עיבוד שפה טבעית – תרגיל בית 2 - רטוב

תיאור המשימה

בתרגיל בית זה תממשו Dependency Parser (כפי שנלמד בשבוע 8) מבוסס רשתות נוירונים ללמידת דקדוק תלויות, ותנתחו את טיב הצלחתכם.

אתם מתבקשים לממש מודל המבוסס על ה-Graph-based Parser של Kiperwasser and Goldberg אשר מאמרם הוצג בכיתה. אנו ממליצים לקרוא לפחות את החלקים הבאים מתוך המאמר טרם ובמהלך מימוש המודל ע"מ לוודא הבנתכם: 2.3 (Bidirectional Recurrent Neural Networks), 3 (Our Approach), 5 (Graph-based Parser), Implementation Details ו-Hyperparameter Tuning תחת חלק 6 (Experiments and Results). כמו כן, אנא חזרו על מצגת התרגול של שבוע 8 ובפרט על אלגוריתם Chu-Liu-Edmonds.
המודל שתממשו אינו זהה ב-100% לזה שמוצג במאמר אך דומה לו מאוד, ההבדלים יפורטו בהמשך ההוראות.

כפי שתראו בהמשך, בתרגיל הבית הנוכחי הופחתו מספר הרכיבים האלגוריתמים אותם אתם צריכים לממש. זאת על מנת שתתעסקו יותר בבניית הייצוג, בארכיטקטורת הרשת, מאפייניה וכו'.

סגל הקורס ממליץ להשתמש בשפת Python 3, ובספריית PyTorch למימוש ואימון רשתות נוירונים.
לצורך אימון הרשתות בזמן סביר, לרשותכם עומדות מכונות GPU בסביבת ה-Azure. מצורפים לתרגיל הוראות התחברות למכונות אלו.

הסבר על מבנה הציון בתרגיל:

- **60%** - מימוש מלא של המודלים הבסיסי והמתקדם כפי שיפורטו בהמשך, אימונם על קובץ ה-`train.labeled`, ועמידה בסף אחוז הדיוק (UAS) של 70% על קובץ ה-`test.labeled` (עבור המודל הטוב מבין השניים).
- **30%** - תחרות מבוססת אחוז דיוק (UAS) בתיוג קובץ התחרות `comp.unlabeled`. הציון יתבסס על תחרות ביצועים כללית על קובץ התחרות ועל ביצועים בתוך קבוצות משפטים לפי דרגות קושי שונות.
הציון על התחרות יתבסס על $\max\{accuracy(comp\_m1), accuracy(comp\_m2)\}$, כלומר נסתכל על התוצאה הטובה יותר מבין שני המודלים שתגישו (הבסיסי והמתקדם).
- **10%** - כתיבת דו"ח תמציתי (עד 3 עמודים) אשר יכלול את הסעיפים הנדרשים ועמידה בתנאי פורמט ההגשה (יפורטו בהמשך המסמך)

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, בעמודה Token Head (וכן בעמודה Dependency Label) יש קו תחתי ('_')

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.

אתם נדרשים לבנות שני מודלים:

1. מודל ***בסיסי***, מבוסס על רשת הנוירונים המוצגת במאמר של Kiperwasser and Goldberg בחלק 5 כפי שהיא ממומשת במאמר, כאשר תדרשו לממש פונקציית loss שונה מזו המתוארת במאמר.
במודל שלנו נמזער פונקציית Negative log-Likelihood Loss (NLLLoss) המוגדרת באופן הבא:

$$\min_{\theta} NLLLoss(D;\theta) = \min_{\theta} \sum_{(X^i,Y^i)\in D} \sum_{(h,m)\in Y^i} -\frac{1}{|Y^i|} \cdot log\left(P\left(S^i_{h,m}|X^i,\theta\right)\right)$$

$$P\left(S^i_{h,m}|X^i,\theta\right) = \frac{exp\left(S^i_{h,m}\right)}{\sum_{j=1}^{|Y^i|} exp\left(S^i_{j,m}\right)} = Softmax\left(S^i_{h,m}\right)$$

Where:

- $D = \{(X^i, Y^i)\}_{i=1}^{n}$ is a dataset consisting of n (sentence, true tree) pairs.
- $X^i = \{x_0 = ROOT, x_1, \dots, x_{k_i}\}$ is the full sequence of words in the sentence.

- $Y^i = \{(h, m)\}$ is the set of all (head_index, modifier_index) edges in the **true** dependency tree of sentence X^i. $|Y^i| = k_i$.
- $S^i \in R^{(k_i+1)^2}$ is the score matrix for all possible (head, modifier) edges in the dependency graph of sentence X^i.
  The cell $S^i_{h,m}$ refers to the score of h being the head of m in sentence X^i.
- θ are all the network's learned parameters.

2. מודל ***מתקדם***, בו תוכלו לבצע כל שינוי שתרצו למודל *הבסיסי*: בארכיטקטורת הרשת, פונקציות אקטיבציה, מימדי השכבות, Hyperparameters, פונקציית loss, ואף שימוש ב-word embeddings מאומנים מראש (כגון Word2Vec או GloVe).

במהלך האימון, תשתמשו באלגוריתם אופטימיזציה (Optimizer) מבוסס גרדיאנט (SGD, Adam וכו') של PyTorch לשערוך המשקולות האופטימליות לכל הקשתות האפשריות בגרף התלויות למשפט נתון.

