(67658) עיבוד שפה טבעית | עבודת סיום

עמוס שטראוס | 311372833

19 בפברואר 2021

שאלה 1

בשאלה הזו נשתמש במודל CRF מסדר 1 (ביגרם), שאוסף התגיות הוא:

$L = \{None, Begin - Person, Continue - Person, Begin - Location, Continue - Location\}$

נכתוב כקיצור $B - Location, C - Location, B - Person, C - Person$ עבור $Begin/Continue$ בהתאמה.

הערה, לא הבנתי את הקלט כל כך והתשובה במייל הייתה שאני יכול להחליט כל עוד הוא ברור:

"למעשה אתה שואל על צורת הקלט שאתה מקבל. אתה מקבל משפט מתויג בנוטציה 1 כקלט.

אתה רשאי להחליט על צורת הקלט כל עוד אתה מסביר אותה בפירות, והיא מתאימה לנוטציה 1. "

לכן אניח שצורת הקלט היא משפט שכל מילה (או סימן פיסוק) בו או לא מקבלים תיוג בכלל או

מקבלים רק תיוג מתוך $Person, Location$ למשל בדוגמה:

נוטציה (1)

$(John, Person)\,(lives)\,(in)\,(New, Location)\,(York, Location)\,(City, Location)$

תתורגם לנוטציה (2)

$(John, B - Person)\,(lives, None)\,(in, None)\,(New, B - Location)\,(York, C - Location)\,(City, C - Location)$

(א)

הפסודוקוד יעבור על הקלט לפי נוטציה (1) וכל רצף של מילים מתויגות יהפוך לרצף של של מילים מתויגות עם נוטציה (2) כאשר המילה הראשונה ברצף תקבל B=being ואז את התיוג המקורי (being/person) ואחריה ברצף המילים יהיו עם התיוג המקורי וC=continue, עד לסיום הרצף המתוייג.

- קלט $x = (x_1, l_1), .., (x_n, l_n)$ משפט מתויג לפי נוטציה (1)
- לכל $i \in \{1, .., n\}$

(1) אם $i = 1$ בדוק האם x_1 מתוייג עם $l \in \{Person, Location\}$

אם כן הוסף את המילה של x_1 עם תיוג מתאים $(x_1, B - l)$.

אם x_1 לא מתוייג הוסף את $(x_1, None)$. קדם את i והמשך למילה הבאה.

(2) אם $i > 1$ בדוק האם x_i מתוייג עם $l \in \{Person, Location\}$,

אם הוא לא מתוייג הוסף את המילה x_i בתיוג המתאים $(x_i, None = l_i')$. קדם את i והמשך.

אם הוא כן מתוייג עם $l_i \in \{Person, Location\}$ בדוק האם באיטרציה הקודמת תוייגה (x_{i-1}, l'_{i-1})
אם l'_{i-1} מאותו טיפוס של $entitiy$/תיוג (מיקום או אישיות אצלנו) כמו של l_i.
- אם $l_i = Person$ וגם $l'_{i-1} \in \{B - Person, C - Person\}$ הוסף את $(x_i, C - Person = l'_i)$. קדם את i והמשך.
- אם $l_i = Location$ וגם $l'_{i-1} \in \{B - Location, C - Location\}$ הוסף את $(x_i, C - Location)$. קדם את i והמשך.

אחרת בהכרח l'_{i-1} מתיוג $None$ כלומר l_{i-1} הקודם לא תויג והנוכחי כן, או שl'_{i-1} לא מאותו טיפוס תיוג (למשל הקודם l_{i-1} $Location$ והנוכחי l_i הוא $Person$ או ההפך) בכל מקרה המילה ה־i היא ראשונה ברצף הוסף את $(x_i, B - l_i = l'_i)$ והמשך למילה הבאה.

(ב)

נתחיל מדוגמאות: נשים לב שאם קיבלנו $(Israel, Loaction)$ $(Jerusalem, Location)$ זה בהכרח יתרוגם ל־ $(Israel, C - Location)$ $(Jerusalem, B - Location)$.
אף פעם לא ל$(Israel, B - Location)$ $(Jerusalem, B - Location)$
או $(America, Location)$ $(Of, Location)$ $(States, Location)$ $(United, Location)$
בהכרח יתורגם ל$(United, B - Location)$ ו3 המילים שבאות אחריה יקבל $C - Location$. אין אפשרות לקבל פה תיוג התחלת מיקום לאמריקה $(America, B - Location)$.
דוגמה נוספת נניח שיש ביטוי של צמד אנשים כמו לנון מקרטני, שני אנשים שונים אך יקבלו תיוג של אישיות והמשך אישיות אם נקבל את הקלט למשל $[Lennon\,Maccartney]_{Person}$.
כלומר אי אפשר לקודד רצף של אותו טיפוס/$entity$ בצורה שונה מ־"התחלה(טיפוס),המשך(אותו טיפוס),..,המשך(אותו טיפוס)".
כלומר כל רצף תגיות אפשרי כזה:
$(x_1, B - l), (x_2, B - l), ...(x_k, B - l)$ כאשר $l \in \{Person, Location\}$ לא יתקבל ולא חוקי לפי הגדרת השאלה.
כלומר התיוג לפי נוטציה (1) מוגבל יותר ולעיתים לא נוכל לקודד רצף של $entities$ שונים מאותו טיפוס $Person/Location$, בעוד שהתיוג לפי נוטציה (2) עשיר יותר וכן יכול.

(ג)

מסעיף זה והלאה יהי $x = x_1, .., x_n$ משפט נתון כלשהו $B(x), A(x)$ כמו שסומנו בשאלה.
נשים לב ש$e^t > 0$ לכל $t \in \mathbb{R}$. לכן עבור רצף $y = y' = B - l$. כאשר $l \in \{Person, Location\}$.
נקבל $M_i(y, y') = \exp\left(w \cdot f(i, y', y, x)\right) > 0$.
כמו כן $Z(x; w) = \sum_{y \in L^n} \prod_{i=1}^{n} \exp\left(w \cdot f(i, y_i, y_{i-1}, x)\right) > 0$, כסכום מכפלות (סופיות) של גורמים חיובים.

