מגישים: איתי צאצי 208489732
עומר מושלין 208271197
איתי כונא 316385962

Natural Language Processing – Exercise 1

Due: Sunday 21.1.2024 23:59

1 Theoretical (60 points)

1. (10 pts)

   (a) Given a bigram language model for sentences of the form START $w_1\ w_2\ w_3\ \cdots\ w_n$ STOP (where w_i for $1 \le i \le n$ is a word), show that if the transition probabilities are well-defined (i.e., sum up to 1) and each word has some non-zero probability to transition to STOP (i.e., $\forall w, p(\text{STOP}|w) > 0$), then the sum of the probabilities over all finite sequences is 1.

   Hint: prove that the complement probability (i.e., the probability to never generate STOP, which is the same as the sum of all the sequences that don't have STOP) is 0.

   (b) Is the property you proved in (a) true for non-Markovian models (i.e., models that predict a word based on unlimited past words)? If yes, prove it. If not, give a contradicting example (i.e., describe a model over some language in which the conditions hold but the probability of some infinite sequence is not 0).

2. (15 pts) We want to build a spelling corrector, focusing on the distinction between "where" and "were". Given a sentence as input, the corrector should predict the true spelling for each instance of "where" or "were" and correct the spelling in the case of a mistake.

   For example, given the sentence "He went where there where more opportunities", the corrector should predict "where" for the first instance and "were" for the second one. It should also correct the word in the second case.

   Suppose we use a language model for this task. Given a language model $p(w_1, w_2, \cdots, w_n)$ where n is the length of the sentence, the corrector returns the spelling that gives the highest probability.

   In our example, the spelling corrector will output "were" for the second instance if:

   p(He went where there **were** more opportunities) $>$ p(He went where there **where** more opportunities)

   (a) Describe formally a unigram language model for the spelling corrector. Assume that the probability of a word is given by its proportion in the corpus (the training set) and that the number of instances in the corpus of each word in the vocabulary is strictly bigger than 0. Given the sentence "He went where there where more opportunities", under which conditions will the spelling corrector give the right answer for the first instance of "where"? for the second instance of "where"? for both instances?

   (b) Describe formally a bigram language model for the spelling corrector. Assume again that we estimate the parameters of the model using relative frequency and that the number of instances in the corpus of each word in the vocabulary is strictly bigger than 0. Why might this model be better than the model in (a)? Can a sentence in this model get a zero-probability? Would it be a problem for the model?

3. (15 pts) Consider the advanced smoothing method called Good-Turing smoothing. Let N_c be the number of word types (unique words) which appeared exactly c times in the training corpus (e.g., N_1 is the number of unique words that appeared one time in the training corpus). N denotes the total number of word instances in the training corpus. An estimate of the total probability of all unseen words (i.e., words that do not appear in the training corpus) is given by $p_{unseen} = \frac{N_1}{N}$.

   The smoothed Good-Turing estimate of a frequency of a word that appears c times in the training corpus is $\frac{(c+1)N_{c+1}}{N_c \cdot N}$.
   Note: Assume that $N_c > 0$ for all values of c up to a certain maximum value c_{max} and $N_c = 0$ for all $c > c_{max}$.

   (a) Show that the sum of smoothed Good-Turing frequency estimates over all word types in the training corpus is $1 - p_{unseen}$

   (b) Write down the equation for the smoothed Add-One estimate of a frequency of a word that appears c times in the training corpus. Show that there is a threshold μ, such that for all words of frequency less than μ, their smoothed estimate is higher than the MLE, and for all words of frequency more than μ, their smoothed estimate is lower than the MLE.

   (c) Show that the property in (b) does not necessarily hold for the smoothed Good-Turing estimate.

4. (15 pts)

   (a) Write down the equation for a trigram language model (without detailing the probability estimations). Which (conditional) independence assumption is made in the model?

   (b) Give an example of an English sentence and a Hebrew sentence where the phenomenon of verb-subject agreement (see below) is captured by the model in (a). That is, give an example where the model in (a) is likely to predict the correct inflection of the verb, given the subject.

   (c) Give an example of an English sentence and a Hebrew sentence where subject-verb agreement is not captured. Which n (for an n-gram model) is necessary for capturing this phenomenon in your example?

