רטוב 1 - NLP

מגישים: אמיר בלדר - 204179659 , רועי גנץ - 204506349

אימון

מודל 1

ראשית, בנינו את המודל שלנו בהתבסס על תכונות $f_{100} - f_{105}$, בהתאם להנחיות התרגיל.

כדי לשפר את הביצועים, הוספנו את התכוניות הבאות:

$$f_{106}(h,t) = \begin{matrix} 1\ if\ previous\ word\ w_{i-1}, and\ t = V\_t \\ 0\ otherwise \end{matrix}$$

$$f_{start\_capital}(h,t) = \begin{matrix} 1\ if\ the\ first\ letter\ is\ a\ capital\ letter\ and\ t = V\_t \\ 0\ otherwise \end{matrix}$$

$$f_{all\_capital}(h,t) = \begin{matrix} 1\ if\ all\ the\ letters\ are\ capital\ letters\ and\ t = V\_t \\ 0\ otherwise \end{matrix}$$

$$f_{is\_numeric}(h,t) = \begin{matrix} 1\ if\ word\ contains\ numbers\ and\ t = V\_t \\ 0\ otherwise \end{matrix}$$

$$f_{contains\_hyphen}(h,t) = \begin{matrix} 1\ if\ word\ contains\ hyphen\ and\ t = V\_t \\ 0\ otherwise \end{matrix}$$

עבור כל אחת מבין התכוניות, בדקנו שהיא אכן משפרת את הביצועים על קבוצת ההערכה טרם הוספתה. כמו כן, ההשראה לתכונות הללו הגיע מפרק 8 בספר Speech and Language Processing מאת Daniel Jurafsky & James H. Martin. תכוניות אלו נועדו בכדי ללמד את המודל לנתח את המבנה התחבירי והסינטקטי של המילים השונות ולהיעזר בהם לצורך תיוג נכון.

הגדרנו את וקטור ההיסטוריה באופן הבא - $h =< t_{i-2}, t_{i-1}, t, w_{i-1}, w_i, w_{i+1} >$. עבור תכוניות f_{101}, f_{102} נעזרנו בכל התחיליות והסופיות בגדלים 1-4. בנוסף, עבור כל תכונית אפשרית, ספרנו את מספר המופעים שלה במסמך האימון שלנו ולבסוף השתמשנו בתכוניות שמספר המופעים שלהן עבר ערך סף מסוים, thr, אשר מהווה היפר-פרמטר למודל שלנו. נציין כי בחרנו להשתמש ב-2 ערכי סף שונים, אחד המתאים לתכוניות נפוצות יותר, ואחד המתאים לתכוניות נדירות יותר. התכוניות הנפוצות הינן $f_{100}, f_{103}, f_{104}, f_{105}, f_{106}$ ואילו היתר נדירות יותר. התכוניות הנפוצות עוסקות בהקשר של המילה ובתיוגים בסביבתה, בעוד שהתכוניות הנדירות עוסקות במבנה המילים עצמן. בחרנו להשתמש ב-2 ערכי הסף כדי להביא לידי ביטוי הן את ההקשר והן את המבנה התחבירי של המילים.

להלן הגודל של כל f, עבור *train1.wtag* -

$\lvert f_{100}\rvert$	$\lvert f_{101}\rvert$	$\lvert f_{102}\rvert$	$\lvert f_{103}\rvert$	$\lvert f_{104}\rvert$	$\lvert f_{105}\rvert$	$\lvert f_{106}\rvert$	$\lvert f_{start\_capital}\rvert$	$\lvert f_{all\_capital}\rvert$	$\lvert f_{is\_numeric}\rvert$	$\lvert f_{contains\_hyphen}\rvert$
2221	7062	10501	2323	632	43	2186	2	2	10	3

ביצועים - המדד שבו נעזרנו לאומדן הביצועים הינו דיוק פר-מילה (*accuracy*), כלומר, $accuracy = \frac{words\ predicted\ correctly}{\#num\ words}$. לאחר כיוונון הפרמטרים, קיבלנו דיוק של **94.7**, כפי שנאמד על *test1.wtag*.

בחירת היפר-פרמטרים למודל -

המודל שלנו מכיל שני היפר-פרמטרים מרכזיים - λ המהווה מקדם הרגולריזציה, ו-*thrs* המהווים ערכי סף למספר מופעים של תכוניות מסוימות. כיוונון נכון של שני ההיפר-פרמטרים הללו היה חיוני להצלחת המודל. נציין כי לשני הערכים הללו חשיבות בקביעת ה-*model capacity* והאקספרסיביות של המודל. שימוש במקדם רגולריזציה גבוה מקטין את הסיכוי להגיע למצב של *overfitting*, אולם גבוה מדי, עשוי להביא אותנו ל*underfitting*. באופן דומה, שימוש בערך סף גבוה, יצמצם את כמות התכוניות ויביא למודל פחות עשיר, אולם שימוש בערך סף נמוך עשוי להביא למודל עשיר מדי, בעל יכולת ההכללה מוגבלת. אי לכך, ביצענו כיוונון היפר-פרמטרים בשיטת *grid-search*, תוך בדיקת ביצועים על קבוצת הערכה. הפרמטרים המיטביים שלנו עבור מודל 1 הינם *thr common = 7*, *thr rare = 3* ו- $\lambda = 2$.