נובע כי גם מילוי הרצפים עד או החל מהמקום ה$i \in [n]$ כלומר $\alpha_i(y) = \sum_{y_1, .., y_i = y} \prod_{k=1}^{i} M_k(y_{k-1}, y_k)$,
ובדומה $\beta_i(y)$ יהיו חיובים, כמכפלות סופיות של חיוביים וסכומים של מכפלות כאלה, כל אלה משמשים באלגוריתם לחישוב ההתפלגות השולית של צלע.
לכן לא משנה מי הם הפיצ'רים והמשקלים w, f של המודל תמיד נקבל שיש הסתברות חיובית לעבור בצלע שתיצור רצף לא חוקי.
ולכן גם עבור רצף לא חוקי ויכיל לפחות מעבר אחד בצלע לא חוקית, $y \in A(x)$ נקבל

$$Pr\left(y|x\right) = \frac{\prod_{i=1}^{n} \exp\left(w \cdot f(i, y_i, y_{i-1}, x)\right)}{Z(x; w)} > 0$$

כיוון שזו מנה ומכפלה (סופית) של גורמים חיוביים ממש.
(נזכור כי $y_0 = *$ או $y_0 = START$ במקרי הבסיס לכן הסימונים מוגדרים)

(ד)

בהנתן מודל CRF מאומן לבעייה זו. נעזר ב M', α' מערכים בגודל $n|L|$ ו $n|L|^2$ בהתאמה שממלאים בעזרת תכנון דינאמי. נרצה להסיר את הצלעות שמחברות רצפים לא חוקיים, כלומר כל צלע $(y_j, y_{j-1}) \in L^2$ כך ש:
$y_j = y_{j-1}\, and\, y_j \in \{B-Person, B-Location\}$. **(*)**
כיוון שכל כניסה בהן חיובית כמו שראינו ונרצה לחשב רק את $y \in B(x)$ נעזר בתכנון דינאמי של M' נמלא כך:

$$M_i'(y_j, y_{j-1}) = \begin{cases} 0 & y_j = y_{j-1}\ and\ y_j \in \{B-Person, B-Location\} \\ \exp\left(w \cdot f(i, y_j, y_{j-1}, x)\right) & else \end{cases}$$

ככה כל מסלול שעובר דרך צלע לא חוקית יתאפס.
כלומר כל מסלול שמכיל רצף תיוגים לא חוקי עבור משפט נתון $y \in A(x)$ יכיל תת רצף לא חוקי, ולפחות צלע אחת לא חוקית (y_j, y_{j-1}) ולכן לפחות אחד מהגורמים במכפלה של y כזה יתאפס, וכל המכפלה גם.
כעת נוכל לחשב על ידי M_i' בתכנון דינאמי את:

$$\alpha_i'(v) = \sum_{v'} \alpha_{i-1}'(v') M_i'(v', v)$$

גם כאן צלעות שאינן חוקיות יתאפסו ולא ייסכמו, לכן $\alpha_i'(v)$ מכיל את סכום כל המסלולים עד למילה הi במשפט הנתון, שלא עוברים דרך צלע לא חוקית, ולכן את כל המסלולים החוקיים בלבד.
בסוף נחשב את $\mathcal{B}(x)$ בעזרת α' כך $\mathcal{B}(x) = \sum_{v'} \alpha_{n+1}'(v')$, ונחזיר את $\frac{\mathcal{B}(x)}{Z(x)}$.
כאשר $Z(x)$ סוכם את כל המסלולים לפי הסכימה הרגילה שראינו בהרצאות ובאופן יעיל תוך שימוש בתכנון דינאמי.
הביטוי $\mathcal{B}(x)$ יכיל את סכום ההסתברויות של כל רצפי תגיות חוקיים על פי המודל הנתון כי צלעות במסלולים האלה גם לא שונו, בין M (ההסתברות השולית של צלע במודל שקיבלנו) וM' שחישבנו.
לסיכום כל מסלול חוקי לא יכיל שום צלע לא חוקית, וישאר כמו שהוא, וכל מסלול לא חוקי יתאפס, נסיק כי :

$$\sum_{y \in B(x)} Pr\left(y_1, .., y_n | x_1, .., x_n\right) = \frac{\mathcal{B}(x)}{Z(x)}$$

והחישוב יעיל.
*הערה על יעילות
לקבלת M' שהיא כאמור טבלה בגודל $|L|^2 \cdot n$ (השינוי בין M הרגילה שראינו בהרצאות לM' למעשה בעלות הבדיקה הקבועה האם הצלע חוקית או לא ואז לפי הצורך או ששמים 0 או אותו חישוב כמו במקרה הרגיל שבו מחשבים אתM).
החישוב של Z' דומה לZ בCRF רגיל אחרי שמלאנו את α' בדומה לα על ידי תכנות דינאמי לפי הנוסחה שהוזכרה בעלות $n \cdot |L|$ תאים ומילוי כל תא מסתכל רק על השורה הקודמת של התגים כלומר בעלות $O(|L|)$ סך הכל מילוי ב$O(n \cdot |L|^2)$.