   Note: Subject-verb agreement is the correspondence between the morphological inflection of the verb and the type of the subject. For instance, in English where the subject is singular, verbs in present tense end with an 's', while the base form is used for plural subjects. (e.g., "a dog bark**s**", "dogs bark")

5. (5 pts) Give an example of a sentence (in Hebrew or in English), where each consecutive pair of words is grammatically valid, but the sentence is not grammatically valid. Do the same with consecutive triplets and consecutive 4-tuples. You will note that doing this exercises for 4-tuples is considerably more difficult than for pairs. What does that indicate, in terms of the suitability of Markov models (of various orders) to be language models?
   Note: A sequence of words is said to be *grammatically valid* in a language, if there is a grammatical sentence in that language that contains the sequence as a sub-string.

2 Practical (40 points)

In this part, you will implement simple unigram and bigram language models and an interpolation between them. You will train the models with text from Wikitext.

Required Packages: To carry out the exercise, you will need to install the *SpaCy* package for language processing and Huggingface's *datasets* package for acquiring the data.

- You can find more details on the *spacy* website, where the two most important data structures are *doc* (for a document) and *token*:

  `https://spacy.io/api/doc`
  `https://spacy.io/api/token`

- You will also need the default English model of SpaCy called *"en_core_web_sm"*. See instructions here:

  `https://spacy.io/usage/models`

- For additional information on the datasets package see -

  `https://huggingface.co/docs/datasets/index`

- You will use the wikitext-2-raw-v1 version and use the train set only (see code example).

- Code example:

```
import spacy
from datasets import load_dataset

nlp = spacy.load("en_core_web_sm")
text = load_dataset('wikitext', 'wikitext-2-raw-v1', split="train")
doc = nlp(text[0]['text'])
```

Principles:

- You will use lemmas instead of exact words. See the *.lemma_* field in the *spacy token* documentation.

- You will use only words and not punctuation or numbers. This should be done with the *.is_alpha* field.

- We regard each line in the data as a separate document. The n-grams can continue across sentences (ignoring the period) but not across documents.

- For the bigram model, you will add a START token at the beginning of each document. This token will be used for the probability of the first actual word and not the START token itself. For both models, you will not add STOP tokens.

- These principles hold for both training and inference. You will regard the training text as a sequence of lemmas without punctuation, and the model will assign probabilities to sentences as if they were a sequence of lemmas. So, for example, the sentences "I eat lunch" and "I ate lunch" should have the same probability.

- We recommend computing all probabilities in log space. This is a good practice when working with very small probabilities to avoid numerical instability. You can report results either as the actual probability or the log probability.

Tasks:

1. Train unigram and bigram language models using their maximum likelihood estimators based on the above-mentioned training data.

2. Using the bigram model, continue the following sentence with the (one) most probable word predicted by the model: "I have a house in ...".

3. Using the bigram model:
   (a) compute the probability of the following two sentences (for each sentence separately).
   (b) compute the perplexity of **both** the following two sentences (treating them as a single test set with 2 sentences).
   - " Brad Pitt was born in Oklahoma "
   - " The actor was born in USA "
4. Now estimate a new model using linear interpolation smoothing between the bigram model and unigram model with $\lambda_{bigram} = 2/3$ and $\lambda_{unigram} = 1/3$ and using the same training data. Given this new model, compute the probability and the perplexity of the two sentences above.

Submission: Submit a .pdf file with your answers to the questions, as well as a .py or .ipynb file with your code.

1 Theoretical (60 points)

1. (10 pts)

   (a) Given a bigram language model for sentences of the form START $w_1\ w_2\ w_3\ \cdots\ w_n$ STOP (where w_i for $1 \le i \le n$ is a word), show that if the transition probabilities are well-defined (i.e., sum up to 1) and each word has some non-zero probability to transition to STOP (i.e., $\forall w, p(\text{STOP}|w) > 0$), then the sum of the probabilities over all finite sequences is 1.

   Hint: prove that the complement probability (i.e., the probability to never generate STOP, which is the same as the sum of all the sequences that don't have STOP) is 0.

צ״ל: ההסתברות של המאורע המשלים - ההסתברות לרצף מילים בו לא מופיע "STOP" היא אפס:

נראה כי ההסתברות ליצירת רצף סופי של מילים היא: 1

ע״י כך שנראה כי ההסתברות למאורע המשלים הוא אפס, כלומר והסתברות ליצור רצף אינסופי (רצף שהמילה STOP אינה מופיעה בו)

על כן, נניח כי ניתן לייצר רצף אינסופי של מילים ונסמני START, w_1, w_2.....

מאחר ורצף מילים זה הוא אינסופי וקיים מס׳ סופי של מילים במאגר המילים (נסמני V) אזי קיימת מילה w_i כך ש: $V \ni$ שמופיעה אינסוף פעמים ברצף.

עפ״י הנתון מתקיים $\forall w\ P(Stop|w) > 0$ כלומר בפרט קיים $0 < \varepsilon \le 1$ כך ש- $P(Stop|w_i) = \varepsilon$ **

מאחר ופונקציית ההסתברות שלנו מוגדרת היטב מתקיים כי:

$$1 = \mathbb{P}(\{START, w_1 \ldots w_n, STOP\} | w_i)$$ → סכימת כל האפשרויות לאחר w_i

$$= \mathbb{P}(\{START, w_1 \ldots\ldots w_n\} | w_i) + \mathbb{P}(STOP | w_i)$$

**

$$= \mathbb{P}(\{START, w_1 \ldots\ldots w_n\} | w_i) + \varepsilon$$

העברת אגפים $\Downarrow$

$$\mathbb{P}(\{START, w_1 \ldots\ldots w_n\} | w_i) = 1 - \varepsilon < 1$$

כלומר, ההסתברות לבחור מילה שאינה STOP בכל פעם קטנה ממש מ-1

כעת, נכפול הסתברויות באות כדי למצוא את ההסתברות לרצף האינסופי

$$p(START, w_1, w_2 \ldots) = \prod_{j=1}^{\infty} p(w_j | w_{j-1}) = \prod_{\substack{j=1 \\ j-1 \neq i}}^{\infty} p(w_j | w_{j-1}) \cdot \prod_{\substack{k=1 \\ w' \in V}}^{\infty} p(w', w_i)$$

נסמן $w_0 = start$

w_i מופיע אינסוף פעמים

קיים מופע של w' לפני w_i

$$\leq \prod_{\substack{j=1 \\ j-1 \neq i}}^{\infty} p(w_j | w_{j-1}) \cdot \prod_{k=1}^{\infty} p(\{START, w_1 \ldots w_n\} | w_i)$$

מכפלה גדולה יותר

$$= \prod_{\substack{j=1 \\ j-1 \neq i}}^{\infty} p(w_j | w_{j-1}) \underbrace{\prod_{k=1}^{\infty} (1-\varepsilon)}_{\text{שואף}} \overset{\infty}{=} c \prod_{\substack{j=1 \\ j-1 \neq i}}^{\infty} p(w_j | w_{j-1}) \cdot 0 = 0$$

ל-0 - מכפלה

אינסופית של איברים

קטנים מ-0

לסיכום הראנו כי הרצף הנ"ל הסתברותו שווה אפס ומכאן הטענה ∎

(b) Is the property you proved in (a) true for non-Markovian models (i.e., models that predict a word based on unlimited past words)? If yes, prove it. If not, give a contradicting example (i.e., describe a model over some language in which the conditions hold but the probability of some infinite sequence is not 0).

הרחבה: מודלים לא מרקוביים אלה מודלים שמשתמשים בהקשר עבר בלתי מוגבל (כלומר כמות מילים או מידע בלתי מוגבל) כדי לנסות ולחזות את המילה הבאה, במקום להסתמך על <u>מספר קבוע</u> של מילים קודמות.

נראה דוגמה נגדית: נראה דוגמה של מודל שפה שבו מתקיימת התכונה מסעיף a אך ההסתברות לרצף אינסופי משהו לא 0:

נגדיר מודל שפה לא-מרקובי באופן הבא:

$$V = \{START, STOP, Hi\}$$

$$\forall n \in \mathbb{N} \qquad P(STOP \mid Hi^{n-1}) = \frac{1}{2n^3}$$

$$P(Hi \mid Hi^{n-1}) = 1 - \frac{1}{2n^3}$$

כאשר w^{n-1} מבטא חזרה של המילה $n-1$ פעמים. מתקיים שההסתברות למילה אינסופית היא:

$$P(Hi^{\infty}) = \prod_{i=1}^{\infty} P(Hi \mid Hi^{i-1}) =$$

$$= \prod_{i=1}^{\infty} \left(1 - \frac{1}{2n^3}\right) \approx 0.548 > 0$$

כלומר ההסתברות ליצירת רצף אינסופי גדולה מ-0.

2. (15 pts) We want to build a spelling corrector, focusing on the distinction between "where" and "were". Given a sentence as input, the corrector should predict the true spelling for each instance of "where" or "were" and correct the spelling in the case of a mistake.

For example, given the sentence "He went where there where more opportunities", the corrector should predict "where" for the first instance and "were" for the second one. It should also correct the word in the second case.

Suppose we use a language model for this task. Given a language model $p(w_1, w_2, \cdots, w_n)$ where n is the length of the sentence, the corrector returns the spelling that gives the highest probability.

In our example, the spelling corrector will output "were" for the second instance if:

$$p(\text{He went where there } \mathbf{were} \text{ more opportunities}) > p(\text{He went where there } \mathbf{where} \text{ more opportunities})$$

(a) Describe formally a unigram language model for the spelling corrector. Assume that the probability of a word is given by its proportion in the corpus (the training set) and that the number of instances in the corpus of each word in the vocabulary is strictly bigger than 0. Given the sentence "He went where there where more opportunities", under which conditions will the spelling corrector give the right answer for the first instance of "where"? for the second instance of "where"? for both instances?

נתאר מודל זה בצורה פורמלית.

נסמן ב- N את גודל הקורפוס, עבור משפט $w_1, \dots, w_n$ מתקיים

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

עבור מילה w_i $i \in [N]$, נסמן ב- $c(w_i)$ את מספר ההופעות של מילה זו בקורפוס.

לפי הגדרה,

$$p(w_i) = \frac{c(w_i)}{N}$$

נסמן - $x = c(\text{where})$

$$y = c(\text{were})$$

מאחר ונבחר משפט לפי ההסתברות הגבוהה ביותר, נסיק כי, נבחר בתשובה הנכונה (where) עבור ההופעה הראשונה אם $x > y$ או בהסתברות $\frac{1}{2}$ אם $y = x$. באופן דומה, עבור המופע השני, נבחר בתשובה הנכונה (were) אם $y > x$ או בהסתברות $\frac{1}{2}$ אם $x = y$.

עבור שתי התשובות הנכונות נצטרך שיתקיים $y=x$, ונבחר בשגיאה הנכונה בהסתברות $\frac{1}{4}$ (יש 4 אופציות סה"כ).

<u>הערה</u> - תשובה זו נכונה כי בהינתן שוויון ההסתברויות נבחר באופן רנדומלי.

(b) Describe formally a bigram language model for the spelling corrector. Assume again that we estimate the parameters of the model using relative frequency and that the number of instances in the corpus of each word in the vocabulary is strictly bigger than 0. Why might this model be better than the model in (a)? Can a sentence in this model get a zero-probability? Would it be a problem for the model?

נתאר מודל זה בצורה פורמלית.
נסמן ב- N את גודל הקורפוס, עבור משפט $w_1,\dots,w_n$ מתקיים:

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

עבור מילה w_i, $i\in[N]$, נסמן ב- $C(w_i)$ את מספר ההופעות של מילה זו בקורפוס, ועבור צמד מילים w_i,w_j, $i,j\in[N]$ נסמן ב- $C(w_i,w_j)$ את מספר ההופעות של צמד מילים אלה ביחד.

לפי הגדרה: $\forall i,j \in [N] \quad p(w_i | w_j) = \frac{c(w_i,w_j)}{c(w_j)}$

מודל זה אמור לחזות טוב יותר ממודל היוניגרם שכן אנו מוסיפים לו context עבור המילה הקודמת. במילים אחרות נצפה שקורפוס המשפטים יכיל משפטים תקינים, לכן המודל יתן ערך גבוה ל- (w_i,w_j) כאשר $w_j \in \{where, were\}$ ו- w_i הינה מילה קודמת תקינה.

משפט יכול לקבל הסתברות 0, שכן הוא עלול להכיל רצף לא תקין אשר לא נצפה בקורפוס האימון, כמו "went were", עובדה זו יכולה להוות יתרון וחיסרון. יתרון כי כך נשלול ביטויים לא נכונים דקדוקית וחסרון כי אם לא ראינו צירוף תקין מסוים, ניתן לו הסתברות 0 למרות שבמודל הקודם, יש סיכוי שיקבל הסתברות גבוהה יותר.

3. (15 pts) Consider the advanced smoothing method called Good-Turing smoothing. Let N_c be the number of word types (unique words) which appeared exactly c times in the training corpus (e.g., N_1 is the number of unique words that appeared one time in the training corpus). N denotes the total number of word instances in the training corpus. An estimate of the total probability of all unseen words (i.e., words that do not appear in the training corpus) is given by $p_{unseen} = \dfrac{N_1}{N}$.

