איילה שעובי-מן

200244242

קורס מבוא לעיבוד שפה טבעית

ממ"ן 12

תאריך 18.4.18

Contents

סטטיסטיקה תיאורית

מדד	הסבר	Gold	Train	All
מופעי יוניגרים של סגמנטים	מספר שורות של סגמנט-תג ווליידיות בקורפוס	11282	127884	139166
סוגי יוניגרים של סגמנטים	מספר מופעים יוניקים של סגמנטים בקובץ	3171	15986	16845
מופעי סגמנט-תג	מספר שורות של סגמנט-תג ווליידיות בקורפוס	11282	127884	139166
סוגי סגמנט-תג	מספר תגים יוניקים בקורפוס	36	37	37
מדד העמימות	ממוצע על מספר מופעי התגים השונים בעבור סגמנט	1.07981072555	1.13487644667	1.13696271669

הבדלים בין קובץ האימון לקובץ הבדיקה

- **גודל קובץ האימון:** קובץ האימון מכיל נתונים רבים יותר, K127 אל מול K11 שורות וכן הדבל זה מתבטא גם במגוון של זוגות סגמנט-תג שונים.
- **מספר מופעים גבוה יותר לסגמנט:** קובץ האימון מכסה מספר רב יותר של סגמנטים, פי 5, וכן בעבור כל סגמנט ישנן מספר רב של שורות. בממוצע 8 מופעים לסגמנט בקובץ האימון ולכן ניתן להסיק כי ישנו ייצוג של אפשרויות טיוג רבות בעבור הסגמנטים שבקובץ האימון. זאת לעומת קובץ הבדיקה שבו בממוצע יש כ-3 מופעים לסגמנט.
משמעותי לביצועי המתייג כאשר מדובר על מודל למידה מבוקרת, שכן שורות בדיקה שלא נתקלנו בהן בקובץ האימון, יהיו כנראה בעלות דיוק נמוך יותר כאשר המודל יסווג אותן.
- **מדד העמימות בקובץ האימון גבוהה יותר**. זהו מצב הגיוני כי סביר שניתקל בריבוי אפשרויות לטיוג סגמנט ככל שאנו נתקלים ביותר דוגמאות מהשפה.
- **כלל התגים בקובצי הבדיקה מופיעים בקובץ האימון**. במידה וזה לא היה המצב, בעבור שורות אלו, המודל היה נותן חיזוי שגוי שכן לא יתכן שיחזיר תשובה שלא נתקל בה כלל בקובל האימון.
- **סגמנטים המופיעים בקובץ הבדיקה, אך לא מופיעים כלל בקובץ האימון:** בעבור סגמנטים אלו, ייתכן וביצועי המודל יהיה נמוכים יותר. ניתן לצמצם את פערי הדיוק באמצעות יוריסטיקות להחלקה.
- **שילובים של סגמנט-תאג אשר מופיעים בקובץ הבדיקה, אך לא נתקלנו בהם בקובץ האימון**: אם לא ראינו את כלל הדוגמאות האפשריות בעבור הסגמנט בשלב האימון, יתכן והמודל יחזיר תשובה שגויה. גם בעבור מצב זה, ניתן לצמצם את פערי הדיוק באמצעות יוריסטיקות להחלקה וניחוש התיוג הסביר ביותר.

המתייג הבסיסי

קוד: src/taggers/basic_tagger.py

סכמת הפרמטרים במודל

מרחב הקלט: סגמנט יחיד (יוניגרם) - w

מרחב הפלט: תג יחיד, הנפוץ ביותר בקובץ האימון בעבור סגמנט הקלט- $t_i\ or\ NAN$

סט הפלטים האפשריים למודל- $T = t_1, \dots t_n$

נוסחאות המשערכים של הפרמטרים במודל

$$f(w) = \begin{cases} argmax_{\{t|t\epsilon T\}} \widehat{P(t|w)} \\ NAN\ if\ total\ instances\ of\ w\ in\ train\ is\ 0 \end{cases}$$

$$\widehat{P(t|w)} = \frac{\boldsymbol{Count(w,t)}}{\boldsymbol{Count(w)}} = \frac{number\ of\ t\ tag\ rows\ for\ w}{total\ instances\ of\ w}$$

סיבוכיות זמן הריצה של המודל

אורך משפט מקסימלי בקובץ האימון= N_max

מספר תגים ייחודיים בקובץ האימון = T

מספר משפטים בקובץ האימון =K

סיבוכיות האימון

שלב זה דורש קריאת קובץ האימון ובחירת התג השכיח ביותר לכל סגמנט אפשרי.

במצב הגרוע ביותר, יש N_max*K סגמנטים סה"כ. לכל אחד מהם יש לבצע:

- ספירת מספר מופעים של סגמנט
- ספירת מופעים של סגמנט-תג
- בחירת תג בעל ערך מקסימלי לסגמנט

בהנחה כי שמירת הזוגות סמגנט-תג תיעשה באמצעות מערך ממוין, אינקרמנטציה של מספר מופעי סגמנט, או מספר מופעי סגמנט תג תיעשה בזמן O(log(N_max*K*T)).

כלומר, מעבר על N_max*K שורות לכל היותר, ולכל שורה אינקרמנטציה של הקטגוריה הרלוונטית עבורה מתוך N_max*K*T קטגוריות אפשריות.

סה"כ O(N_max*K* log(N_max*K*T))

סיבוכיות התיוג

אורך משפט הקלט – N סגמנטים

שלב זה דורש לכל סגמנט במשפט:

- מציאת הסגמנט הרצוי בקובץ האימון ושליפת התג בעל מספר מופעים מקסימלי (נתון ברגע שמצאנו את הסגמנט)- לוגריתמי במספר השורות בקובץ האימון (בהנחה שקובץ האימון אינו ממוין)

אם בשלב האימון היו לנו S סגמנטים ייחודיים: **סה"כ יש N סגמנטים במשפט, ולכן זמן הריצה הוא O(N*S)**

במצב הגרוע ביותר יש N_max*K סגמנטים ואז נקבל זמן ריצה O(N_max*K*N) בעבור קלט באורך N.

דיוק קובץ הבדיקה

- תיקיית קלט/פלט: results\ exp_1_basline_run_results , exps\ exp_1_basline_run

$$A_j = \frac{1}{n_j} \sum_{\{t'_{ij} | t_{ij} = t'_{ij}, i=1 \dots n_j} 1$$

$$\boldsymbol{seg - accuracy = A = \frac{\sum_{j=1}^{N} A_j * n_j}{\sum_{j=1}^{N} n_j} = 0.83061513916}$$

$$All_j = 1 \; iif \; \left(A_j == 1\right), otherwise \;\; All_j = 0$$

$$\boldsymbol{sen - accuracy = All = \frac{\sum_{j=1}^{N} All_j}{N} = 0.106}$$

מתייג מסדר ראשון

קוד: Src/taggers/first_ord_tagger_logprobs.py

פונקציית המטרה של המודל

- קלט= רצף מילים $w_1, \dots, w_n$
- פלט= רצף תגים $t_1, \dots, t_n$ בעלי הסתברות מקסימלית בהינתן רצף המילים

$$\boldsymbol{f_T(w_1^n) = argmax_{\{t_1^n | t_i \in T\}} P(t_1^n | w_1^n)}$$

נוסחאות הפרמטרים של המודל

$$t_1^n *= argmax_{\{t_1^n | t_i \in T\}} P(w_1^n | t_1^n) * P(t_1^n)$$

$$\boldsymbol{P(w_1^n | t_1^n) = \prod_{i=1}^{n} P(w_i | t_i)}$$

מהנחת אי תלות:

$$\boldsymbol{P(t_1^n) = \prod_{i=1}^{n} P(t_i | t_{i-1})}$$

לכן:

$$\boldsymbol{t_1^n *= argmax_{\{t_1^n | t_i \in T\}} \prod_{i=1}^{n} P(w_i | t_i) * P(t_i | t_{i-1})}$$

נוסחאות המשערכים של הפרמטרים במודל

- Emission probabilities

$$\boldsymbol{\widehat{P}(w_i | t_i) = \frac{Count(w_i, t_i)}{Count(t_i)}}$$

- Transition probabilities

$$\boldsymbol{\widehat{P}(t_i | t_{i-1}) = \frac{Count(t_i, t_{i-1})}{Count(t_{i-1})}}$$

המרה למרחב לוגריתמי

פונקציית המטרה של המודל

$$\boldsymbol{f_T(w_1^n) = argmin_{\{t_1^n | t_i \in T\}}(-1) * lnP(t_1^n | w_1^n)}$$

נוסחאות הפרמטרים של המודל

$$t_1^n *= argmin_{\{t_1^n | t_i \in T\}}(-1) * \ln[P(w_1^n | t_1^n) * P(t_1^n)]$$

$$= argmin_{\{t_1^n | t_i \in T\}}(-1) * [\ln P(w_1^n | t_1^n) + \ln P(t_1^n)]$$

$$lnP(w_1^n | t_1^n) = \sum_{i=1}^{n} P(w_i | t_i)$$

$$lnP(t_1^n) = \sum_{i=1}^{n} P(t_i | t_{i-1})$$

לכן:

$$\boldsymbol{t_1^n *= argmin_{\{t_1^n | t_i \in T\}} \sum_{i=1}^{n} (-lnP(w_i | t_i) - lnP(t_i | t_{i-1}))}$$

נוסחאות המשערכים של הפרמטרים במודל

- Emission log-probabilities

$$\boldsymbol{-ln\widehat{P}(w_i | t_i) = -ln\frac{Count(w_i, t_i)}{Count(t_i)} = -lnCount(w_i, t_i) + lnCount(t_i)}$$

- Transition log-probabilities

$$\boldsymbol{-ln\widehat{P}(t_i | t_{i-1}) = -ln\frac{Count(t_i, t_{i-1})}{Count(t_{i-1})} = -lnCount(t_i, t_{i-1}) + lnCount(t_{i-1})}$$

$$\boldsymbol{\widehat{t} *= argmin_{\{t_1^n | t_i \in T\}} \sum_{i=1}^{n} \left(-ln\frac{Count(w_i, t_i)}{Count(t_i)} - ln\frac{Count(t_i, t_{i-1})}{Count(t_{i-1})} \right)}$$

סיבוכיות זמן הריצה של אלגוריתם האימון

אורך משפט מקסימלי בקובץ האימון= N_max

מספר תגים ייחודיים בקובץ האימון = T

מספר משפטים בקובץ האימון =K

מספר סגמנטים מקסימלי בקובץ האימון =K*N_max

1. חישוב emission probabilities

שלב זה דורש קריאת קובץ האימון וחישוב לכל צירוף סגמט-תג את האומדן $\widehat{\boldsymbol{P}}(\boldsymbol{w_i}|\boldsymbol{t_i})$. כלומר:

השוואה בזוגות של סגמנט- תג. $O(N_{max} * K * \log(N_{max} * K * T))$

2. חישוב transition probabilities

שלב זה דורש לכל זוג תגים עוקבים, חישוב האומדן $\widehat{\boldsymbol{P}}(\boldsymbol{t_i}|\boldsymbol{t_{i-1}})$**.** כלומר השוואה בזוגות של תגים.

לכל שילוב 'T-T, במקרה הגרוע ביותר, נעבור על כל השורות בקובץ האימון, כלומר

$$O(T^2 * K * N_{max})$$

סה"כ $\boldsymbol{O\big(K * N_{max} * (T^2 + \log(N_{max} * K * T))\big)}$

סיבוכיות זמן הריצה של אלגוריתם התיוג

בעבור **משפט קלט עם N מילים**, האלגוריתם יבצע כ- $\boldsymbol{O(T^2 * N)}$ צעדים. (N שלבים ובכל שלב מבצעים מעבר על זוגות מצבים).

אם נניח כי גישה לפלט האימון היינו בעל זמן קבוע (שמירת המידע במפות) זמן התיוג יהיה כנ"ל.

סה"כ $\boldsymbol{O(T^2 * N)}$

דיוק macro-avg בעבור קובץ הבדיקה

- תיקיות פלט/קלט: results\exp_2_hmm_tagger_results , exps\exp_2_hmm_tagger
- בשלב זה מעברים לא ידועים או מצבים בהם $\widehat{\boldsymbol{P}}(\boldsymbol{w_i}|\boldsymbol{t_i})$ בעל ערך 0 קיבלו את הערך 0. כלומר **:**

$$V_i(s,w_i) = \begin{cases} if\ exist\ s'in\ step\ i-1\ such\ that\ p(s|s') > 0\ and\ P(w_i|s) > 0\ than\ max_{s'\in T} V_{i-1}(s') * P(s|s') * P(w_i|s) \\ if\ exist\ s'in\ step\ i-1\ such\ that\ p(s|s') > 0\ and\ P(w_i|s) = 0\ than\ 0 \\ if\ P(w_i|s) > 0\ and\ \ p(s|s') = 0\ for\ each\ step\ in\ i-1\ with\ V_{i-1} > 0\ than\ 0 \end{cases}$$

$$B_i(s,w_i) = \begin{cases} if\ exist\ s'in\ step\ i-1\ such\ that\ p(s|s') > 0\ and\ P(w_i|s) > 0\ than\ argmax_{s'\in T} V_{i-1}(s') * P(s|s') * P(w_i|s) \\ rand\ s' \in T \end{cases}$$

- אם כל המצבים בעלי ערך 0 לשלב , יבחר מצב באופן רנדומלי להיות המצב הקודם של S.
- **הדיוק המתקבל:**

$$\boldsymbol{seg - accuracy = A = \frac{\sum_{j=1}^{N} A_j * n_j}{\sum_{j=1}^{N} n_j} = 0.836982537009}$$

$$\boldsymbol{sen - accuracy = All = \frac{\sum_{j=1}^{N} All_j}{N} = 0.151696606786}$$

החלקה למעברים לא ידועים

- תיקיית קלט/פלט: results\exp_3_hmm_tagger_results , exps\exp_3_hmm_tagger
- ההחלקה תשמש את המודל במצבים בהן לכל מצב אפשרי t_i, מתקיים:

$$\hat{P}(w_i|t_i) = 0 \; or \; \hat{P}(t_i|t_{i-1}) = 0$$

- כלומר, לא קיים מצב באיטרציה נוכחית בעבורו הנוסחה $max_{s' \in T} V_{i-1}(s') * P(s|s') * P(w_i|s)$ מספקת ערך גבוהה מ-0.
- אתייחס למצבים האפשריים כאוסף המצבים עבורם $\hat{P}(t_i|t_{i-1}) > 0$ (3) $\hat{P}(w_i|t_i) > 0$ (2) or.

$$V_i(s, w_i) = \begin{cases} (1) if \; exist \; s' \; in \; step \; i-1 \; such \; that \; p(s|s') > 0 \; and \; P(w_i|s) > 0 \; than \; \boldsymbol{max_{s' \in T} V_{i-1}(s') * P(s|s') * P(w_i|s)} \\ max_{s'} \begin{cases} (2) \; if \; not \; (1) \; and \; exist \; s' \; in \; step \; i-1 \; such \; that \; p(s|s') > 0 \; and \; P(w_i|s) = 0 \; than \; \boldsymbol{max_{s' \in T} V_{i-1}(s') * P(s|s')} \\ (3) \; if \; not \; (1) \; and \; P(w_i|s) > 0 \; and \; p(s|s') = 0 \; for \; each \; step \; in \; i-1 \; with \; V_{i-1} > 0 \; than \; \boldsymbol{max_{s' \in T} V_{i-1}(s') * P(w_i|s)} \end{cases} \end{cases}$$

$$B_i(s, w_i) = \begin{cases} if \; exist \; s' in \; step \; i-1 \; such \; that \; p(s|s') > 0 \; and \; P(w_i|s) > 0 \; than \; argmax_{s' \in T} V_{i-1}(s') * P(s|s') * P(w_i|s) \\ argmax_{s' \in T} \{\boldsymbol{V_{i-1}(s') * P(s|s')}, \boldsymbol{V_{i-1}(s') * P(w_i|s)}\} \end{cases}$$

- הדיוק המתקבל:

$$\boldsymbol{seg - accuracy = A = \frac{\sum_{j=1}^{N} A_j * n_j}{\sum_{j=1}^{N} n_j} = 0.866235262831}$$

$$\boldsymbol{sen - accuracy = All = \frac{\sum_{j=1}^{N} All_j}{N} = 0.157684630739}$$

ניתן לראות לפי המדדים כי תוצאות התיוג השתפרו מאוד ביחס להרצה ללא החלקה כלל.

Seg-accuracy עלה בכ-0.3 נקודות וכן אחוז המשפטים המתויגים באופן נכון לחלוטין עלה מ-15.2% ל- 15.8%.

החלקה למילים לא ידועות

- תיקיית קלט/פלט: results\exp_4_hmm_tagger_results , exps\exp_4_hmm_tagger
- החלקה למילים לא ידועות בנוסף להחלקה למעברים לא ידועים (משלב קודם)
- ההחלקה למילים לא ידועות (במקום טיוגן כ-NNP) תבוצע באופן הבא:
  - בעבור מילה w_i התג שייבחר יהיה בהתאם להסתברויות המעבר של השלב הקודם. כלומר נבחר את התג בעל $\hat{P}(t_i|t_{i-1})$ מקסימלי. ערכו יהיה $\boldsymbol{V_{i-1}(t_{i-1}) * P(t_i|t_{i-1})}$ והמצב הקודם בעבורו יהיה t_{i-1}.
- **הדיוק המתקבל:**

$$seg - accuracy = A = \frac{\sum_{j=1}^{N} A_j * n_j}{\sum_{j=1}^{N} n_j} = 0.867753944336$$

$$sen - accuracy = All = \frac{\sum_{j=1}^{N} All_j}{N} = 0.168$$

ישנו שיפור נוסף של המדדים לאחר ההחלקה של מילים לא ידועים אם כי לא באופן משמעותי דרסטי.

ההשפעה העיקרית על Seg-accuracy שעלה ב1% משלב קודם.

ניתוח התוצאות

מטריצה הבלבול

- קוד: Src/scripts/confusion_matrix.py
- תיקיית קלט/פלט: exps\exp_4_confusion_matrix, results\exp_4_confusion_matrix_results
- המודל אשר הניב את הדיוק הטוב ביותר הינו המודל האחרון אשר נבדק, HMM מסדר ראשון, עם החלקה למילים לא ידועות ומעברים לא ידועים.
- שלושת השגיאות הנפוצות ביותר במודל הנבחר הן:

Tag-gold	Tag-model	Number of confusions
VB	IN	100
NNT	NN	96
NNP	IN	84

תיוג משפט

"אישה נעלה נעלה נעלה, נעלה את הדלת בפני בעלה"

קבצי קלט / פלט: exps\tag_sentence_4_2, results\tag_sentence_4_2

תיוג ידני אל מול תיוג המודל:

תרגום המשפט לסגמנטים	AIFH	NELH	NELH	NELH	yyCM	NELH	AT	H	DLT	BPNI	BELH	yyDOT
תיוג ידני	NN	NNT	VB	NN	yyCM	VB	AT	H	NN	IN	NN	yyDOT
תיוג ע"י מתייג	NN	IN	NN	IN	yyCM	IN	PRP	H	NN	IN	NN	yyDOT

- **היכן צדק המתייג?** במילים אשר יש להם ייצוג נרחב בקובץ האימון, וכן הם תואמים לרצף של זוגות תגים מסוים , ניתן לראות כי המתייג צדק.
- **היכן טעה המתייג?** מכיוון שאנו משתמשים במודל הלוקח בחשבון זוגות של טיוגים וכן זוגות של סגמנט-תג, אין התחשבות בזוגות מילים המגיעות אחת לאחר השנייה בבחירת תג המתאים. במילים בעלות עמימות גבוהה וכפל משמעות, המתייג יטעה בסבירות גבוהה.
- **אחוז הדיוק של המודל**, ביחס לתיוג הידני, בעבור משפט זה הינו 58.3%
- האינפורמציה החסרה במודל לשם שיפור הדיוק הינה:

1. **התייחסות לסימני פיסוק כמילים שונות**. לדוגמא, הצירוף ", נעלה..." היה מאפשר להבין כי "נעלה" הינו פועל (שמבוצע ע"י האישה)
2. **התחשבות ברצף מילים בנוסף להתחשבות ברצף התגים**. במצב זה לדוגמא, היינו יכולים לזהות כי "נעלה את..." מתייחס לפועל (שכן "את" מגיע לאחר מכן)

   דוגמא נוספת היא המופע הראשון של "נעלה" המגיע לאחר תג "NN" ותויג כ- "IN". גם המופע השלישי של "נעלה", מכיוון שהמופע השני תויג כ- "NN", תויג כ-"IN" עקב תג המקדים, ללא התחשבות שיש רצץ של שני תגים זהים.
3. אפשרות נוספת היא **שימוש במתייג מסדר גבוהה יותר**. זה יפתור מצבים בהם יש רצפים פחות הגיוניים כגון "NN-IN-NN-IN"

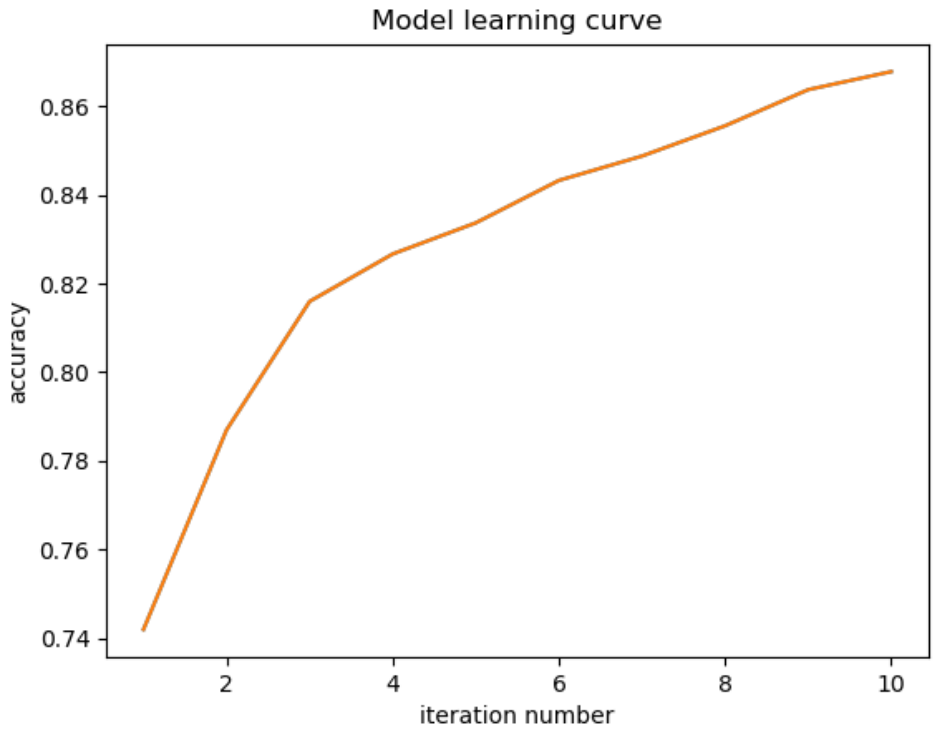
עקומת למידה של המודל

- קוד src/scripts/learning_curve.py
- תיקיית קלט/פלט: exps\exp_4_learning_curve, results\exp_4_learning_curve_results
- המודל אשר הניב את הדיוק הטוב ביותר הינו המודל האחרון אשר נבדק, HMM מסדר ראשון, עם החלקה למילים לא ידועות ומעברים לא ידועים.

הערכים המתקבלים בכל איטרציה

itr	1	2	3	4	5	6	7	8	9	10
acc	0.74197837	0.78700585	0.81545825	0.82653785	0.83371743	0.84337883	0.84878568	0.85561071	0.86367665	0.86775394

עקומת הלמידה של המודל



סיכום ומסקנות

תיאור סכמתי של הקוד

שפת התכנות בה בחרתי להשתמש הינה python.

1. הקוד מכיל 4 חבילות תחת Src:
   - Datasets- מכילה את קבצי הנתונים ואת הלוגיקה לפירוק קבצי המידע
   - Evaluation- מכילה את הלוגיקה לחישוב מדדי ההערכה ומדדים אגרגטיביים
   - Taggers- מכילה את המתייגים השונים (מתייג בסיסי, מתייג מסדר ראשון וכן מתייג מסדר ראשון לאחר המרה למרחב לוגריתמי)
   - Scripts- מכילה סקריפטים לביצוע של חלקיו השונים של הדו"ח (מטריצת הבלבול, יצירת עקומת הלמידה)
2. הסקריפטים הבאים לביצוע אימון, קידוד והערכה בהתאמה: train.py, decode.py, evaluate.py מוציאים לפועל את הפעולות השונות בהתאם לקלט (מודל נבחר, קובץ האימון וכו')
3. במהלך העבודה על הדו"ח ביצעתי הרצות שונות של הסקריפטים הנ"ל.
   קלטיים לכלל ההרצות, ימצאו תחת תיקיית exp, כל ניסוי תחת תיקייה ייעודית בה ייצאו קבצי הקלט, פירוט שורת הקריאה לסקריפטים הנ"ל, והסבר נוסף אם נדרש. הפלט התואם לניסוי ימצא תחת תיקיית results בשם תואם.

תוצאות ומסקנות מהתהליך

כיוצא מתוצאות הדו"ח, אנו רואים כי באמצעות המתייג הבסיסי, אדר לוקח בחשבון כל סגמנט באופן בלתי תלוי לקודמיו במשפט, ניתן להגיע לדיוק של כ-83%.

כאשר מוסיפים למתייג את התלות בסגמנטים מקדימים לסגמנט הנבחן, הדיוק עולה באופן משמעותי לכ- 86%.

כלומר, לסגמנטים המקדימים של ישנה משמעות רבה בבחירות התג המתאים. ישנם סגמנטים אשר תמיד יאופיינו כעוקבים לסגמנטים אחרים.

בשלב השלישי, הוספנו התמודדות עם מילים ומעברים לא ידועים, מה שתרם לשיפור הדיוק אף יותר ל- 86.7%.

ניתן להסיק אם כן, כי ע"י בחינה נוספת של שתי הנקודות הבאות, ניתן לשפר את הדיוק אף יותר:

- בחינה של מתייגים מסדר גבוהה יותר מ-1
- סקירה של טכניקות החלקה מתוחכמות יותר, לטיפול במילים ומעברים לא ידועים

בנוסף, באמצעות בחינת עקומת הלימוד של המודל האחרון אשר נבחן, ניתן לראות כי ישנה משמעות גבוהה לגודל מאגר הנתונים המתויג המשמש ללימוד המודל. אף על פי שהמידע המשמש להערכה נותר אותו הדבר לכל אחת מהאיטרציות, ככל שגודל סט האימון עולה, כך הדיוק של המודל עולה.

הדיוק הסופי של המודל, בהינתן סט הנתונים הנוכחי, הינו 86.7%, אך שיפוע עקימת הלמידה טרם התייצב, ומכך ניתן ללמוד כי ע"י הגדלת סט הנתונים המתויג ניתן לשפר את דיוק המודל במידה ניכרת.

מענה על שאלות מחקריות לגבי המתייג בעברית

תהליך העבודה על המתייג נעשה על בסיס מידע מתויג של סגמנטים בשפה העברית.

ישנם שני הבדלים עיקריים בין השפה העברית לבין השפה האנגלית והם חוסר הקשיחות בסדר הופעת המילים בשפה העברית לעומת השפה האנגלית וכן גודל המידע המתויג הזמין בעבור מודלים לשפה העברית, שהינו קטן בהרבה מזה הזמין בשפה האנגלית.

כפי שהשתמע מפסקה קודמת, לגודל המידע המתויג משמעות רבה על דיוק המודל וניתן לראות זאת בבירור בעקומת הלמידה של המודל.

בנוסף, ככל שהדפוסים בשפה יותר קשיחים וחזרתיים, כך ניתן לתייגם בדיוק גבוהה יותר. בהסתכלות במטריצת הבלבול, ניתן לראות שהבלבול הגבוהה ביותר הינו בין VB ל- IN. אחת מהסיבות לבלבול זה הינה חוסר הקשיחות בסדר המילים. לדוגמא, המשפט "**חגגתי** את יום העצמאות **בים**" ניתן גם לביטוי כ- "את יום העצמאות **חגגתי בים**".

דרכים אפשריות לטיוב ושיפור עתידי של המתייג

- שימוש במתייג מסדר גבוהה יותר (3-gram, 4-gram)
- הגדלת סט הנתונים המתויג, בפרט, בעבור תגים המדורגים גבוהה במטריצת הבלבול
- שימוש בטכניקת החלקה מתוחכמות