תרגיל בית 1 - רטוב - דו"ח

מגישים- אליעזר שלומי 315227686, יקטרינה סופרפין 321170383

הסבר על סט המאפיינים:

השתמשנו בפיצ'רים $f_{100} - f_{105}$ של רטנפרקי והפיצ'רים הנוספים של המספרים והמילים המכילות אותיות גדולות- וזאת כנדרש בתנאי התרגיל. בנוסף הוספנו גם את הפיצ'רים $f_{106} - f_{107}$. נפרט על כל פיצ'ר:

פיצ'רים שנדרשנו לממש:	כמות באימון מודל1	כמות באימון מודל2
f_{100}- פיצ'ר של התאמה של תיוג למילה.	15415	1805
f_{102}- פיצ'ר של התאמת תיוג למילה ותחילית מסויימת של המילה.	13265	2556
f_{101}- פיצ'ר של התאמת תיוג למילה וסיומת מסויימת של המילה.	22393	3978
f_{103}- פיצ'ר טריגרם.	8150	1168
f_{104}- פיצ'ר בייגרם.	1060	316
f_{105}- פיצ'ר יוניגרם.	44	32
feature_words_with_numbers – פיצ'ר של הופעת מספר	1248	180
feature_words_with_capital_letters- פיצ'ר של מילים אותיות גדולות	397	420
פיצ'רים שהוספנו:		
f_{106}- פיצ'ר של התאמת תיוג לפי מילה קודמת.	38164	3314
f_{107}- פיצ'ר של התאמת תיוג לפי מילה הבאה	35613	3141
feature words with capital hyphens	1018	219
Feature words with capitals only	22	6
Feature words with capital letters only	36386	420
$\langle \text{next next word, tag} \rangle$ feature	36386	2928
$\langle pre\ pre\ word, tag \rangle\ feature$	38050	3065
$\langle pre\ pre\ word, pre\ word, tag \rangle$	82798	5007
$\langle next\ next\ word, next\ word, tag \rangle$	82798	5007
$\langle pre\ pre\ word, pre\ tag, tag \rangle\ feature$	33511	2826
$feature\ 100\ lower$ – כמו 100 רק באותיות קטנות	14719	1736
$feature\ 101\ lower$ – כמו 101 רק באותיות קטנות	12608	2467
$feature\ 101\ lower\ complete$ – כמו 101 רק אותיות קטנות וניקח את המשלים (הצד השני של המילה)	42044	5574
$feature\ 102\ lower$ – כמו 102 רק אותיות קטנות	20156	3593
$feature\ 102\ lower\ complete$ – כמו 102 רק אותיות וניקח את המשלים	35049	5226
Feature upper tag exist - $\langle \text{tag} \rangle$ אם יש $capital\ letter$.	35	16
Feature hyphen exist - $\langle \text{tag} \rangle$ אם יש מקף במילה.	15	7

Feature digit exist - ⟨tag⟩ אם יש ספרה במילה.	6	4
מבנה מילה עבור המילה הנוכחית XXxxxdd – X אות גדולה, x אות קטנה ו – d זאת ספרה	1025	466
מבנה מילה עבור המילה הקודמת XXxxxdd – X אות גדולה, x אות קטנה ו – d זאת ספרה	1025	772
מבנה מילה עבור המילה הבאה XXxxxdd – X אות גדולה, x אות קטנה ו – d זאת ספרה	2472	466

<u>שלב האימון:</u>

מפרט בסיסי של התוכנה עליה הרצנו מכונת Azure – (4 vcpus, 16 GiB memory) Standard D4s v3

בשביל לאמן השתמשנו בפונקציה של scipy הנקראת fmin_l_bfgs_b שהשתמשה בפונקציה שכתבנו שמחשבת גרדיאנט ולוס.

<u>מודל 1</u> – יצירת הפיצ'רים לקח 3 דקות, האימון עוד 5-7 דקות, חישוב הדיוק על הטסט 3 דקות והתיוג לקח גם 2-3 דקות.

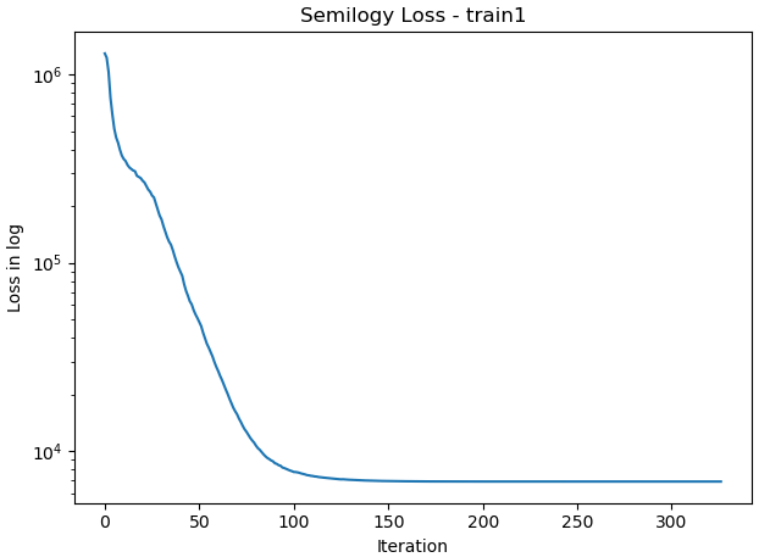
```
NN      10   26    2    19     0    8
Accuracy of train 1 test is  0.96321

Process finished with exit code 0
```

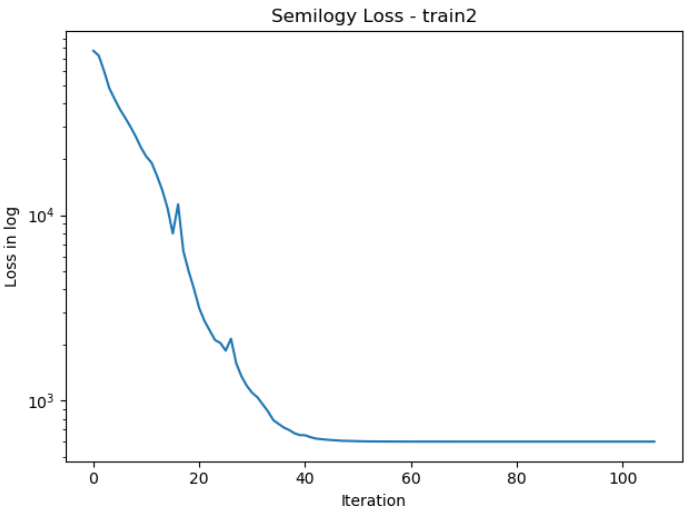
- מדד ה-*Accuracy*- 0.96321
- זמן האימון- 315 איטרציות כאשר איטרציה לוקחת בערך שנייה.
- מקדם רגולריזציה- 1.
- Threshold- 1 – חשבנו שאולי יהיה לנו אוברפיט אך סט הולידציה מראה ששימוש ב threshold=1 מביא לתוצאות טובות.
- באימון מודל זה השתמשנו בכל הפיצ'רים המצוינים בטבלה. ניתן לראות בטבלה שלעיל את מספר הפיצ'רים מכל סוג ששימשו לאימון מודל1 ו 2.
- בשביל מימוש יעיל השתמשנו בפעולות וקטוריות ומטריצה sparseית של scipy. משתנים שיכולנו לחשב פעם אחת לפני האימון חושבו פעם אחת והוכנסו כפרמטר ($emprical\ count$).

בהניתן ה Confusion Matrix ניתן לראות שיש בלבול רב בין ה JJ ו - NN. הייתי מנסה לתת למודל כוח להבדיל ביניהם ע"י זה שהייתי מוסיף פיצ'ר של תחילת המשפט. לדעתי פעמים רבות אנו מזכירים שם של מישהו בתחילת המשפט ובהמשך מוסיפים עליו תארים, אולי בדרך זו ניתן יהיה להבין מתי אנחנו מתכוונים לשם עצם ומתי לשם תואר.
עוד רעיון שחשבתי עליו (שהוא מתקשר בעיקר למילים לא מוכרות ויש לו קשר גם לטבלה) יהיה לנסות לעשות stemming או משהו דומה כדי שאם המודל יראה מילה לא מוכרת הוא ימצא את הגרעין שלה (למקרה שהמודל לא הכיר את המילה רק בגלל שהיא וראיציה של מילה אחרת) והוא יוכל לנחש מה התפקיד שלה בהינתן סיומת/גרעין וכו'.

Confusion Matrix

	VB	VBG	VBD	NNP	NNPS	RB	VBN	IN	JJ	NN
VB	554	0	1	1	0	1	0	1	2	15
VBG	0	368	0	2	0	0	0	0	1	29
VBD	1	0	795	0	0	0	34	0	0	2
NNP	0	0	0	1940	18	0	0	0	10	12
NNPS	0	0	0	29	32	0	0	0	4	0
RB	1	0	0	0	0	732	0	17	16	6
VBN	0	0	31	0	0	0	434	0	30	3
IN	0	1	0	2	0	48	0	2492	1	1
JJ	3	6	6	8	0	11	27	2	1367	65
NN	10	26	2	19	0	8	1	3	79	3173





<u>מודל 2</u> – יצירת הפיצ'רים לקח דקה, 3-2 דקות לאימון ועוד 2-1 דקות לתיוג התחרותי.

- זמן האימון- 97 איטרציות
- מקדם רגולריזציה- 1.0
- Threshold=1
- באימון מודל זה השתמשנו באותם פיצ'רים כמו במודל 1
- ניתן לראות בטבלה שלעיל את מספר הפיצ'רים מכל משפחת פיצ'רים ששימשו לאימון מודל 2.

שלב ההסקה:

את שלב ההסקה ביצענו ע"י אלגוריתם ה*Viterbi*, תוך שימוש ב-*beam search*. ביצענו את האלגוריתם ללא חריגות ושינויים דרסטיים מעבר לשימוש ב-*beam search* . לצורך שיפור זמן הריצה נעשה שימוש בחישובים וקטוריים ומטריצות ספרסיות.

תחרות:

- הוספנו הרבה מאוד פיצ'רים, יש לנו בסה"כ 543,060 פיצ'רים ו 28 משפחות פיצ'רים בסה"כ.
- ביצענו אופטימיזציה על ה hyperparameters כמו ה threshold ומקדם הרגולריזציה.
- קראנו מאמרים כדי לחשוב אילו פיצ'רים כדאי להוסיף (למשל from 97% to 100%).
- בדקנו את הפיצ'רים שהוספנו כל פעם והאם הם באמת עוזרים למודל.

חלוקת עבודה:

החשיבה התבצעה ביחד אך בכתיבת הקוד והמימוש אליעזר עשה את שלב האימון, את האופטימיזציה והפיצ'רים הבסיסיים כדי לעבור את אחוזי הדיוק הבסיסיים ואילו קטיה מימשה את ה Viterbi והוסיפה משפחות פיצ'רים למודל כדי לשפר את המודל לתחרות ולהגיע למקומות גבוהים. החשיבה, הרעיונות וחלוקת העבודה התבצעו יחד.