מודל 2

קובץ האימון עבור מודל 2 מצומצם משמעותית ביחס לקובץ האימון עבור מודל 1 (250 משפטים לעומת 5000). כמו כן, עבור מודל 1 נתון לנו סט מתויג אשר יכול להוות קבוצת הערכה על ביצועי המודל לצורך בחירת הארכיטקטורה וכיוונון ההיפר-פרמטרים, אולם עבור מודל 2 לא נתון לנו סט כזה.

בתחילה, שקלנו לצמצם מראש את מספר התכוניות שבהן נשתמש עבור מודל זה, שכן מספר הדוגמאות המצומצם עשוי לפגוע ביכולת לאמן מודל עשיר. בסופו של דבר, בחרנו לנסות ערכי סף ורגולריזציה שונים אשר יאפשרו לשלוט בגודל האפקטיבי של המודל ולבחור את הפרמטרים שעבורם נקבל את התוצאות המיטביות.

אי לכך, בעבור מודל 2 הגדרנו רשימת היפר-פרמטרים למודל וכיוונו אותם. היות ולא הייתה ברשותנו קבוצת הערכה, היה עלינו להיעזר ב- *k-folds cross validation* על קבוצת האימון - פיצלנו את הקובץ *train2.wtag* ל-5 חלקים שווים, ובכל פעם, חמישית אחרת היוותה קבוצת הבקרה, בעוד שהיתר היוו את קבוצת האימון. ערכי ההיפר-פרמטרים שנבחנו הם $(thr\ rare, thr\ common) \in [(3,5),(3,7),(5,5),(5,7)]$, ו$\lambda \in [0.1, 0.5, 2, 5]$. עבור *thr rare = 3* ,*thr common = 5* ו- $\lambda = 0.5$ קיבלנו את התוצאות המיטביות ב- *k-folds cross validation* - **0.91**.

להלן הגודל של כל f, עבור *train2.wtag* -

$\lvert f_{100}\rvert$	$\lvert f_{101}\rvert$	$\lvert f_{102}\rvert$	$\lvert f_{103}\rvert$	$\lvert f_{104}\rvert$	$\lvert f_{105}\rvert$	$\lvert f_{106}\rvert$	$\lvert f_{start\_capital}\rvert$	$\lvert f_{all\_capital}\rvert$	$\lvert f_{is\_numeric}\rvert$	$\lvert f_{contains\_hyphen}\rvert$
179	992	1288	269	152	26	173	1	0	3	0

הערות ושיפורים כלליים -

ביצועים ויעילות - השקענו מאמץ ומחשבה רבה באופן המימוש כדי להשיג ביצועים מיטביים. מימשנו את הקוד בצורה וקטורית, דבר המאפשר להאיץ את הביצועים בצורה משמעותית. כמו כן, כדי לייצג את וקטור התכונות, נעזרנו במטריצות דלילות מסוג *csr_matrix*. השימוש בתצורה הזו של המטריצות גם עזר לנו להגיע להרצה היעילה. גולת הכותרת הייתה בחישוב ה- *normalization term*. יש אפשרות לבצע מספר חישובים במקביל, על ידי יצירת מטריצה שמכילה את כל האפשרויות השונות של $f(x_i, y')$ ולאחר מכן אפשר לחלק אותה בפעולות יעילות לחלקיה השונים להמשך החישוב. הודות לכך, זמן האימון של המודל על *train1.wtag* הינו כ-300 שניות, בעוד שהאימון של מודל 2 ארך כ-10 שניות בלבד. להלן מפרט המחשב בו השתמשנו :

Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

Installed memory (RAM): 4.00 GB (3.88 GB usable)

טריק יציבות נומרית - בעת תהליך האופטימיזציה אנו מחשבים את פונקציית ה- *softmax* אשר מורכבת מהביטוי $\frac{e^{f_i * v}}{\sum e^{f_j * v}}$. נציין כי הערך המתקבל בחישוב האקספוננט עשוי להביא ל*overflow*. כדי למנוע זאת, שמנו לב כי $\frac{e^{f_i * v}}{\sum e^{f_j * v}} = \frac{e^{-val} * e^{f_i * v}}{e^{-val} * \sum e^{f_j * v}} = \frac{e^{(f_i * v - val)}}{\sum e^{f_j * v - val}}$. לכן, חיסרנו הן מהמונה והן מהמכנה את הערך המקסימאלי עבור $f_i * v$, דבר המצמצם את גדלי הביטויים משמעותית, מבלי לשנות את ערך ה*softmax*.

הסקה

בשלב ההסקה, השתמשנו בגירסת "חיפוש האלומה" של אלגוריתם ויטרבי, כדי לצמצם את זמן הריצה. הודות לשינוי זה, במקום לבדוק את כל 44 אפשרויות התיוגים, נבדק קומץ אפשרויות בלבד. גודל האלומה היווה גם כן היפר-פרמטר של שלב ההסקה אותו גם כן כיוונו. הערך הנבחר הינו *beam_size = 3*. נציין כי גם כאן, נעזרנו בחישובים ובפעולות וקטוריות אשר מייעלות מאוד את הקוד ומקצרות משמעותית את זמן הריצה. אי לכך, הסקה על קובץ *test1.wtag* ארכה כ-100 שניות בלבד!

```
inference time :  117.58175015449524
```

מבחן

מודל 1

כפי שכתבנו בשלב האימון, בכדי לבחון את ביצועי מודל 1, נעזרנו בקובץ *test1.wtag* כקבוצת ההערכה שלנו. הביצועים אותם קיבלנו הינם כדלקמן:

```
training time =  242.66519856452942
inference time :  117.58175015449524
model's accuracy:  0.9477886199467748
```

להלן ה-*confusion matrix* אשר מציגה את 10 התגים אשר המודל טעה עליהם הכי הרבה (מופיע בקובץ *conf_mat.html*):

	#	$	''	*	,	-LRB-	-RRB-	.	:	CC	CD	DT	EX	FW	IN	JJ	JJR	JJS	MD	NN	NNP	NNPS	NNS	PDT	POS	PRP	PRPS	RB	RBR	RBS	RP	SYM	TO	UH	VB	VBD	VBG	VBN	VBP	VBZ	WDT	WP	WPS	WRB	``
WDT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0
RP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JJR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	76	0	0	10	1	0	0	0	0	0	1	0	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
#	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	1	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NNPS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	31	33	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FW	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PDT	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

נשים לב כי *NNPS* ממתויג רבות כ- *NPS*. כאמור, ההבדל בין השניים זה יחיד-רבים. כדי לשפר את הדיוק בתיוגים הללו, נוכל להוסיף תכונה אשר בודקת האם המילה הינה ברבים או ביחיד - על ידי בחינת הסיומת. כלומר, הוספת תכונה שבודקת האם מילה מסתיימת ב-*s*.

מודל 2

עבור מודל 2, לא קיבלנו קובץ אשר יכול להוות קבוצת הערכה. אי לכך, נעזרנו בפיצול קבוצת האימון ושימוש ב- *k-folds cross validation*. פיצלנו את הקובץ *train2* לחמישה קבצים, כל אחד בן 50 שורות, וביצענו *k-folds cross validation* כאשר *k-5*. נעזרנו אמצעי זה הן לצורך הערכת ביצועים, והן לצורך כיוונון פרמטרים. עבור סט הפרמטרים המיטבי, קיבלנו את התוצאות הבאות:

```
k-folds cross validation on model2 :
estimated accuracy :  0.9120175190383957
```

תחרות

מודל 1

כאמור, מודל 1 שלנו אשר התאמן על *train1*, קיבלנו בקירוב 95% דיוק. שמנו לב כי *train1* הינו טקסט בעל אופי כלכלי, *test1* בעל אופי של תחבורה אולם מגוון יותר. *Comp1* הינו בעל אופי מגוון אף הוא. בשל השוני בין קורפוס האימון לקורפוס התחרות, אנו מעריכים שנקבל אחוזי דיוק נמוכים במעט.

מודל 2

ניסינו לבחון את טקסט זה, ושמנו לב שבניגוד לכל קודמיו, הוא היה בעל אופי מדעי יותר. כלומר, שהוא הכיל מספר גדול יחסית של שמות באותיות גדולות ושל מספרים, לעיתים קרובות אף משולבים זה עם זה. עובדה זו חיזקה את בחירת התכונות שהוספנו מעבר לתכונות שהיו נתונות בתרגיל. עבור מודל 2 ביצענו הערכת ביצועים באמצעות *k-folds cross validation*. בשיטה זו, המידע בקבוצת האימון ובקבוצת ההערכה מגיע מפילוג דומה. אמנם הן *train2* והן *comp2* עוסקים בתחומים דומים, אנו מניחים שעדיין קיים שוני באופי הפילוג של הקבצים השונים, לכן נצפה לקבל תוצאות נמוכות במקצת מאלו שקיבלנו בעת הערכת הביצועים (91.2%).

חלוקת העבודה - עבדנו יחדיו בשיתוף פעולה מלא, ללא חלוקת עבודה.