שאלה 2

(א)

אי אפשר לשלב את האורך הכולל בתווים של עץ T כפי שמוגדר בשאלה (סכום אורכי הצלעות בתווים עבור הצלעות שבT) כפיצ׳ר ב$edge-factored\,model$ (EFM).
נגדיר את המשימה ואז נראה את הבעיתיות של הפיצ׳ר המוצע.
באופן כללי בEFM לכל צלע יש משקל שמתקבל על ידי מכפלה בין וקטור משקולות θ ופונקציית פיצ׳רים Φ,
עבור פונקציית פיצ׳רים המקבלת שני קודקודים v_1, v_2 שנחשוב עליהם כצלע אפשרית בפרסינג (הקודקודים מייצגים מילים במשפט ומנסים לאמוד חיבור אפשרי ביניהם) ואת המשפט x. נקבל את התוצאה מפונקציית הפיצ׳רים $\Phi\left(v_1, v_2, x_{1:n}\right)$.
בשאלה עוסקים ב־ Unlabeled Dependency Parsing. במקרה עם תיוג נעביר גם תיוג כלשהו $l \in TAGS$, $\Phi\left(v_1, v_2, l, x_{1:n}\right)$.
כאשר אנחנו עוסקים ב$Graph\,based\,parsing$, לפי תבנית העבודה שלמדנו בהנתן משפט $x_{1:n}$ אנחנו רוצים למצוא את ה$MST\ Parser$ אנחנו מוצאים בכל שלב את משקול/ניקוד כל צלע אפשרית על ידי

$$score_\theta\left(v_1, v_2, x_{1:n}\right) = \theta^T \cdot \Phi\left(v_1, v_2, x_{1:n}\right)$$

בתהליך הזה כפי שראינו דרושים:

1. מישקול/ניקוד של הצלעות. צריך לעשות באופן **בלתי תלוי** ולשקף עד כמה גבוהה סבירות הצלע להופיע בעץ הפירסור.
2. למצוא עץ פורש מקסימלי מכוון ($MST\,parser$) לפי הצלעות ומישקולם כפי שתאור ב1.

לאחר שהגדרנו את הבעיה נוכל לשים לב שפיצ׳ר כמו האורך הכולל בתווים של עץ T, אינו פיצ׳ר שיכול להיות מחושב באופן בלתי תלוי.
זאת מאחר שכדי לחשב את האורך הכולל בתווים ולשקלל אותו ב$score_\theta\left(v_1, v_2, x_{1:n}\right)$ של צלע כלשהי(v_1, v_2),
צריך לדעת מי הן כל שאר הצלעות שנמצאות איתה כבר בעץ.
כלומר הפיצ׳ר הזה תלוי בבחירת כל שאר הצלעות ולא רק בזו שהעברנו ל$score_\theta$, כלומר זה לא פיצ׳ר מתאים לEFM כי הדרישה שתהליך המישקול של כל צלע יהיה בלתי תלוי באחרות.
נשים לב שבעצם אם היה פיצ׳ר כזה עבור EFM אז היינו יודעים לשקלל את סכום אורכי הצלעות בT כפיצ׳ר עבור צלע כלשהי, בפרט היינו יודעים לחשב את סכום אורכי הצלעות של T זה אומר שאנחנו כבר יודעים את סכום אורך כל צלע אחרת בT.
אז חוזרים לנקודה שבשביל לדעת למשקל את הצלע צריך לדעת מראש מי הן הצלעות שנבחרו (כדי לחשב את מספר האותיות במילים ביניהם).
אמנם בשביל לדעת מי הן הצלעות שנבחרו עבור עץ T זה אומר שעלינו לדעת מראש מי הוא העץ הפורש המדובר T.
כדי למצוא את ה$MST\,parser$ בשלב (2) צריך שוב לדעת מה המשקול שהתקבל מהפיצ׳רים בשלב (1) נתנו לכל צלע, זה מצב מעגלי שמוביל לסתירה לאופן פעולת האלגוריתם.
כאמור הפיצ׳ר לוקח בחשבון תלות בין צלעות והוא אינו בלתי תלוי בהנתן צלע בודדה.

(ב)

נראה דוגמאות לפיצ׳רים של עץ T ($parse\,tree$) שלא ניתן לייצג באמצעות שני קודקודים אך ניתן באמצעות שלושה.
ההבדל בין המודלים EFM ו$grandparent\,model$ הוא שהמישקול/ניקוד בשני מתאפשר גם על שלושה קודקודים רצופים המהווים רצף קשתות של אב בן ונכד או (ילד אבא סבא איך שרוצים להסתכל על זה).

במובן שהפיצ'רים מוגדרים על שלשות כלומר 2 זוגות צלעות בקשר המתואר שיוצרות שרשרת (ולא רק על זוג קודקודים כלומר צלע אחת בנפרד).
פורמאלית עבור $v_i, v_j, v_k \in V$ כלשהם כך ש$(v_i, v_j) \in T$ וגם $(v_j, v_k) \in T$
כעת המישקול/ניקוד של של צלע אחת יכול להשפיע (לחיוב או לשליליה) על המישקול של צלע אחרת אם הן באותה שרשרת.
כלומר המישקול עבור שרשרת בה מופיעות שתי הצלעות $score_\theta (v_1, v_2, v_3, x_{1:n})$, יכול להיות מושפע לטובה או לרעה בהנתן נוכחות של אחת מהצלעות בשרשרת $(v_i, v_j)\,(v_j, v_k)$.
במודל הזה מתאפשרת תלות בין 2 צלעות שונות באותה שרשרת, בשונה מהEFM שם כאמור הניקוד/משקול נעשה על כל צלע באופן בלתי תלוי.
דוגמה לפיצ'ר כזה תיהיה סוג של $TRIGRAM$, שישקלל מידע על הPOS של שרשרת קודקודים v_i, v_j, v_k כך שv_j נמצא ביניהם כלומר $(v_i, v_j) \in T$ וגם $(v_j, v_k) \in T$.
נניח שהפיצ'ר בינארי והוא אינדיקטור עבור שלשה של POS מסויימת.
דוגמאות שאני יכול לחשוב בהן פיצ'ר כזה יועיל הן, למשל מצבים של שם עצם שמחובר לפועל שמחובר לעוד שם עצם, או למעשה כל שרשרת אינדיקטיבית של שלושה $POS\,Tag$ שסביר יותר או פחות שתהיה ביניהם תלות.

$$score_\theta (v_1, v_2, v_3, x_{1:n}) = 1 \iff v_1\, is\, NN \wedge v_2\, is\, V \wedge v_3\, is\, NN$$

דוגמה נוספת לפיצ'ר אחר למודל $grandparent\, model$ יכולה להיות אינדיקטור לכך שאם הצלע בין הסבא לאבא הייתה ימינה ובין האבא לבן שמאלה, או כל סדר שביניהם, למשל בדוגמה שכתבתי אם v_1 הסבא:

$$score_\theta (v_1, v_2, v_3, x_{1:n}) = 1 \iff (v_1, v_2)\ is\ right\ arc \wedge (v_2, v_3)\ is\ left\ arc$$

או כל קומבינציה אחרת של שמאל ימין (למשל ימין ימין).
בדומה אפשר לשלב עוד פיצ'רים עבור המרחקים של הצלעות שמרכיבות את השרשרת, כלומר שרשרת של קשתות ארוכות או קצרות, או קצרה ארוכה או ארוכה קצרה.
כאשר נקבע סף מסוים שצלעות מאורך גדול ממנו הן ארוכות ומתחת אליו קצרות.

שאלה 3

נתונים 3 מודלי שפה ומוגדר מודל שפה חדש:

$$P_M\left(x_n|x1,...,x_{n-1}\right)=\sum_{i=1}^{3}\lambda_i P_{Mi}\left(x_n|x1,...,x_{n-1}\right)$$

(א)

התנאי שצריך להתקיים (תנאי מספיק כפי שהובהר לי מהצוות) הוא שהלמדות יהיו צירוף קמור.

$$\lambda_1+\lambda_2+\lambda_3=1,\ and\forall i\in[3]\,\lambda_i\geq 0$$

ראשית נבחין כי בהנתן משפט $x=x_1,..,x_n$ נוכל להתנות במאורע $B=(X_1=x_1,..,X_{n-1}=x_{n-1})$.
כיוון ש$\forall i\in[3]\,P_{Mi}$ היא פונקציית הסתברות אז גם $P_{Mi_{|B}}\left(X_n=w\right)=P_{Mi}\left(w|B\right)$ היא פונקציית הסתברות (ראינו בקורס הסתברות) ובפרט גם נסכמות ל־1, על פני אוצר המילים $w\in V$. (*)
ולכן

$$\sum_{w\in V}P_M\left(w|B\right)\overset{\text{by def.}}{=}\sum_{w\in V}\left(\sum_{i=1}^{3}\lambda_i P_{Mi}\left(w|B\right)\right)=$$

כיוון שמספר הנסכמים סופי נוכל לשנות סדר סכימה (ובכל מקרה הם אי שליליים) ואז הלמדות לא תלויות בסכימה על פני אוצר המילים.

$$\sum_{i=1}^{3}\left(\sum_{w\in V}\lambda_i P_{Mi}\left(w|B\right)\right)=\sum_{i=1}^{3}\lambda_i\left(\sum_{w\in V}P_{Mi}\left(w|B\right)\right)\overset{(*)}{=}\lambda_1+\lambda_2+\lambda_3$$

לכן אם $\lambda_1+\lambda_2+\lambda_3=1$ זה תנאי מספיק לכך שP_M תיהיה פונקציית התפלגות מנורמלת (כי אם הם שונים היא לא מונרמלת ל1).
אם אחת הלמדות שלילית אפשר לבנות דוגמאות בה P_M תצא שלילית או לא תסכם ל־1.
לכן בתוספת התנאי $\forall i\in[3]\,\lambda_i\geq 0$ ידאג לכך שP_M תהא פונקציה אישלילית ולכן שניהם יחד מהווים תנאי מספיק לכך שהפונקציהP_M תגדיר התפלגות חוקית.

(ב)

נסמן C את הקורפוס שנלקח מאתרי החדשות.
נשים לב שעל מנת לקבוע את הלמדות ולהתאים את M הלמדות למשימה על אתרי החדשות, נרצה ללמוד צירוף קמור שימקסם את ההסתברות.

נתייחס לP_{M1}, P_{M2}, P_{M3} הנתונים ונמצא את ערכי הלמדות האופטימליות עבור המודל M.
כלומר זו בעיית מקסימיזציה עבור ערכי λ_i, נמצא את אלה שיפיקו הסתברות מקסימלית על C, הביטוי שנקבל הוא:

$$\underset{\lambda_1,\lambda_2,\lambda_3}{argmax}\left\{\prod_{x\in C} P_M(x)\right\} = \underset{\lambda_1,\lambda_2,\lambda_3}{argmax}\left\{\prod_{x\in C}\prod_{j=1}^{n(x)} P_M(x_j|x_1,..,x_{j-1})\right\} =$$

$n(x)$ - אורך המשפט x בC, והשתמשנו בכלל המכפלה/שרשרת להסתברות.
נוכל למקסם את $\log$ הביטוי (את הלוג ליקליהוד) כיוון שזו פונקציה עולה ואנחנו לא מחפשים את ערך המקסימום, אלא את ערכי $\lambda_1, \lambda_2, \lambda_3$ שימקסמו.
בעיית האופטימיזציה אותה צריך לפתור שקולה לזו של לוג הביטוי, ועל ידי תכונות של לוג של מכפלה נוכל לכתוב את הבעיה כך:

$$\underset{\lambda_1,\lambda_2,\lambda_3}{argmax}\left\{\log\left(\prod_{x\in C}\prod_{j=1}^{n(x)} P_M(x_j|x_1,..,x_{j-1})\right)\right\}$$

$$= \underset{\lambda_1,\lambda_2,\lambda_3}{argmax}\left\{\sum_{x\in C}\sum_{j=1}^{n(x)} \log\left(P_M(x_j|x_1,..,x_{j-1})\right)\right\} \overset{\text{by def.}}{=} \underset{\lambda_1,\lambda_2,\lambda_3}{argmax}\left\{\sum_{x\in C}\sum_{j=1}^{n(x)} \log\left(\sum_{i=1}^{3}\lambda_i P_{Mi}(x_j|x_1,..,x_{j-1})\right)\right\}$$

(ג)

בכדי לקחת בחשבון את המידע הנוסף נגדיר כמה שלשות שונות של למדות עבור המודל ונשתמש בהן לפי מקרים, כולם יהיו צירופים קמורים.
נקבל 3 שלשות $\lambda_i^{(j)}$ $i,j\in[3]$ שונות עבור רצפים שונים, פורמאלית נגדיר:
עבור $j=2$ נגדיר : $\lambda_1^{(2)}=0, \lambda_2^{(2)}=1, \lambda_3^{(2)}=0$.
עבור $j=1$ נגדיר: $\lambda_1^{(1)}=1, \lambda_2^{(1)}=0, \lambda_3^{(1)}=0$.
עבור $j=3$ נגדיר $\lambda_1^{(3)}, \lambda_2^{(3)}, \lambda_3^{(3)}$, השלשה שנלמדה מסעיף ב - כלומר הערכים הרגילים שנלמדו על קורפוס החדשות.
בעזרת 3 שלשות שכל שלשה מהווה צרוף קמור, אז נבחר את השלשה לפי:

$$\Pi(x_{n-1}) = \begin{cases} (1) & x_{n-1}\, starts\, with\, 'a' \\ (2) & x_{n-1}\, starts\, with\, 'b' \\ (3) & else \end{cases}$$

ואז

$$P_M(x_n|x_1,..,x_{n-1}) = \sum_{i=1}^{3}\lambda_i^{\Pi(x_{n-1})} P_{Mi}(x_n|x_1,..,x_{n-1})$$

והיא אכן פונקציית התפלגות כי כל השלשות שמוגדרות לפי המקרים מקיימות את התנאי שראינו בסעיף א.

שאלה 4

מדובר במודל $Trigram\, MEMM$. ונעבוד לפי ההנחות בשאלה $y_0 = y_{-1} = START$. נסמן Y קבוצת התיוג האפשריים.

$$P\left(y_1,..,y_n|x_1,..,x_n\right) = \prod_{i=1}^{n} P\left(y_i|x_1,..,x_n,y_{i-1},y_{i-2}\right)$$

זהו מודל דיסקרימינטיבי הסתברותי לוג ליניארי (המניח שכל אחד מהגורמים במכפלה הוא לוג ליניארי).
בשאלה זו הנחת המרקוביות מסדר 2.

(א)

נוסחא מפורשת:

$$P\left(y_i|x_1,..,x_n,y_{i-1},y_{i-2}\right) = \frac{\exp\left(w\cdot f(x_1,..,x_n,y_i,y_{i-1},y_{i-2},i)\right)}{\sum_{y'\in Y}\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)}$$

פונקציית הפיצ'רים יכולה להיות לממשיים אך בדרך כלל ראינו פיצ'רים בינאריים או אינדיקטורים
הארגומנטים שמקבלת פונקציית הפיצ'רים $f(x_1,..,x_n,y_{i-1},y_{i-2},i)$ הם :
ייצוג כל המשפט $x = x_1,..,x_n$ שעבורו מנסים לחזות תיוג (למשל POS), בתוספת 2 תגים קודמים הרלוונטיים כי המודל מסדר שני y_{i-1},y_{i-2}, ואת התג הנוכחי y_i שמנסים לחזות במשפט והמיקום שלו i (זה גם המיקום עבור המילה הנוכחית במשפט x_i).
נזכור כי במודל זה יש נירמול לוקאלי, המכנה בביטוי המפרש מעלה: $Z(y_{i-1},y_{i-2};w) = \sum_{y'\in Y}\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)$

(ב)

נזכור כי המודל בהנתן סט האימון המתואר (כאשר נסמן בקיצור $x^{(i)} = x_1^{(i)},..,x_{n(i)}^{(i)}$) הוא

$$P\left(y|x\right) = \prod_{i=1}^{N}\prod_{j=1}^{n(i)} P\left(y_j^{(i)}|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right) = \prod_{i=1}^{N}\prod_{j=1}^{n(i)} \frac{\exp\left(w\cdot f(x^{(i)},y_j^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)},j)\right)}{Z(y_{j-1}^{(i)},y_{j-2}^{(i)};w)}$$

הלוג ליקליהוד המותנה הוא לוג על הביטוי הנ"ל (מתכונות לוג של אקספוננט, לוג של מכפלה ומנה):

$$LL(w) = \sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[w\cdot f(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j) - \log(Z(y_{j-1}^{(i)},y_{j-2}^{(i)};w))\right]$$

נכתוב את הנוסחא עבור הגרדיינט שלו, ראשית הנגזרות הכיווניות (תכונות גזירה ליניארית+גזירה של לוג):

$$\frac{\partial LL(w)}{\partial w_k} = \sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[f_k(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j) - \frac{1}{Z(y_{j-1}^{(i)},y_{j-2}^{(i)};w)}\cdot\frac{\partial}{\partial w_k}Z(y_{j-1}^{(i)},y_{j-2}^{(i)};w)\right] =$$

$$\sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[f_k(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j)-\frac{\sum_{y'\in Y}\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)}{\sum_{y''\in Y}\exp\left(w\cdot f(x_1,..,x_n,y'',y_{i-1},y_{i-2},i)\right)}\cdot f_k(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)\right]$$

זו הנוסחא המפורשת, נשים לב שחלק זה המחוסר מאוד דומה להסתברות המפורשת ושלחלק סכום זה כמו לחלק כל אחד מהנסכמים ולכן נוכל לכתוב גם כך:

$$\sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[f_k(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j)-\sum_{y'\in Y}\frac{\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)}{\sum_{y''\in Y}\exp\left(w\cdot f(x_1,..,x_n,y'',y_{i-1},y_{i-2},i)\right)}\cdot f_k(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)\right]=$$

$$\sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[f_k(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j)-\sum_{y'\in Y}P\left(y'|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right)\cdot f_k(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)\right]$$

הגרדיינט יהיה כל הקורדינטות האלה בוקטור אחד בגודל $t=|w|$ וקטור המשקלים הנלמד.
כלומר $\nabla LL(w)=\left(\frac{\partial LL(w)}{\partial w_1},..,\frac{\partial LL(w)}{\partial w_t}\right)$. מפורשות:

$$\nabla LL(w)=\sum_{i=1}^{N}\sum_{j=1}^{n(i)}\left[f(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j)-\sum_{y'\in Y}P\left(y'|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right)\cdot f(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)\right]$$

(ג)

ניתן לחשב את הגרדיאנט בזמן פולינומי בגודל הקלט, כאשר גודל פונקציית הפיצ'רים כלומר הפלט שלה (השווה לוקטור המשקל w) הוא גם פולינומי בגודל הקלט נסמן אותו m.
נשים לב שיש לנוN משפטים באורך $n(i)$ לכל $i\in\{1,..,N\}$. כלומר סכום כל המילים המתוייגות (יתכנו חזרות) על פני המשפטים בקלט הוא $S=\sum_{i=1}^{N}n(i)$.
בחישוב הגרדיאנט אנחנו עוברים על כל S, כלומר N משפטים לכל אורכם בסיגמה הכפולה בגרדייאנט. S הוא ליניארי ובפרט פולינומי בגודל הקלט.
כעת נתבון בכל חישוב בתוך האיטרציות/סכימות החיצוניות כלומר הביטוי בתוך הסוגריים המרובעים.
בכל מחובר אנחנו מחשבים את הביטוי $f(x^{(i)},y_j^{(i)},y_{j-1}^{(i)}y_{j-2}^{(i)},j)$ בזמן חסום על ידי קבוע עבור הערכים הנתונים כל פעם (זו פונקציית פיצ'רים וזמן חישובה חסום ע"י החישוב הארוך ביותר של קלט -משפט, שלשת תיוגים ואינדקס).
מהביטוי הזה מחסרים את הביטוי $\sum_{y'\in Y}P\left(y'|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right)\cdot f(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)$.
כלומר מכפלה וסכימה של $|Y|$ גורמים כאשר את $f(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)$ גם כן אפשר לחשב בזמן קבוע, ו$|Y|$ גודל כל התיוגים קבוע בהנתן קבוצת תיוגים.

באשר ל$P\left(y'|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right)$ אנחנו יודעים כאשר מחשבים את y' כי כבר נתונים לנו 2 ערכי תיוגים קודמים $y_{j-1}^{(i)},y_{j-2}^{(i)}\in Y$.
מסעיף א יודע שהביטוי הוא $\frac{\exp(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i))}{\sum_{y''\in Y}\exp(w\cdot f(x_1,..,x_n,y'',y_{i-1},y_{i-2},i))}$ כאשר את החישוב של המכפלה והאקספוננט - $\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)$ - אפשר לעשות בזמן פולינומי כיוון שהנחנו שגודלם m פולינומי בגודל הקלט.
ואת חישוב גורם הנירמול במכנה גם אפשר לעשות בצורה יעילה, למעשה זה סכום של ביטויי $\exp\left(w\cdot f(x_1,..,x_n,y',y_{i-1},y_{i-2},i)\right)$ עבור כל $y'\in Y$ שחישבנו במונה.
לכן נוכל לשמור את הסכום ובסוף נעבור על כל הערכים ונוכל לנרמל אותם,
כך יהיה לנו את $\sum_{y'\in Y}P\left(y'|x^{(i)},y_{j-1}^{(i)},y_{j-2}^{(i)}\right)\cdot f(x^{(i)},y',y_{j-1}^{(i)}y_{j-2}^{(i)},j)$ ויחד עם הפירוט הקודם זה כל הדרוש עבור מחובר אחד,
לכן מחובר אחד בעלות פולינומית מתוך S מחוברים, שגם S פולינומי כפי שתארתי.
נקבל כי זמן הריצה סך הכל פולינומי.
(באופן שקול אפשר היה לחשב את המכפלה של הפיצ'ר וקטור עם w בעלות פולינומית עבור כל $y'\in Y$ לשמור את הסכום של כולם את הביטוי עבור כל תג יחיד לחלק בסכום ולהפעיל $softmax$).

שאלה 5

נעסוק בבעיה של סיווג טקסט ל3 קטגוריות $L = \{l_1, l_2, l_3\}$.
כאשר $x^{(i)}$ המסמך או המשפט הi (תלוי באיך מתקבל הקלט בדיוק), $y^{(i)}$ התיוג האמיתי עבורו.
L מרחב התגיות/קטגוריות אפשריות. כל תוצאה אפשרית לסיווג נסמן $y \in L$.

(א)

אנחנו רוצים מודל שידע לסווג מה הנושא מתוך L של טקסט כלשהו, בפיצ'רים בינאריים של $bag\, of\, words$ אנחנו מייצגים את המשפטים כוקטורים בינאריים.
כל מילה ייחודית, היא פיצ'ר או מימד בייצוג של $bag\, of\, words$, נכתוב BOW לקיצור.
לצורך כך מסדרים אוצר מילים נתון $V = \{w_i\}_{i=1}^{|V|}$ לפי סידור כלשהו (שצריך להשאר קבוע).
כך שאם במשפט מופיעה המילה w_i אז הייצוג של הוקטור המתאים לפי גישה זו יקבל 1 בקורדינטה המתאימה i. כך לגבי כל המילים (הייחודיות) מתוך V אוצר מילים נתון שהופיעו במשפט $x = x_1, .., x_n$ ו0 באחרות.
כלומר בהנתן מילה כלשהי לא משנה מה המקום בו היא מופיעה במשפט, תמיד היא תקבל על ידי הפיצ'ר הבינארי את אותה קורדינטה בוקטור. בסוף נקבל ייצוג של x על ידי וקטור בינארי שבו יש 1 בכל המילים במשפט $x_1, .., x_n$.
לדוגמה: אם $V = \{\text{Amos, the, snow, is, bored, in}\}$ אז עבור זו תיהיה דוגמה למשפט וייצוג לפי BOW יהיה $[1, 0, 0, 1, 1, 0] \leftarrow$ Amos is bored .
באשר לפונקציית הפיצ'רים ϕ היא צריכה לשקלל גם את התגיות האפשריות, כעת כל ייצוג בינארי של BOW, כלומר כל משפט יקבל יהיו 3 בלוקים המתאימים לL (במקרה הכללי $|L|$ בלוקים). בבלוק שמתאים לתג הנוכחי ש$\phi(x, y)$ מקבלת יופיע הייצוג הבינארי של המשפט ושאר הבלוקים יהיו בלוקים של אפסים.
הערך של ϕ הוא וקטור באורך המשפט כפול מספר הקטגוריות, במקרה של המודל בשאלה זו, בהנתן משפט $x = x_1, .., x_n$ אז הערך הוא וקטור באורך $3|V|$, כגודל אוצר המילים כפול מספר המחלקות בקבוצת הסיווג.
אצלנו עבור תג ומשפט נתונים ב$\phi(x, y)$ 2 בלוקים יהיו ריקים ו1 לפי הפי'צרים הבינאריים של המילים שמופיעות במשפט.
אם נחזור לדוגמה שנתתי ונניח כי 3 הסיווגים האפשריים הם לימוד, חופשה או פנאי
$L = \{\text{leisure, study, vacation}\}$ אז נקבל כי הייצוג על ידי ϕ הוא:

$$\phi\,(\text{Amos is bored, leisure}) = [1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\phi\,(\text{Amos is bored, study}) = [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]$$

$$\phi\,(\text{Amos is bored, vacation}) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0]$$

מודל לוג ליניארי משתמש בתוצאה שלϕ כפול וקטור משקלים w שהמודל לומד, מימדו של w צריך להיות כמו התוצאה שלϕ.
עבור x, y משפט ותיוג כלשהם נחשב את הניקוד/מישקול שלהם כך $score_w(x, y) = w \cdot \phi(x, y)$.
ניקח את המישקולים $score_w(x, y) = w \cdot \phi(x, y)$ ונהפוך אותם לפונקציית הסתברות בעזרת $softmax$.
כעת נוכל להגדיר את המודל ההסתברותי הלוג ליניארי:

$$P_w(y|x) = \frac{\exp\left(w \cdot \phi(x,y)\right)}{\sum\limits_{y' \in L} \exp\left(w \cdot \phi(x,y')\right)}$$

נבחין כי אכן מתקבלת פונקציית הסתברות אי שלילית הנסכמת ל1.
לא כתוב להרחיב על איך הלמידה פה גם נעשית, אמנם בדומה לשאלה קודמת גם פה הלמידה מבוססת גרדיינט למציאת w של הממקסם את לוג הנראות המירבית המותנית על ה$training\, data$. רק שפה הביטוי לגרדיינט מעט יותר פשוט.
לאחר שמצאנו את w וקטור המשקול באמצעות מידע האימון, ההיסק במודל מתבצע על ידי חיפוש הy שממקסם את ההסתברות המותנית
בהנתן המשפט x כלומר הצבה של כל $y \in L$ מציאת הערך שהמודל חוזה עבורו בהנתן המשפט, והחזרת הסיווג שקיבל ערך מקסימלי:

$$\underset{y \in L}{\operatorname{argmax}} \left\{P_w\left(y|x\right)\right\}$$

הערה: נבחין כי הנירמול הוא גורם קבוע במודל זה $\sum\limits_{y' \in L} \exp\left(w \cdot \phi(x,y')\right)$ ולכן אפשר בחיזוי לא להתייחס אליו ולמקסם ישר על המישקול $score_w(x,y)$.

(ב)

נעשה שימוש ב$word\, embeddings$ על מנת להתאים את ה$log\, linear\, classifer$ מסעיף קודם להתמודד טוב יותר עם המשימה בסעיף זה, כלומר להתמודד עם מילים בקיצורים, איות שאינו סטנדרטי או עם שגיאות הכתיב.
במקום להסתכל על וקטור כל המילים שהופיעו במסמך/משפט (כפי שעשינו בסעיף א עם $bag\, of\, words$),
נסתכל על ה$word\, embeddings$ של המילים במסמך.
כפי שראינו בהרצאות $word\, embeddings$ שימושי כאשר יש הרבה מילים שהותאמו להם השיכונים בשיטה זו. כך אפשר ללמוד על קשרים ויחסים בין מילים בעזרת היחסים בין הוקטורים של ה$word\, embeddings$ המתאימים להם.
שימוש זה יועיל למשימה שלנו כיוון שאין לנו $data$ מתוייג מהסוג החדש, אך יש לנו מודל מאומן עבור $data$ רגיל מהסוג הקודם, והרבה $data$ לא מתוייג מהסוג החדש.
על ידי ה$word\, embeddings$ נוכל למצוא מילים דומות ושונות באמצעות המרחק בין הוקטורים u,v שיותאמו להם על ידי ה$embedding$. למשל על ידי מדד ה$similarty$ עבור זוג הוקטורים.
עבור מילים דומות נצפה שהזווית בין הוקטורים תיהיה קרובה יותר ל־0 בייצוג האוקלידי (מה שמעיד על קרבה). כלומר קוסינוס הזווית יהיה קרוב יותר ל1. נוסחא מפורשת עבור ה$similarity$:

$$similarity\left(u,v\right) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum\limits_{i=1}^{n} u_i v_i}{\sqrt{\sum\limits_{i=1}^{n} u_i^2}\sqrt{\sum\limits_{i=1}^{n} v_i^2}}$$

נקבל ייצוג של המילים משוכנות במרחב אוקלידי ממימד $n \in \mathbb{N}$ ־ מימד של המרחב אליו ה$word\, embeddings$ שלנו משכן־ וייצוג יהיה יעיל אם מילים כמו u וyou (קיצור),
$entzyclopidya$ ו $Encyclopedia$ (שגיאת כתיב) יהיה קרובות במרחב זה.

אם נפעיל את ה$word\,embeddings$ עבור המילים הלא ידועות שמופיעות ב$data$ הלא מתוייג, ונעשה ממוצע על השיכונים, ייתכן שנוכל למצוא ייצוג וקטורי עבור המילה שישקף גם את המשמעות הסמנטית והקונטקסטים בהם היא מופיעה.
כך נוכל להשוות את השיכון הממוצע שמתקבל על מילה לא ידועה, למילים או ביטויים ב$training\,data$ שה$similarity$ בין הייצוג הוקטורי שלהם דומה לשיכון הממוצע של המילה הלא ידועה, ולבחור את המתאימה ביותר מבין המילים המתויגות.
בשיטה זו נוכל להחליף מילים לא ידועות במילים מה$training\,data$ שיהיו הכי קרובות מבחינת ה$similarity$, שיהוו תחליף לאלה הלא ידועות. כלומר מילה ידועה דומה מספיק בהקשר למילה הלא ידועה.
נצפה שעל התחליף המודל יוכל לתת פרדיקציה טובה יותר כי הוא הופיעו בדאטה המתוייג וכי מילים או ביטויים קצרים עם משמעות סמנטית דומה יופיעו בהקשרים או סביבות סמנטיות דומות, גם בדאטה המתויג וגם בחדש.
לדוגמה:
$you\,nailed\,it!$ מול $u\,nailed\,it!$ או אפילו $naaailed$ ייתכן כי נוכל להסיק את המשמעות מהקירבה היחסית בין הביטויים, במובן שזה ביטוי "מפרגן" ו"חיזוק חיובי", שנאמר בכל מקרה בהקשר של ביצוע עבודה בצורה טובה.
כלומר במקום ממש להגדיר משמעות של הטקסט והמילים המופיעות בו שאינן מוכרות, שלא נצפו גם ככה באימון (שגיאות כתיב וקיצורים לא סטנדרטיים), נסתכל על ה$word\,embeddings$ שלהן ושל השכנים שלהן. כך נוכל לספק חיזוי טוב יותר, כי פחות משנה איזו מילה הופיעה עם איזו מילה, ויותר משנה זה שההקשרים שהן מופיעות דומים, בפרט עבור מילים המכילות קיצור שגיאה או כלשהי.

(הערה) שיפור נוסף אפשרי הכולל עיבוד מקדים:
ייתכן שהטיפול הקודם יהיה יעיל במשמעות סמנטית כאשר יהיו מילים שנרצה להתאים בין קיצור לצורה לא מקוצרת כמו u וyou, ואולי גם עבור שגיאות כתיב.
אך עבור מילים עם שגיאות כתיב נוכל לעשות עוד עיבוד מקדים ולנסות לשפר את המצב אפילו יותר, כלומר לתקן את ה$misspelling$ קודם למילים הנכונות הדומות ביותר שנמצא בV אוצר המילים המקורי.
את השגיאות כתיב אפשר לתקן עם מתקן שגיאות כלשהו אם נתון ואפשר למשל לקחת בחשבון בשגיאות כתיב את מרחק העריכה־$edit\,distance$(כמה תווים צריך לשנות כדי להגיע ממחרוזת אחת לשניה).
שימוש בו ייתכן ויהיה יעיל עבור מילים בינוניות וארוכות שבהן יש שגיאת כתיב (שימוש בו אולי פחות מתאים בקיצורים כי המרחק עריכה בקיצורים לרוב לא רלוונטי למשל u קרוב יותר לv מאשר לyou, לכן זה יהיה עיבוד מקדים רק עבור מילים מאורך מסויים).
תהליך כזה יתאפשר אם יהיה לנו $character-level\,word\,embeddings$, נוכל עם המודל שמייצר את השיכונים ברמת האות למצוא ייצוג וקטורי עבור מילים בינוניות/ארוכות עם שגיאות כתיב ולחפש את השכן הקרוב ביותר בV אוצר המילים עם הדאטא המתוייג שקיבלנו, ולעבוד בהתאם למילה הידועה.
בדמיון מסויים ל$word\,embeddings$ הרגיל, נצפה שמילים עם מספיק אותיות דומות יקבלו ייצוג דומה, ואם למשל נכין ייצוג וקטורי מראש עבור כל אוצר המילים מה$data$ המתויג V, מעיין וריאציה של $'one\,hot'$.
כלומר כל מילה תקבל ייצוג וקטורי של השיכון ברמת האות, נוכל להשוות את הייצוגים עבור המילים הרלוונטיות, ולמצוא את השכן הכי קרוב מV בחיפוש יעיל לפי הייצוג הוקטורי שהוכן מראש (אולי עם חיפוש בינארי על הייצוג של V ממויין).
אם השכן הכי קרוב מספיק קרוב (למשל לפי סף פרמטר שנוכל לקבוע) אפשר להחליף את המילה ובכך לשפר את ביצועי המודל עם השגיאה לדומה לה שנראתה בדאטה המתויג.
לאחר שלב עיבוד מקדים זה נמשיך לשלב שתואר בתחילת הסעיף.