**עיבוד שפה טבעית – תרגיל בית 3**

**תיאור המשימה**

בתרגיל בית זה תממשו Dependency Parser (כפי שנלמד בשבוע 7) מבוסס רשתות נוירונים ללמידת דקדוק תלויות, ותנתחו את טיב הצלחתכם. תוכלו להסתמך על המודל שהוצג במאמר של [Kiperwasser and Goldberg](https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00101/43362/Simple-and-Accurate-Dependency-Parsing-Using), אך אין חובה כזו.

סגל הקורס ממליץ להשתמש בספריית PyTorch למימוש ואימון רשתות נוירונים.  
לצורך אימון הרשתות בזמן סביר, לרשותכם עומדות מכונות GPU בסביבת ה-Azure.

**הסבר על מבנה הציון בתרגיל:**

* **40%** - מימוש מלא של מודלים כפי שיפורטו בהמשך, אימונם על קובץ ה-train.labeled, ועמידה בסף אחוז הדיוק (UAS) של 70% על קובץ ה-test.labeled.
* **20%** - כתיבת דו"ח תמציתי (עד 3 עמודים) אשר יכלול את הסעיפים הנדרשים ועמידה בתנאי פורמט ההגשה (יפורטו בהמשך המסמך)
* **20%** - תחרות מבוססת UAS על קובץ comp.unlabeled
* **20%** - חלק יבש (hw3-dry)  
    
  מצורף נספח בסוף הוראות התרגיל המציג דוגמא לחישוב אחוז הדיוק UAS.

**נתונים:**

הסבר על הקבצים המצורפים –

1. train.labeled – קובץ המכיל 5000 משפטים מתוייגים. עליכם להשתמש בקובץ זה בשלב האימון (הסבר בהמשך)
2. test.labeled – קובץ המכיל 1000 משפטים מתוייגים, בפורמט זהה לפורמט של הקובץ הקודם.
3. comp.unlabeled – קובץ המכיל 1000 משפטים לא מתוייגים.

פורמט קבצי האימון (הדוג' היא המשפט השני בקובץ train):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Token Counter** | **Token** | **\_** | **Token POS** | **\_** | **\_** | **Token Head** | **Dependency Label** | **\_** |  | **\_** |
| 1 | Mr. | \_ | NNP | \_ | \_ | 2 | NAME | \_ |  | \_ |
| 2 | Vinken | \_ | NNP | \_ | \_ | 3 | VMOD | \_ |  | \_ |
| 3 | is | \_ | VBZ | \_ | \_ | 0 | ROOT | \_ |  | \_ |
| 4 | chairman | \_ | NN | \_ | \_ | 3 | VMOD | \_ |  | \_ |
| 5 | of | \_ | IN | \_ | \_ | 4 | NMOD | \_ |  | \_ |
| 6 | Elsevier | \_ | NNP | \_ | \_ | 7 | NAME | \_ |  | \_ |
| 7 | N.V. | \_ | NNP | \_ | \_ | 5 | PMOD | \_ |  | \_ |
| 8 | , | \_ | , | \_ | \_ | 7 | P | \_ |  | \_ |
| 9 | the | \_ | DT | \_ | \_ | 12 | NMOD | \_ |  | \_ |
| 10 | Dutch | \_ | NNP | \_ | \_ | 12 | NMOD | \_ |  | \_ |
| 11 | publishing | \_ | VBG | \_ | \_ | 12 | NMOD | \_ |  | \_ |
| 12 | group | \_ | NN | \_ | \_ | 7 | APPO | \_ |  | \_ |
| 13 | . | \_ | . | \_ | \_ | 3 | P | \_ |  | \_ |

* כל שורה מייצגת מילה, וכוללת 10 עמודות, המופרדות ע"י התו ‘\t’.
* העמודות היחידות הרלוונטיות למשימה שלנו הן הצבועות באדום – מיקום המילה במשפט, המילה עצמה, חלק הדיבר המתאים עבורה והראש שלה. נא להתעלם מ-Dependency Label, שכן בתרגיל זה אנו נבצע חיזוי רק לעץ התלויות של המשפט ולא לתוויות של הקשתות.
* בין כל זוג משפטים בקובץ ישנה שורה ריקה בה מופיע התו ‘\n’ בלבד.
* בקובץ התחרות, בעמודהToken Head (וכן בעמודה Dependency Label) יש קו תחתי ('\_')
* **שימו לב: נתון לכם גם חלק הדיבר של כל מילה, תוכלו להשתמש בו במידול שלכם**

**אימון (Train) :**

כאמור את שערוך הפרמטרים תבצעו על הנתונים שבקובץ train.labeled. קשתות גרף התלויות האפשריות במשפט ימושקלו על סמך התכוניות אשר יילמדו ע"י רשת הנוירונים הנשנית שתממשו.

עליכם להסביר בהרחבה על המודל שבניתם והרכיבים השונים בו בדו"ח המצורף.

בנוסף, יש לפרט את הHyperparameters-, את זמן האימון הכולל ולצרף גרף המציג את ערך פונקציית ה-loss וערך ה- UAS(בציר y) על פני ה-epochs (בציר x - מס' המעברים על כל המשפטים שבקובץ train.labeled). ניתן לייצר גרפים נפרדים ל-loss ו-UAS. **הסקה (Inference):**

הסקת עצי התלויות (בהינתן המשקולות שנלמדו) תתבצע ע"י אלגוריתם Chu-Liu-Edmonds הנלמד בתרגול 7. אינכם נדרשים לממש אלגוריתם זה, מצורף לתרגיל קוד (chu\_liu\_edmonds.py) המממש את האלגוריתם, אנא השתמשו בו. עם זאת, אלגוריתם זה ישפיע באופן ישיר על אחוז הדיוק שתקבלו עבור המודל, כך שאנו ממליצים להבינו היטב ע"מ להשתמש בו בצורה מיטבית.

**מבחן (Test):**

יש לבצע הסקה (Inference) על הקובץ test.labeled, ולדווח את תוצאות הדיוק(UAS) ברמת מילה.  
בנוסף, יש לצרף גרף המציג את ערכי ה-loss ו-UAS (ציר y) על פני ה-epochs (ציר x). ניתן לייצר גרפים נפרדים ל-loss ו-UAS.

**תחרות:**

יש לבצע הסקה (Inference) על הקובץ comp.unlabeled (אשר אינו כולל תיוגים), ולכתוב את תוצאות התיוג לתוך קובץ חדש בפורמט labeled (כמו קבצי האימון) (שם הקובץ הרצוי מופיע בהמשך). לדוג', עבור המשפט:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Token Counter** | **Token** | **\_** | **Token POS** | **\_** | **\_** | **Token Head** | **Dependency Label** | **\_** |  | **\_** |
| 1 | The | \_ | DT | \_ | \_ | \_ | \_ | \_ |  | \_ |
| 2 | Boy | \_ | NNP | \_ | \_ | \_ | \_ | \_ |  | \_ |

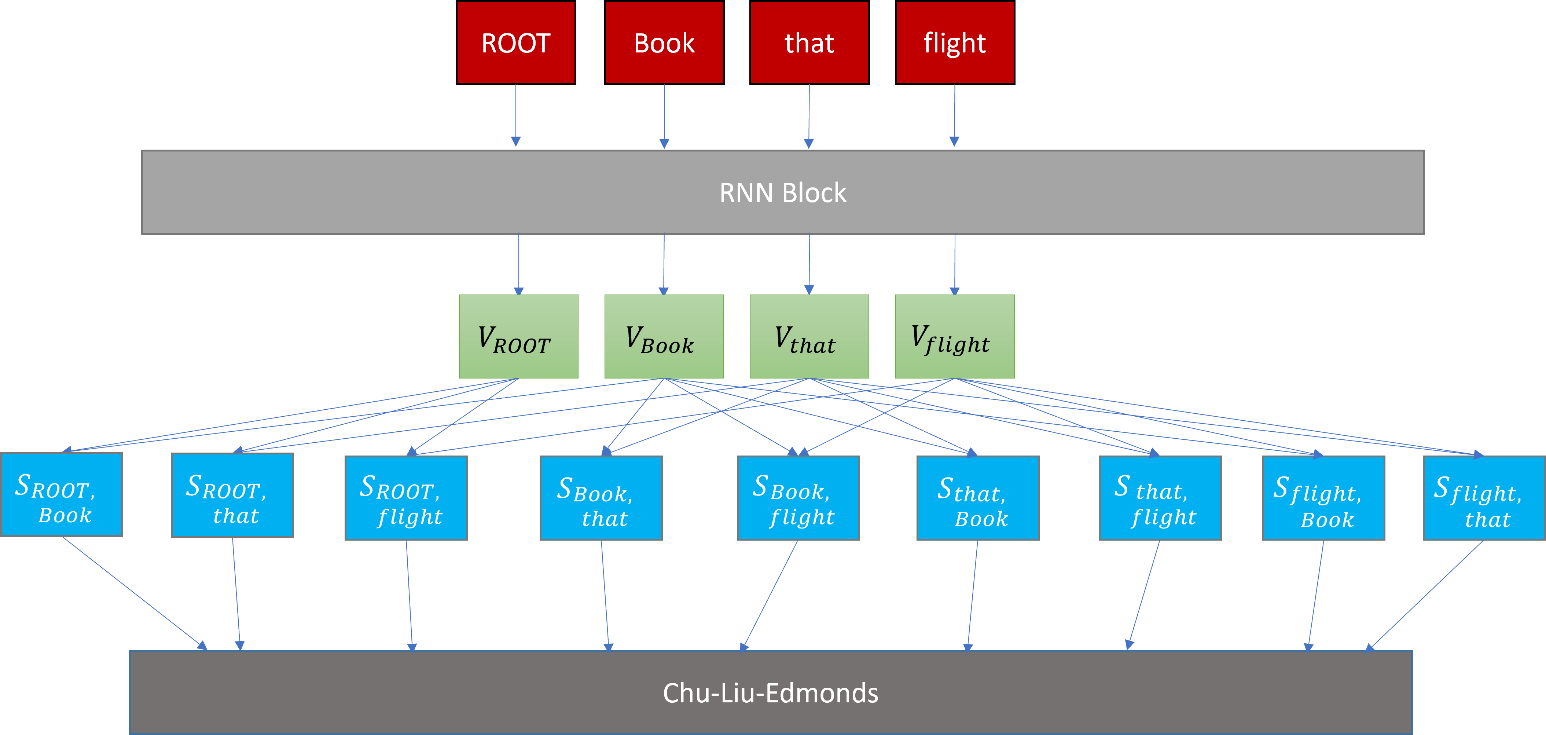
יש לבצע הסקה, שתיתן לכם את התלויות. בהנחה שהתלויות שמצאתם הן , תכתבו אותן לקובץ ההגשה באופן הבא -

