עיבוד שפה טבעית - תרגיל 1

שיר שבח 322407701, אליה חסון 208845032

17 בינואר 2024

חלק תיאורטי

1. א.

בתרגיל התבקשנו להוכיח שהסתברות של כל משפט סופי היא 1. נוכיח את המקרה המשלים, שהוא להוכיח שההסתברות למשפט אינסופי (משפט שאף פעם לא מגיע ל־STOP) הוא 0, וכיוון שמדובר במאורע משלים, נקבל כי הסתברות של כל משפט סופי היא אכן 1.

נגדיר מאורע $P(A)$ כאשר המאורע A הוא המאורע בו מייצר סדרה של מילים שלא מגיעות לSTOP. נשים לב שמדובר בשרשור אינסופי של מילים $w_1, w_2, ...$, שעל פי הנתון לכל $i, i+1$ $0 < P(w_1 \mid w_2) < 1$. לכן:

$$P(A) = \prod_{i=1}^{\infty} P(w_i \mid w_{i+1}) = 0$$

כי מכפלה אינסופית של איברים הקטנים מ־1 וגדולים מ־0 שואפת ל־0.

לכן קיבלנו שההסתברות למאורע בו המשפט הוא סופי (מגיע ל־STOP) הוא

$$1 - P(A) = 1 - 0 = 1$$

כנדרש.

ב.

לא. נראה דוגמא נגדית:

נגדיר את אוצר המילים הבא: $V = \{\text{START}, \text{hello}, \text{END}\}$. ונגדיר את מודל השפה הבא:

הסיכוי למילה hello בהינתן hello הוא:

$$P(w_n = \text{hello} \mid w_{n-1} = \text{hello}) = 1 - \frac{1}{2n^2}$$

הסיכוי לסוף המילה בהינתן hello הוא:

$$P\left(w_n = \text{END} \mid w_{n-1} = \text{hello}\right) = \frac{1}{2n^2}$$

לכן נקבל כי הסיכוי למשפט אינסופי (שלא מגיעים ל־END) הוא:

$$\begin{aligned}
P\left(\text{START,hello, hello.....}\right) &= \prod_{n=1}^{\infty} P\left(w_n = \text{hello} \mid w_{n-1} = \text{hello}\right) \\
&= \prod_{n=1}^{\infty} \left(1 - \frac{1}{2n^2}\right) \\
&= \lim_{i\to\infty} \prod_{n=1}^{i} \left(1 - \frac{1}{2n^2}\right) \\
&= \frac{\sqrt{2}\sin\left(\frac{\pi}{\sqrt{2}}\right)}{\pi} \approx 0.358
\end{aligned}$$

2. א.

במודל unigram ההסתברות למשפט נקבעת כך:

$$P\left(x_1 ... x_n\right) = \prod_{i=1}^{n} p\left(x_i\right)$$

כך ש־

$$p\left(x_i\right) = \frac{\text{count}\left(x_i\right)}{\sum\limits_{w_j \in V} \text{count}\left(w_j\right)}$$

לפי מודל unigram, עבור כל משפט שהמתקן איות שלנו יקבל, עבור המילים where או were האלגוריתם שלנו יבדוק איזה מילה מוסיפה למכפלת הסתברויות המילים סיכוי גבוה יותר.

נשים לב כי אם $P\left(\text{"where"}\right) > P\left(\text{"were"}\right)$ אזי:

$$\begin{aligned}
P\left(\text{"He went where"}\right) &= p\left(\text{"He"}\right) \cdot p\left(\text{"went"}\right) \cdot p\left(\text{"where"}\right) \\
&> p\left(\text{"He"}\right) \cdot p\left(\text{"went"}\right) \cdot p\left(\text{"were"}\right) \\
&= P\left(\text{"He went were"}\right)
\end{aligned}$$

לכן התנאי שנצטרך כדי שהאלגוריתם יצדוק בחלק הראשון של המשפט הוא שההסתברות של "where" תהיה יותר גדולה משל "were", כלומר "where" צריך להופיע יותר פעמים ב־corpus מאשר "were".

באופן דומה, אם $P(\text{"were"}) > P(\text{"where"})$ אזי

$$\begin{aligned} P(\text{"there were more opportunities"}) &= p(\text{"there"}) \cdot p(\text{"were"}) \cdot p(\text{"more"}) \cdot p(\text{"opportunities"}) \\ &> p(\text{"there"}) \cdot p(\text{"were"}) \cdot p(\text{"more"}) \cdot p(\text{"opportunities"}) \\ &= P(\text{"there where more opportunities"}) \end{aligned}$$

התנאי שנצטרך כדי שהאלגוריתם יצדוק בחלק השני של המשפט הוא שההסתברות של "were" תהיה יותר גדולה משל "where", כלומר "were" צריך להופיע יותר פעמים ב־corpus מאשר "where".

נשים לב שהמקרה בו האלגוריתם יצדוק בכל המשפט לא ייתכן. נניח בלי הגבלת הכללית $P(\text{"where"}) > P(\text{"were"})$ אזי:

$$\begin{aligned} &= P(\text{"He went where there were more opportunities"}) \\ &= p(\text{"He"}) \cdot p(\text{"went"}) \cdot p(\text{"where"}) \cdot p(\text{"there"}) \cdot p(\text{"were"}) \cdot p(\text{"more"}) \cdot p(\text{"opportunities"}) \\ &< p(\text{"He"}) \cdot p(\text{"went"}) \cdot p(\text{"where"}) \cdot p(\text{"there"}) \cdot p(\text{"where"}) \cdot p(\text{"more"}) \cdot p(\text{"opportunities"}) \\ &= P(\text{"He went where there where more opportunities"}) \end{aligned}$$

* נשים לב שבטוח משפט אחד ייבחר כי אנו מניחים שלכל המילים הסתברות יותר גדולה מ־0, אחרת, אם אחת המילים במשפט הנ"ל היתה מקבלת הסתברות 0, היא היתה מאפסת את הסיכוי לכל המשפט וכך האלגוריתם לא היה נותן עדיפות ל־where מאשר were.

במקרה בו $P(\text{"where"}) = P(\text{"were"})$, נניח כי האלגוריתם שלנו בוחר את המילה were לפני where, ולכן המודל יחזיר את המשפט "He went were there were more opportunities" כי הוא בחר את were פעמיים.

קיבלנו לפי מודל unigram שהסיכוי למשפט התקין נמוך יותר ולא ייבחר.

ב.

המודל bigram מוגדר באופן הבא, הסיכוי למשפט הוא:

$$P(x_1 ... x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i-1})$$

לכן האלגוריתם שלנו, למילים where או were יחליט איזו מילה תקנית יותר לפי המילה הקודמת שמופיעה במשפט. אם $P(\text{where} \mid w) > P(\text{were} \mid w)$ אז where תבחר, אחרת were תבחר.

נשים לב שאם אנו מניחים שבcorpus יש מספיק דוגמאות נכונות דקדוקית,

$$P(\text{where} \mid \text{went}) > P(\text{were} \mid \text{went})$$

משום שהצירוף went where הינו צירוף תקין באנגלית.

וגם

$$P\left(\text{were} \mid \text{there}\right) > P\left(\text{where} \mid \text{there}\right)$$

משום שהצירוף there were גם יותר נכון תקנית מהצירוף השני.

לכן לפי מודל bigram, כאשר נפגוש במילה where או were, האלגוריתם יחליט בהתבסס על המילה הקודמת במשפט את המילה התקנית יותר, כי סיכויה יותר גבוה לפי ה־corpus.

אם אנו פועלים לפי המודל הזה, יכול להיות סיכוי של משפט לקבל הסתברות 0. זאת משום שהמשפט יכול לכלול צירופים לא נכונים תקנית, ואז הם לא יופיעו ב־corpus, או שתהיה דוגמא מאוד נדירה של צירוף תקני, שלא מופיע ב־corpus. מקרה זה מעט בעייתי כי גם המשפט נכון לא בהכרח נקבל את התיקון הנכון.

3. א.

נסכום את כל האומדני תדירויות Good-Turing על כל סוגי המילים בקורפוס האימון:

$$\begin{aligned}
\sum_{c=1}^{c_{max}} \frac{(c+1)\,N_{c+1}}{N_c \cdot N} \cdot N_c &= \sum_{c=1}^{c_{max}} \frac{(c+1)\,N_{c+1}}{N} \\
&= \frac{2 \cdot N_2 + 3 \cdot N_3 + ... + c_{max} \cdot N_{c_{max}} + (c_{max}+1) \cdot N_{c_{max}+1}}{N} \\
&= \frac{N - N_1}{N} \\
&= 1 - \frac{N_1}{N} \\
&= 1 - P_{unseen}
\end{aligned}$$

ב.

לכל התפלגות $q\left(w\right)$ ומילה w התפלגות החלקה Add-One מוגדרת באופן הבא:

$$q_{add-one}\left(w\right) = \frac{c+1}{N+|V|}$$

MLE מוגדר באופן הבא:

$$q_{ML}\left(w\right) = \frac{c}{\sum\limits_{w_i \in V} count\left(w_i\right)} = \frac{c}{N}$$

נרצה למצוא את μ. נרצה למצוא את ה־c שבו $q_{add-one}(w)$ יותר קטן מ־ $q_{ML}(w)$:

$$\begin{aligned}
q_{add-one}(w) &< q_{ML}(w) \\
\frac{c+1}{N+|V|} &< \frac{c}{N} \\
N(c+1) &< c(N+|V|) \\
Nc+N &< cN+c|V| \\
N &< c|V| \\
\frac{N}{|V|} &< c
\end{aligned}$$

מצאנו שכל המילים שתדירותן c גדול מ־ $\frac{N}{|V|}$ מתקיים ש־ $q_{add-one}(w) < q_{ML}(w)$.
עבור אי השיוויון השני נקבל בצורה סימטרית שעבור $\frac{N}{|V|} > c$ מתקיים ש־ $q_{add-one}(w) > q_{ML}(w)$.
לכן עבור ערך סף $\mu = \frac{N}{|V|}$, מצאנו את הדרוש.

ג.

נראה שלא בהכרח נוכל למצוא ערך סף ל־c שעבורו $q_{good-turing}(w) > q_{ML}(w)$.
נשים לב שאם $|V| = N = 1$, כלומר יש לנו בקורפוס מילה אחת w. אזי מתקיים:

$$q_{good-turing}(w) = \frac{(c+1)N_{c+1}}{N_c \cdot N} = \frac{(1+1)N_2}{N_1 \cdot N} = \frac{2 \cdot 0}{1 \cdot 1} = 0$$

$N_2 = 0$ משום ש־ $N_1 = N_{c_{max}}$ ולפי ההגדרה של good turing smoothing לכל $c > c_{max}$, $N_c = 0$.
ומשום ש־ $q_{ML}(w) = \frac{c}{N} = \frac{1}{1} = 1$, אנו מקבלים כי:

$$0 = q_{good-turing}(w) > q_{ML}(w) = 1$$

סתירה.

4. א.

מודל ה־trigram הוא שההסתברות למשפט הוא:

$$P(x_1 ... x_n) = \prod_{i=1}^{n} p(x_i \mid x_{i-1}, x_{i-2})$$

אנו מניחים תלות רק בין כל שלושת מילים צמודות במשפט.

ב.

עבור המשפט באנגלית "She eats pizza", לפי מודל trigram, המודל בדיוק יתפוס לנו את שלושת המילים במשפט, ויחשב את ההסתברות עבור כל השלשה עם s ובלי s ב־eat, ויגיע למסקנה שההסתברות היותר גבוה היא עם s.

עבור המשפט בעברית "היא אוכלת פיצה", לפי מודל trigram, המודל בדיוק יתפוס לנו את שלושת המילים במשפט, ויחשב את ההסתברות עבור כל השלשה האם הפועל יהיה "אוכלת" או צורות פועל אחרות: אוכל, אוכלים, אוכלות... ויגיע למסקנה שההסתברות היותר גבוה היא עם הפועל "אוכלת".

ג.

עבור המשפט באנגלית "The girl with the red shirt eats pizza", המודל שלנו יגיע למקרה הבא:

$$P\left(\text{"eat"} \mid \text{"red shirt"}\right) \stackrel{?}{=} P\left(\text{"eats"} \mid \text{"red shirt"}\right)$$

משום ששני המקרים הגיוניים באנגלית.

מודל ה־ 6-gram נקבל בדיוק את ההסתברות הבאה:

$$P\left(\text{"eats"} \mid \text{"girl with the red shirt"}\right) > P\left(\text{"eat"} \mid \text{"girl with the red shirt"}\right)$$

ועל כן המודל ייבחר במילה הנכונה.

עבור המשפט בעברית "הילדה עם החולצה האדומה אוכלת פיצה", המודל שלנו יגיע למקרה הבא: ההסתברות של "אוכלת" בהינתן "החולצה האדומה" כנראה שווה להסתברות של "אוכל" בהינתן "החולצה האדומה" משום ששני המקרים הגיוניים בעברית.

מודל ה־ 5-gram יגלה שההסתברות שהמילה "אוכלת" תופיע בהינתן "הילדה עם החולצה האדומה" גדולה יותר מהמילה "אוכל"/"אוכלים"/"אוכלות" ועל כן המודל ייבחר במילה הנכונה.

.5

נראה דוגמאות למשפטים לא תקינים דקדוקית בעברית, אולם כל צמד של n מילים צמודות במשפט תקין דקדוקית.

2-gram: "הילדה בגן אוכל פסטה כל יום".

3-gram: "הילדה בגן המשחקים משחק במגלשה כל שבת".

4-gram: "הילד עם הכובע הצהוב משחקת תופסת"

משום שצריך להוסיף אחרי שם העצם יותר תיאורים לפני הפועל כדי לגרום שיהיה מספיק מרחק ביניהם. וגם ככל שנגדיל את n הוא יתפוס את כל המקרים n-1-gram, n-2-gram... המשפטים שהמודל לא יתפוס יהיו יותר נדירים, והמודל n-gram יהיה יותר חזק.

חלק פרקטי

```
Q2.
The word with the highest probability to come next is: the
Q3.a.
The probability of the first sentence by Bigram Model is: -inf
The probability of the second sentence by Bigram Model is: -29.66684697358842
Q3.b.
The perplexity of the sentences by the Bigram Model is: inf
Q4.
The probability of the first sentence by LIS Model is: -36.19159630034921
The probability of the second sentence by LIS Model is: -30.99285170428172
The perplexity of the sentences by the LIS Model is: 270.0761619145355
```