עיבוד שפה טבעית - תרגיל 1

ערן תורג׳מן (208484147) ודור דהוקי (315145490)

12 בנובמבר 2022

חלק I

חלק תיאורטי

1 שאלה 1

נסמן את אוצר המילים שלנו ב-V.
נתון כי לכל $w \in V$ מתקיים $P(STOP|w) > 0$ וגם שסכום הסתברויות המעברים בשפה שווה ל-1 כלומר $\sum_{u \in V} P(w|u) = 1$. לכן בהכרח ההסתברות לעבור ממילה w למילה u שאינה $STOP$ קטנה ממש מ-1, כלומר:

$$P(STOP|w) > 0 \Rightarrow \forall u \neq STOP : P(u|w) = 1 - P(STOP|w) < 1$$

מכאן שההסתברות לג׳רנט משפט שאינו מסתיים ב$STOP$, שהוא בהכרח משפט אינסופי לפי הגדרת המודל שווה לאפס:

$$\begin{aligned}
P(\text{infinite_sentences}) &= \prod_{i=1}^{\infty} P(w_i|w_{i-1}) \\
&= \prod_{i=1}^{\infty} (1 - P(STOP|w_{i-1})) \\
&= \lim_{n \to \infty} \prod_{i=1}^{n} (1 - P(STOP|w_{i-1})) \\
&= 0
\end{aligned}$$

על כן סכום ההסתברויות לג׳נרט את כל המשפטים הסופיים, שהוא המאורע המשלים, שווה ל1:

$$\begin{aligned}
P(\text{sentences finite}) &= \sum_{m=1}^{\infty} \prod_{i=1}^{m} P(w_i|w_{i-1}) \\
&= 1 - P(\text{infinite_sentences}) \\
&= 1 - 0 \\
&= 1
\end{aligned}$$

2 שאלה 2

2.1 סעיף א'

מודל שפה $uniform$י הוא מודל שמניח שהסתברות של מילה להופיע במשפט אינה תלויה במילים שקדמו לה, כלומר מתקיים:

$$P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i)$$

נשתמש במודל הנ"ל כדי לבנות מתקן איות הפועל באופן הבא בהינתן משפט לתיקון:

- מתקן האיות יעבור על כל מילה במשפט.
- ברגע שהגיע למילה $where$ או $were$, יחשב את ההסתברויות $P(where)$ ו-$P(were)$.
- יחליף את המילה הנוכחית במילה בעלת ההסתברות הגבוהה יותר, כלומר $w' = \arg\max_{w \in \{where, were\}} P(w)$. במידה וההסתברויות שוות ישאיר את המילה כמו שהיא.

נשים לב שבהינתן המשפט "He went where there where more opportunities" מתקן האיות שלנו יחזיר את התשובה הנכונה עבור המופע הראשון של $where$ (כלומר ישאיר את המילה $where$) במידה ו-$P(where) > P(were)$, ויחזיר את המילה הנכונה עבור המופע השני של $where$ (כלומר $were$) במידה ו-$P(where) < P(were)$. ועל כן מתקן האיות הנ"ל לא יחזיר אף פעם תשובה נכונה עבור שני המופעים.

2.2 סעיף ב'

מודל שפה $bigram$י הוא מודל שמניח שהסתברות של מילה להופיע במשפט תלויה אך ורק במילה שקדמה לה, כלומר מתקיים:

$$P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{i-1})$$

נשתמש במודל הנ"ל כדי לבנות מתקן איות הפועל באופן הבא בהינתן משפט לתיקון:

- מתקן האיות יעבור על כל מילה במשפט.
- ברגע שהגיע למילה $where$ או $were$, יחשב את ההסתברויות $P(where|u)$ ו-$P(were|u)$ כש-u היא המילה שקדמה למילה הנוכחית.
- יחליף את המילה הנוכחית במילה בעלת ההסתברות המתונית הגבוהה יותר, כלומר $w' = \arg\max_{w \in \{where, were\}} P(w|u)$. במידה וההסתברויות שוות, ישאיר את המילה הנוכחית כמו שהיא.

מתקן האיות הנ"ל יעבוד יותר טוב מזה שהצענו בסעיף א' מכיוון שהוא לוקח בחשבון את ההקשר של כל מילה, ולא יחליף מופעים של $where$ או $were$ רק על סמך כמות הפעמים שכל מילה מופיעה בקורפוס - אלא על סמך כמות הפעמים שכל מילה מופיעה בקורפוס כש-u היא המילה שקדמה לה.
לכן סביר להניח שבהינתן המשפט "He went where there where more opportunities" מתקן האיות שלנו יחזיר את התשובה הנכונה עבור שני המופעים של $where$.
משפט יכול לקבל הסתברות אפסית לפי המודל אם הוא מכיל זוג מילים עוקבות $u, were$ או $u, where$ וגם $u, where$ וגם $u, were$ לא מופיעות בקורפוס. במקרה זה יתקיים:

$$P(were|u) = P(where|u) = 0$$

ומתקן האיות לא יחזיר את התשובה הנכונה (באופן שאנחנו הגדרנו את המודל הוא ישאיר את המילה כמו שהיא).

3 שאלה 3

3.1 סעיף א'

מספר המילים בקורפוס הוא N. N_c הוא מספר המילים בקורפוס שמופיעות c פעמים עבור ערכי c בין 1 ל-c_{max}. עבור כל מילה בקבוצה N_c נתון כי התדירות למילה זו היא:

$$\frac{(c+1)\cdot N_{c+1}}{N_c \cdot N}$$

על כן מתקיים:

$$N = \sum_{c=1}^{c_{max}} c \cdot N_c$$

נראה כי סכום התדירויות על פני כל המילים בקורפוס שווה ל-$1 - p_{unseen}$:

$$\begin{aligned}
\sum_{c=1}^{c_{max}} N_c \cdot \frac{(c+1)\cdot N_{c+1}}{N_c \cdot N} &= \sum_{c=1}^{c_{max}} \frac{(c+1)\cdot N_{c+1}}{N} \\
&= \frac{1}{N} \sum_{c=1}^{c_{max}} (c+1)\cdot N_{c+1} \\
&= \frac{1}{N} \sum_{c=2}^{c_{max}} c \cdot N_c
\end{aligned}$$

בנוסף מתקיים:

$$\begin{aligned}
N &= \sum_{c=1}^{c_{max}} c \cdot N_c = 1 \cdot N_1 + \sum_{c=2}^{c_{max}} c \cdot N_c \\
N - N_1 &= \sum_{c=2}^{c_{max}} c \cdot N_c
\end{aligned}$$

ולכן נקבל:

$$\begin{aligned}
\sum_{c=1}^{c_{max}} N_c \cdot \frac{(c+1)\cdot N_{c+1}}{N_c \cdot N} &= \frac{1}{N} \sum_{c=2}^{c_{max}} c \cdot N_c \\
&= \frac{1}{N} \cdot (N - N_1) \\
&= 1 - \frac{N_1}{N} \\
&= 1 - p_{unseen}
\end{aligned}$$

כנדרש.

3.2 סעיף ב'

הנוסחה עבור החלקת $Add - One$ היא:

$$q_{add-1}(w) = \frac{c(w)+1}{\sum_{w'}(c(w')+1)} = \frac{c(w)+1}{N+|V|}$$

כש-N הוא כמות המילים בקורפוס ו-$|V|$ מספר המילים השונות בקורפוס.
התדירות של מילה w כלשהי היא: $\frac{c(w)}{N}$.
נחפש את הסף μ אשר עבורו ה-$Smoothed\ Estimate$ של המילה גדולה מה-MLE.

$$\begin{aligned}
\frac{c(w)+1}{N+|V|} &> \frac{c(w)}{N} \\
N \cdot (c(w)+1) &> c(w) \cdot (N+|V|) \\
N \cdot c(w) + N &> N \cdot c(w) + |V| \cdot c(w) \\
N &> |V| \cdot c(w) \\
\frac{c(w)}{N} &< \frac{1}{|V|}
\end{aligned}$$

ועל כן כאשר תדירות של מילה קטנה מ-$\mu = \frac{1}{|V|}$ אזי ה-$Smoothed\ Estimate$ של המילה גדולה מה-MLE. באופן דומה אם תדירות המילה גדולה מה-$\mu = \frac{1}{|V|}$ אזי ה-$Smoothed\ Estimate$ קטנה מה-MLE.

$$\begin{aligned}
\frac{c(w)+1}{N+|V|} &< \frac{c(w)}{N} \\
N \cdot (c(w)+1) &< c(w) \cdot (N+|V|) \\
N \cdot c(w) + N &< N \cdot c(w) + |V| \cdot c(w) \\
N &< |V| \cdot c(w) \\
\frac{c(w)}{N} &> \frac{1}{|V|}
\end{aligned}$$

3.3 סעיף ג'

נסתכל על המקרה בו הקורפוס מקיים:

$$\begin{aligned}
N_1 &= 100 \\
N_2 &= 1 \\
N_3 &= 1 \\
N_4 &= 2 \\
N &= 113
\end{aligned}$$

נבחן את ה-$smoothed\ estimate$ של מילה שמופיעה פעם אחת בלבד בקורפוס:

$$gt(w_1) = \frac{(1+1) \cdot N_{1+1}}{N_1 \cdot N} = \frac{2 \cdot 1}{100 \cdot 113} = \frac{1}{5650} < \frac{1}{113} = \frac{c(w_1)}{N} = MLE(w_1)$$

מכיוון שמתקיים $gt\left(w_1\right) < MLE\left(w_1\right)$ נוכל להסיק ש-μ שווה לכל היותר ל-$\frac{1}{113}$:

$$\mu < \frac{1}{113}$$

נבחן אצ המילה שמופיעה פעמיים, מתקיים:

$$MLE\left(w_2\right) = \frac{2}{113} > \frac{1}{113} > \mu$$

ולכן צריך להתקיים:

$$gt\left(w_2\right) < MLE\left(w_2\right)$$

אבל אם נחשב נקבל:

$$gt\left(w_2\right) = \frac{(2+1)\cdot N_{2+1}}{N_2 \cdot N} = \frac{3\cdot 1}{1\cdot 113} = \frac{3}{113}$$

כלומר:

$$gt\left(w_2\right) > MLE\left(w_2\right)$$

ולכן התכונה אינה מתקיים עבור $Good-Turing$.

4 שאלה 4

4.1 סעיף א'

ההנחה של מודל ה-$trigram$ היא שההסתברות של כל מילה להופיע במשפט תלויה רק בשתי המילים שקודמות לה, כלומר:

$$P\left(w_i|w_{i-1},\ldots,w_1\right) = P\left(w_i|w_{i-1},w_{i-2}\right)$$

ולכן ההסתברות של משפט כלשהו היא:

$$P\left(w_1,\ldots,w_n\right) = \prod_{i=1}^{n} P\left(w_i|w_{i-1},w_{i-2}\right)$$

4.2 סעיף ב'

ניתן דוגמה לשני משפטים אשר מודל ה-$trigram$ יצליח לחזות נכון את נטיית הפועל במשפט לפי שם העצם.

- הילדה הגבוהה אוכלת
- yell girls little The

4.3 סעיף ג'

ניתן דוגמה לשני משפטים אשר מודל ה-$trigram$ **לא** יצליח לחזות נכון את נטיית הפועל במשפט לפי שם העצם.

- הילדה עם השמלה הירוקה אכלה

- The dog with the big nose barks

בשביל המשפט הראשון נצטרך להשתמש במודל $5-gram$ על מנת לתפוס את ההטייה הנכונה של הפועל (נקבה) בעזרת שם העצם שנמצא ארבע מילים אחורה.
במשפט השני נצטרך מודל $6-gram$ כי צריך להסתכל 5 מילים אחורה.

5 שאלה 5

הכלבה של אבא אוכל בשר- המשפט אינו תקין תחבירית, בעוד שכל זוג כן תקין:

- הכלבה של (מוטי)
- (אח) של אבא
- אבא אוכל
- אוכל בשר

הכלבה של אבא שלי אוכל בשר- המשפט אינו תקין תחבירית בעוד שכל שלשה כן תקינה:

- הכלבה של אבא
- (אח) של אבא שלי
- אבא שלי אוכל
- (חבר) שלי אוכל בשר

הכלבה של אבא שלי אוכל בשר אדום- המשפט אינו תקין בעוד שכל רבעייה כן תקינה:

- הכלבה של אבא שלי
- (דוד) של אבא שלי אוכל
- אבא שלי אוכל בשר

שמנו לב שככל שעולים בסדר, קשה למצוא דוגמה למשפט שאינו תקין תחבירית שלא יזוהה על ידי המודל, על כן ניתן להסיק כי מודלים מרקוביים מתאימים להיות מודלי שפה וככל שסדר המודל גדול יותר הוא
הוא יכסה יותר ויותר משפטים בשפה (עד כדי משפטים חריגים וארוכים).
ולרוב יספיקו מודלי מרקוב מסדרים נמוכים יחסית (לרוב לא נצטרך להסתכל 20 מילים אחורה) על מנת לזהות בצורה נכונה את רוב המשפטים בטקסט נתון.

חלק II

חלק פרקטי

```
/usr/bin/python3 /mnt/c/Users/dor/Projects/NLP/nlp/ex1.py
Found cached dataset wikitext (/home/dor/.cache/huggingface/datasets/wikitext/wikitext-2-raw-v1/1
 .0/a241db52902eaf2c6aa732210bead40c090019a499ceb13bcbfa3f8ab646a126)
Loading models...
Q2:
The most probable word continuation for the sentence: 'I have a house in' is 'the'
Q3 A:
The probability of the sentence: Brad Pitt was born in Oklahoma is -inf.
The probability of the sentence: The actor was born in USA is -29.686567347483418.
Q3 B:
The perplexity is: inf
Q4:
The probability of the sentence: Brad Pitt was born in Oklahoma is -36.176302610738425.
The probability of the sentence: The actor was born in USA is -30.996327459140225.
The perplexity is: 269.81031430478953

Process finished with exit code 0
```