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Token Counter** | **Token** | **\_** | **Token POS** | **\_** | **\_** | **Token Head** | **Dependency Label** | **\_** |  | **\_** |
| 1 | The | \_ | DT | \_ | \_ | 0 | \_ | \_ |  | \_ |
| 2 | Boy | \_ | NNP | \_ | \_ | 1 | \_ | \_ |  | \_ |

שימו לב שסדר המשפטים (הלא מתוייגים) בקובץ המקורי זהה לסדר המשפטים בקובץ הפלט, שמספר העמודות זהה ושאף אחד מן הערכים חוץ מהעמודה ששיניתם לא נפגע.

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

כדי לפתור את הבעיה תצטרכו לממש אלגוריתם המכיל את הרכיבים הבאים:



**קוד חיצוני המותר לשימוש:**

הסביבה py38\_default במכונה מכילה את הספריות הרלוונטיות לתרגיל זה. יש לוודא שהקוד רץ על הסביבה הזו.

עבור ההסקה מומלץ להשתמש בקוד המצורף לתרגיל (chu\_liu\_edmonds.py) המממש את אלגוריתם Chu-Liu-Edmonds.

**הגשה:**

קובץ zip בלבד, בשם HW3\_123456789\_987654321.zip (עבור שני סטודנטים שמספרי הזהות שלהם 123456789 ו 987654321). הקובץ הנ"ל יכלול:

1. **דו"ח קצר** (עד 3 עמודים בפורמט PDF) המכיל הסברים תמציתיים, דיווח וניתוח תוצאות, הכולל:
   1. שמות המחברים ות"ז
   2. אימון - דיווח אחוז דיוק (UAS) על קובץ האימון, גרפים והערות על תהליך אימון המודל (לפי הדגשים בסעיף "אימון")
   3. הסקה - הערות על שימוש באלגוריתם ההסקה (לפי הדגשים בסעיף "הסקה")
   4. מבחן - דיווח אחוז דיוק (UAS) על קובץ המבחן, גרפים והערות עבור המודל (לפי הדגשים בסעיף "מבחן").
   5. תחרות **-** הסבר קצר על שיפורים שעשיתם למודל עבור תיוג קבצי התחרות (לפי הדגשים בסעיף "תחרות")
   6. 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 מכונת Azure עם סביבת העבודה המתאימה. אנא כתבו ממשקי הרצה פשוטים לאימון, מבחן וייצור קובץ התחרות המתויג.
3. **קובץ התחרות** **מתויג** – על קובץ התוצאות להיות בפורמט labeled (כפי שמפורט בחלק "אימון"). על מנת להימנע מאי נעימויות, אנא ודאו כי אם שמים '\_' בעמודות ששיניתם מקובץ התחרות מקבלים בדיוק את הקובץ comp.unlabeled (אותן שורות לפי אותו סדר). חוסר התאמה פירושו ציון 0 בחלק הזה.

על שם הקובץ להיות comp\_987654321\_123456789.labeled

1. **ממשק לתיוג קבצי התחרות -** על קבצי התחרות להיות ניתנים לשחזור (Reproducible).   
   הדרישה היא שניתן יהיה לקחת את הקוד והמודל המאומן שהגשתם ולייצר באמצעותם קובץ תחרות מתויג זהה לחלוטין לקבצים שהגשתם.   
   לטובת שחזור הקבצים, יש לכתוב ממשק הרצה פשוט, בקובץ נפרד בעל השם –generate\_comp\_tagged.py  
   להרצת Inference בלבד על המודל המאומן ויצירת קובץ התחרות המתויג ע"י המודל.
2. חלק יבש - קובץ מוקלד עם תשובות לחלק היבש (HW3-dry) **בשם** dry\_123456789\_987654321.pdf

**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 להעתיק קטעי קוד מוכנים מהאינטרנט, ובכלל אין להסתמך על שום מקור אחר לקוד מלבד פרי יצירכם והחבילות החיצוניות אשר צוינו בסעיף הרלוונטי.

**נספח** - דוגמא לחישוב אחוז דיוק UAS:

