ריטוב 2 – NLP

אמיר בלדר – 204179659 | רועי גנץ – 204506349

המודל הבסיסי:

מימשנו את מודל ה-graph-based אשר מתואר במאמר של Kiperwasser ו-Goldberg. לצורך מימוש משאבי האימון (Datasets & Dataloaders) וה-Embeddings, נעזרנו בקוד אשר סופק לנו בסדנא הייעודית. את חלק ה-MLP מימשנו לפי הטריק אשר צוין במאמר בתת הפרק Speed improvements - פיצלנו את השכבה הראשונה בMLP ל-2 כך שבמקום שתקבל כקלט זוגות, מימשנו שתי שכבות לינאריות, אחת ל-heads ואחת ל-modifiers. יתר על כן, בשונה מהמאמר, מימשנו פונקציית NLLLoss, כפי שנתבקשנו בהנחיות התרגיל. לצורך ביצוע Inference השתמשנו באלגוריתם Chiu-Liu Edmonds אשר סופק לנו.

אימון:

בהתאם לתת הפרק "Implementation Details" במאמר, מימשנו את ה-word dropout המוצע. בחרנו ב-Adam בתור optimizer ואימנו באמצעות mini-batches "virtual", בהתאם לטריק אשר הוצג בסדנא, כאשר בחרנו בגודל batch של 128. בנוסף, השתמשנו ב-gradient clipping לתחום ערכים של [-1,1], כדי להביא לאימון יציב יותר. בתור היפר-פרמטרים למודל שלנו, בחרנו את הפרמטרים המוצעים במאמר:

Word embedding dim	100
POS embedding dim	25
MLP hidden dim	100
#Bi-LSTM layers	2
LSTM hidden dim	125
α for word dropout	0.25

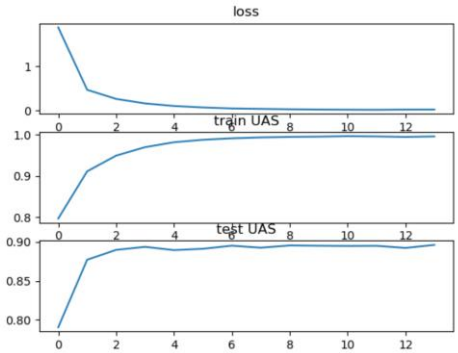
כדי לנסות ולמנוע מצב של overfitting, מימשנו מנגנון של Early Stopping. מנגנון זה אמור להביא להקטנת הGeneralization Gap – כאשר המודל לא השתפר במשך 10 אפוקים על קבוצת המבחן, עצרנו את האימון ושמרנו את המשקולות של המודל אשר השיג את התוצאה הטובה ביותר. נציין כי אימון על epoch יחיד ארך כ-3 דק'.

הסקה:

לצורך ביצוע ההסקה, נעזרנו במימוש המסופק לאלגוריתם Chiu-Liu Edmnods.

מבחן:

מצ״ב גרף המתאר את ה-loss וה-UAS:



כפי שעולה מן הגרף, המודל ממזער את ה-loss לאורך האימון על קבוצת האימון, ומגיע להצלחה כמעט מלאה עליו. כמו כן, המודל הצליח להגיע כמעט ל-90% דיוק על קבוצת המבחן.

ה-UAS של המודל הבסיסי על קבוצת המבחן הינו 89.78%.

המודל המתקדם:

לצורך בניית המודל המתקדם, בחנו מספר רעיונות. להלן פירוט הרעיונות והמוטיבציה העומדת מאחוריהם:

- שימוש ב-ReLU בתור האקטיבציה הלא לינארית ב-MLP במקום Tanh. אקטיבציה זו פופולארית יותר בימינו שכן אמפירית, היא מביאה לתוצאות טובות יותר מאשר Tanh במרבית המשימות.
- העמקת הרשת ושיפור ההיפר-פרמטרים – כפי שלמדנו בקורס, במודלים מבוססי רשתות נוירונים, בתחום ה-Deep Learning בכלל, וב-NLP בפרט, העמקת המודל ושיפור ההיפר-פרמטרים שלו, עשויים לשפר מאוד את ביצועי המודל. הסיבה נובעת מכך שהדבר מאפשר משפחת היפותזות עשירה יותר ומודלים אקספרסיביים יותר. אי לכך, בחנו את השפעת שינוי ההיפר-פרמטרים של המודל שלנו, ובפרט את מספר שכבות ה-Bi-LSTM על ביצועי המודל (הגדלה ל-4 שכבות). למעשה, ביצענו hyper-parameter tuning על גדלי ה-Embeddings, מספר שכבות ה-Bi-LSTM, מימד ה-MLP וה-Bi-LSTM. נציין כי לצורך כיוונון ההיפר-פרמטרים, חילקנו את קבוצת האימון המקורית לקבוצת אימון וקבוצת הערכה, ובחנו את השפעת ההיפר-פרמטרים השונים על קבוצת ההערכה, ולא על המבחן, כדי להימנע ממצב של overfitting על קבוצת המבחן.
- שימוש ב-Pretrained word embeddings: בחנו את ההשפעה של שימוש ב-GloVe ואימונו לעומת אימון מלא של embeddings. היות ו-GloVe אומן על קורפוס עצום המכיל המון מילים, סביר כי הוא יקודד מידע שימושי אודות כל המילים בקורפוס שלנו, וכנ״ל למילים שיופיעו בקבוצת המבחן אבל לא בקבוצת האימון. נציין כי השתמשנו ב-Embeddings של ה-GloVe בתור אתחול ל-Embeddings שלנו, והמשכנו לאמן אותם כך שיתאימו למשימה הייחודית שלנו.

- גורמי רגולריזציה – מגרפי האימון שלל המודל הבסיסי עולה כי ישנו Generalization Gap גדול יחסית בין ה-Train לבין ה-Test. כדי להקטין את פער ההכללה, בחנו את השימוש ב-Dropout, הן ב-LSTM והן ב-MLP.

כדי להכריע אילו שדרוגים ושכלולים להוסיף, הרצנו ניסויים רבים שבהם בחנו את התוצאות על פני קבוצת ההערכה שיצרנו מתוך קבוצת האימון. בסופו של דבר, בחרנו לבצע את השינויים הבאים:

- ❖ שימוש ב-ReLU בתור שכבת האקטיבציה הלא-לינארית ב-MLP/
- ❖ שימוש ב-GloVe בתור pre-trained embeddings אך לא הקפאנו אותם, אלא אימנו אותם.
- ❖ שימוש ב-4 שכבות Bi-LSTM.
- ❖ שימוש בהיפר-פרמטרים הבאים:

Word embedding dim	300 (Glove)
POS embedding dim	25
MLP hidden dim	200
#Bi-LSTM layers	4
LSTM hidden dim	200
α for word dropout	0.25

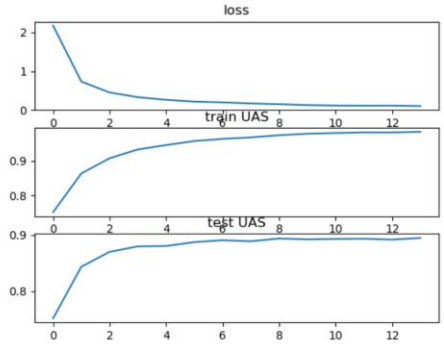
- ❖ שימוש ב-Dropout, הן ב-Bi-LSTM והן ב-MLP.

<u>אימון והסקה:</u>

באופן דומה למודל הבסיסי.

<u>מבחן:</u>

מצ"ב גרף המתאר את ה-loss וה-UAS:



ה-UAS של המודל הבסיסי על קבוצת המבחן הינו 89.5%.

<u>תחרות:</u> לא ביצענו שינויים נוספים במודלים עבור יצירת קבצי התחרות המתויגים.