<u>כל שיפור שהכנסתם למודל המתקדם צריך להיות מוסבר היטב</u>, כולל המוטיבציה לבצע אותו.

בנוסף, יש לפרט את ה-Hyperparameters, 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`) לכל מודל. ניתן לייצר גרפים נפרדים ל-loss ו-UAS.

<u>הסקה (Inference):</u>

הסקת עצי התלויות (בהינתן המשקולות שנלמדו) תתבצע ע"י אלגוריתם Chu-Liu-Edmonds הנלמד בתרגול 8. אינכם נדרשים לממש אלגוריתם זה, מצורף לתרגיל קוד (`chu_liu_edmonds.py`) הממש את האלגוריתם, אנא השתמשו בו. עם זאת, אלגוריתם זה ישפיע באופן ישיר על אחוז הדיוק שתקבלו עבור המודלים, כך שאנו ממליצים להבינו היטב ע"מ להשתמש בו בצורה מיטבית.

<u>מבחן (Test):</u>

לכל אחד מן המודלים, יש לבצע הסקה (Inference) על הקובץ `test.labeled`, ולדווח את תוצאות הדיוק (UAS) ברמת מילה, כפי שנעשה בהרצאת הוידאו (שקף דוגמא מצורף בנספח בסוף הוראות התרגיל). 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.

התייחסו להבדל בביצועים בין המודלים השונים, והעלו מספר סיבות שעשויות לגרום להבדלים אלו.

בנוסף, אנא ציינו כמה זמן לקח לתייג את הקובץ לפי כל אחד מהמודלים.

תחרות:

לכל אחד מן המודלים, יש לבצע הסקה (Inference) על הקובץ `comp.unlabeled` (אשר אינו כולל תיוגים), ולכתוב את תוצאות התיוג לתוך קובץ חדש בפורמט labeled (כמו קבצי האימון) (שמות הקבצים הרצויים מופיעים 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 $1 \rightarrow 2$, $0 \rightarrow 1$, 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').

קוד חיצוני המותר לשימוש:

החבילות הסטנדרטיות ב-Python 3, והחבילות הבאות:
pytorch, torchtext, ignite, numpy, scipy, matplotlib, seaborn, pandas, tabulate, jupyter, jupyterlab

מצורף לתרגיל קובץ (`nlp_hw2_env.yml`) המייצר סביבת Anaconda מתאימה על מכונת ה-Azure, המכילה את כל החבילות המותרות לשימוש. זו גם הסביבה בה ייבדקו התרגילים, אנו ממליצים לעבוד איתה.

מעבר לחבילות המפורטות בקובץ זה, **אין להשתמש** באף חבילה או ספריה חיצונית אחרת, ובפרט שום ספריה או חבילה שעושה עיבוד על טקסט או מממשת Dependency Parser.

עבור ההסקה מומלץ להשתמש בקוד המצורף לתרגיל (`chu_liu_edmonds.py`) המממש את אלגוריתם Chu-Liu-Edmonds.

במודל המתקדם, אם תרצו להשתמש ב-word embeddings מאומנים מראש, תוכלו להיעזר בחבילת torchtext.

הגשה:

קובץ zip בלבד, בשם `HW2-Wet_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:
   a. שמות המחברים ות"ז
   b. אימון - דיווח אחוז דיוק (UAS) על קובץ האימון, גרפים והערות על תהליך אימון המודלים (לפי הדגשים בסעיף "אימון")
   c. הסקה - הערות על שימוש באלגוריתם ההסקה (לפי הדגשים בסעיף "הסקה")
   d. מבחן - דיווח אחוז דיוק (UAS) על קובץ המבחן, גרפים והערות עבור כל מודל (לפי הדגשים בסעיף "מבחן").
   e. תחרות - הסבר קצר על שיפורים שעשיתם למודלים עבור תיוג קבצי התחרות (לפי הדגשים 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 מכונת Azure עם סביבת העבודה המתאימה. אנא כתבו ממשקי הרצה פשוטים לאימון, מבחן וייצור קבצי התחרות המתויגים.
3. **קבצי התחרות מתויגים** – על קבצי 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 שמות הקבצים להיות – (123456789 הוא ת"ז של אחד הסטודנטים)
   a. `comp_m1_123456789.labeled` – קובץ labeled שאומן על ידי המודל הבסיסי.
   b. `comp_m2_123456789.labeled` – קובץ labeled שאומן על ידי המודל המתקדם.
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.

בסה"כ מבנה קובץ ההגשה צריך להיראות כך (דומה לתרגיל 1):

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

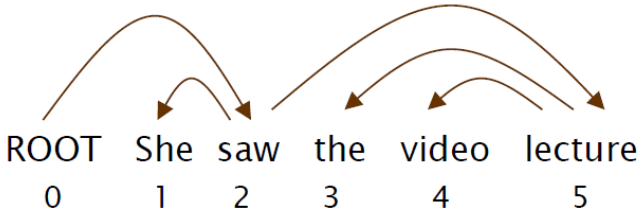
העתקות:

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 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 (מתוך הרצאת הוידאו):

Christopher Manning

Evaluation of Dependency Parsing: (labeled) dependency accuracy



$$\text{Acc} = \frac{\#\text{ correct deps}}{\#\text{ of deps}}$$

UAS = 4 / 5 = 80%

LAS = 2 / 5 = 40%

Gold

1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed

1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp