

אימון:

מספר המאפיינים הבסיסיים בהם השתמשנו:

1. p-word, p-pos: 15,909
2. p-word: 14,165
3. p-pos: 49
4. c-word, c-pos: 15,909
5. c-word: 14,165
6. c-pos: 49
7. p-pos, c-word, c-pos: 779,541
8. p-word, p-pos, c-pos: 779,541
9. p-pos, c-pos: 2,401

סך הכל 1,621,729 מאפיינים לכל זוג מילים.

המודל המורכב משתמש באותם מאפיינים של המודל הבסיסי בתוספת המאפיינים הבאים:

1. p-pos, p-pos+1, c-pos-1, c-pos: 5,764,801
2. p-pos-1, p-pos, c-pos-1, c-pos: 5,764,801
3. p-pos, p-pos+1, c-pos, c-pos+1: 5,764,801
4. p-pos-1, p-pos, c-pos, c-pos+1: 5,764,801

ארבעת המאפיינים הקודמים נותנים מידע על הסביבה של זוג התייגים של המילים p, c. אלו מאפיינים בינאריים כמו למשל:

if p-pos=DT and p-pos+1=NN and c-pos-1=NN and c-pos=Vt: 1
else: 0

5. distance: 60

זהו מאפיין בינארי אשר נותן מידע על המרחק בין זוג במילים p, c כמו למשל:

if distance(p, c) = 32: 1
else: 0

6. direction: 2

זהו מאפיין בינארי שנותן מידע על כיוון הקשת:

if position(p) > position(c): 1
else: 0

7. in-between pos: 49

זהו מאפיין בינארי שנותן מידע על אלו תיגים קיבלו המילים בין זוג המילים p, c לדוגמה:

if DT is pos of a word between p and c: 1
else: 0

סך הכל 24,681,044 מאפיינים לכל זוג מילים.

ניסינו להכניס למודל שיפור להאצת זמן האימון של גיזום קשתות שהן בסבירות מאוד גבוהה לא נכונות על פי המאמר הבא [Fast and Accurate Arc Filtering for Dependency Parsing](#) שהתפרסם בכנס ACL בשנת 2010 אך מכיוון שקיבלנו ירידה חדה בדיוק של המודל החלטנו לוותר על סוג זה של גיזום.

זמני אימון של המודלים:

מודל	20	50	80	100
בסיס	2191 שני'	5314 שני'	8473 שני'	10,261 שני'
מורכב	2945 שני'	6994 שני'	10,824 שני'	13,443 שני'

החומרה עליה הרצנו את הניסויים היא: 2x Intel(R) Xeon(R) CPU E5-2699A v4, סך הכל 88 ליבות לוגיות, 256 ג'יגה RAM, מערכת הפעלה Ubuntu 16.04. השתמשנו במחשב זה רק כדי שנוכל להריץ מספר מודלים במקביל.

חשוב לציין שהמודלים שלנו גם מסוגלים לרוץ על מחשב רגיל ולשם השוואה הרצנו את המודל המורכב למשך 20 איטרציות של אימון:

- זמן אימון: 2675 שניות
- חומרה: Intel(R) Core(TM) i7-4700MQ CPU, סך הכל 8 ליבות לוגיות, 8 ג'יגה RAM, מערכת הפעלה Ubuntu 18.04.

מבחן:

תוצאות המבחן של המודלים:

מודל	20	50	80	100
בסיס	44.8%	29.9%	31.2%	30.1%
מורכב	80.9%	77.6%	79.2%	80.5%

זמני התיג של קובץ המבחן:

מודל	זמן תיג
בסיס	26 שני'
מורכב	56 שני'

אנו שמים לב להבדלים משמעותיים בביצועים של שני המודלים. אנו חושבים שההבדלים יכולים לנבוע משתי סיבות עיקריות:

1. המודל הבסיסי בעצם מניח שהניקוד של כל זוג מילים הוא בלתי תלוי בשאר זוגות המילים ולמעשה מוציא כל זוג מילים לגמרי מהקונטקסט של המשפט. ראינו בקורס מספר דוגמאות לכך שאי אפשר לעשות הנחה כזו והמשמעות של כל מילה נקבעת במידה רבה על פי ההקשר שלה. במודל המורכב אנו מבטלים את ההנחה הזו בכך שאנו מציגים מאפיינים שממדלים את ההקשר, לדוגמה מאפיינים שנותנים מידע על התיגים בסביבה של זוג המילים, מאפיינים של מרחק בין שתי מילים וכו'.
2. שיטת ההסקה שנבחרה Chu-Liu-Edmonds, שיטה זו בעצם מוצאת את העץ הפורש המקסימלי של גרף מלא מכוון, כאשר כל מילה במשפט היא צומת בגרף. שיטה זו אינה מונעת מצבים של קשתות שחוצות אחת את השניה. כמו שלמדנו אנגלית היא שפה בה עצי התלות הם projective כלומר קשתות אינן חוצות אחת את השניה. כאשר ניתחנו את העצים שהוסקו על ידי המודל הבסיסי ראינו שרוב השגיאות הם קשתות שחוצות אחת את השניה. אנו מניחים שהמידע הנוסף שקיים במודל המורכב גרם לו ליצור עצים ללא קשתות חוצות, כמו למשל המאפיין שנותן מידע על כל החלקי

דיבר שנמצאים בין שתי מילים, מאפיין זה יכול לומר אם יש פועל בין זוג שמות עצם ולכן לא סביר שתהיה קשת ביניהם כיוון שקשת כזו כנראה תחצה קשת היוצאת מן הפועל.

תחרות:

אנו מתייגים את קבצי התחרות בעזרת המערכות שהביאו לנו את התוצאות הטובות ביותר על סט המבחן. אנו מניחים כי מודל שהביא לנו תוצאות טובות על סט המבחן בעצם למד הכללה של המקרה הכללי של Dependency Parsing. אנו מצפים לראות תוצאות דיוק נמוכות יותר על קבצי התחרות מכיוון שבזמן אימון המודלים עשינו התאמה גם לסט המבחן ולכן המודל למד בעקיפין גם את סט המבחן, לעומת זאת המודל לא ראה אף פעם את הדאטה בקבצי התחרות ולכן אנו מצפים לראות ירידה בביצועים. למרות זאת אנו מניחים כי הדאטה בקבצי התחרות נלקח מאותה התפלגות של קבצי האימון והמבחן ולכן אנו מצפים שהירידה בביצועים אינה תהיה דרמטית.

צילום מסך של תהליך האימון של המודל המורכב שהגשנו עבור התחרות:

```
(py36-gpu) ofir_nlp@:~/DependencyParser-NLP$  
extract train features  
extract ended 158.04655861854553  
learn model weights  
iteration 1 / 20  
iteration 2 / 20  
iteration 3 / 20  
iteration 4 / 20  
iteration 5 / 20  
iteration 6 / 20  
iteration 7 / 20  
iteration 8 / 20  
iteration 9 / 20  
iteration 10 / 20  
iteration 11 / 20  
iteration 12 / 20  
iteration 13 / 20  
iteration 14 / 20  
iteration 15 / 20  
iteration 16 / 20  
iteration 17 / 20  
iteration 18 / 20  
iteration 19 / 20  
iteration 20 / 20  
learning ended: 2786.867024421692  
train evaluation  
train accuracy: 0.962202109109026  
evaluation ended: 130.94348907470703  
extract test features  
extract ended 30.026404857635498  
test evaluation  
test accuracy: 0.88896197327852  
evaluation ended: 20.327112426757812
```

חלוקת עבודה:

את תכנון המערכת ביצענו יחד, כלומר איך לעבור על הדאטה, איך להוציא מאפיינים עבור מילים, איזה מאפיינים להוציא, כיצד לבצע את האימון, כיצד לבצע את ההסקה נעשה בחשיבה משותפת. לאחר מכן פתחנו git repository ועבדנו על הקוד במקביל. מכיוון וכל אחד מאיתנו נגע בכל מודול של הפרויקט והיה שותף לתכנון, לכתיבת קוד ודיבוג קשה לנו לעשות הפרדה מוחלטת בין העבודה של שנינו.