Confusion Matrix על קובץ המבחן עבור מודל 1 (הגדול) (לפי הדגשים בסעיף "מבחן").
   יש לצרף צילום מסך מתהליך המבחן של מודל 1 המכיל הדפסה של אחוז הדיוק שהתקבל.
   הסבר קצר על דרך התמודדותכם עם בחינת ביצועי מודל 2 (הקטן).
   e. תחרות - הסבר קצר על שיפורים שעשיתם למודלים עבור תיוג קבצי התחרות (לפי ההסברים כפי שמפורטים בסעיף "תחרות")
   f. הסבר קצר 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 להיות מסוגל לרוץ על כל מכונה שהיא. אנא כתבו ממשקי הרצה פשוטים לאימון, מבחן וייצור קבצי התחרות המתויגים. על קבצי הקוד להיות בתיקייה code.
3. **קבצי התחרות מתויגים** – על קבצי התוצאות להיות בפורמט wtag (כפי שמפורט בחלק "אימון"), הכולל את המילים והתגים. על מנת לוודא נכונות ולהימנע מאי נעימות בנוגע לציון, אנא ודאו כי אם משמיטים את הקו התחתי והתגים מהקבצים המתויגים שאתם מגישים, מקבלים בדיוק את אותם משפטים (ולפי אותו סדר) ובאותו פורמט כמו בקובץ התחרות המתאים. חוסר התאמה פירושו ציון 0 בחלק הזה.
   על שמות הקבצים להיות – (123456789 הוא ת"ז של אחד הסטודנטים)
   a. comp_m1_123456789_987654321.wtag – קובץ wtag שתויג ע"י מודל 1 (הגדול).
   b. comp_m2_123456789_987654321.wtag – קובץ wtag שתויג ע"י מודל 2 (הקטן).
4. **ממשק לתיוג קבצי התחרות** - על קבצי התחרות להיות ניתנים לשחזור (Reproducible).
   הדרישה היא שניתן יהיה לקחת את הקוד והמודלים המאומנים שהגשתם ולייצר באמצעותם קבצי תחרות מתויגים זהים לחלוטין לקבצים שהגשתם.
   לטובת שחזור הקבצים, יש לכתוב ממשק הרצה פשוט, בקובץ נפרד בעל השם generate_comp_tagged.py–
   להרצת Inference בלבד על המודלים המאומנים ויצירת קבצי התחרות המתויגים ע"י כל מודל.

לנוחיותכם, הקובץ check_submission.py יבדוק את תקינות ההגשה שלכם. יש להריצו מתיקייה בה נמצא הקובץ HW1_ID1_ID2.zip וכן התיקייה data.

<u>העתקות:</u>

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 להעתיק 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.,

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

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

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

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

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