The smoothed Good-Turing estimate of a frequency of a word that appears c times in the training corpus is $\dfrac{(c+1)N_{c+1}}{N_c \cdot N}$.

Note: Assume that $N_c > 0$ for all values of c up to a certain maximum value c_{max} and $N_c = 0$ for all $c > c_{max}$.

(a) Show that the sum of smoothed Good-Turing frequency estimates over all word types in the training corpus is $1 - p_{unseen}$

נראה כי: $\displaystyle\sum_{c=1}^{c_{max}} N_c \cdot \frac{(c+1)N_{c+1}}{N_c \cdot N} = 1 - P_{unseen} = 1 - \frac{N_1}{N}$

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

מכיוון שאנו סופרים את סך כל המופעים של מילה, במספר המילים הקיימות בקורפוס.

מכאן נסיק: $\displaystyle\sum_{c=2}^{c_{max}} N_c \cdot c = N - N_1$ ★

מתקיים:

$$\sum_{c=1}^{c_{max}} N_c \cdot \frac{(c+1)N_{c+1}}{N_c \cdot N} = \sum_{c=1}^{c_{max}-1} N_c \cdot \frac{(c+1)N_{c+1}}{N_c \cdot N} + N_{c_{max}} \frac{(c_{max}+1) N_{c_{max}+1}}{N_{c_{max}} \cdot N} = 0$$

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

$$\overset{\star}{=} \frac{1}{N} \cdot (N - N_1) = 1 - \frac{N_1}{N} \blacksquare$$

(b) Write down the equation for the smoothed Add-One estimate of a frequency of a word that appears c times in the training corpus. Show that there is a threshold μ, such that for all words of frequency less than μ, their smoothed estimate is higher than the MLE, and for all words of frequency more than μ, their smoothed estimate is lower than the MLE.

המשוואה עבור Smoothed Add-One, עבור מילה w היא:
עבור הקורפוס בגודל $N \in \mathbb{N}$

$$q_{add1}(w) = \frac{c(w)+1}{\sum_{i=1}^{N}(c(w_i)+1)} = \frac{c(w)+1}{N+\sum_{i=1}^{N} c(v_i)}$$

נראה כי קיים μ ע״פ השאלה.

עבור $w \in$ קורפוס המילים:

$$q_{add1}(w) = \frac{c(w)+1}{N+\sum_{i=1}^{N} c(v_i)}$$

$$MLE(w) = \frac{c(w)}{\sum_{i=1}^{N} c(w_i)}$$

נבדוק מתי מתקיים $q_{add1}(w) > MLE(w)$ ולכן:

$$\frac{c(w)+1}{N+\sum_{i=1}^{N} c(w_i)} > \frac{c(w)}{\sum_{i=1}^{N} c(w_i)}$$

נסמן $\sum_{i=1}^{N} c(w_i) = y$

$$\Downarrow$$

$$\frac{c(w)+1}{N+y} > \frac{c(w)}{y} \qquad (y,\ N+y > 0)$$

$$\Updownarrow$$

$$y\,c(w) + y > N c(w) + y\,c(w) \qquad / - y\,c(w)$$

$$\Updownarrow$$

$$\gamma > N_c(w) \iff \frac{\gamma}{N} > c(w)$$

$$\Downarrow$$

נקבל את הנדרש ש $= \frac{\gamma}{N}$

בנוסף נשים לב כי אם היינו בודקים את אי השוויון מהכיוון השני, כלומר
אי השוויון ההפוך זהה

$\blacksquare$

(c) Show that the property in (b) does not necessarily hold for the smoothed Good-Turing estimate.

נראה כי הטענה אינה בהכרח נכונה באמצעות דוגמא נגדית:

נגדיר אוצר מילים V והקורפוס באופן הבא:

$|V| = 102$, כך ש: 100 מילים מופיעות 1 פעמים בקורפוס
מילה אחת מופיעה פעמיים
ומילה אחת מופיעה 3 פעמים

נניח בשלילה כי הטענה מתקיימת, כלומר: עבור $w \in V$ כך ש: $c(w)=1$
מתקיים:
לוקחים אחד מ-100

$$MLE(w) = \frac{1}{100\cdot 1 + 1\cdot 2 + 1\cdot 3} = \frac{1}{105}$$

$$\left(;\frac{(c+1)N_{c+1}}{N_c \cdot N}\right) \rightarrow GT(w) = \frac{2\cdot N_2}{N_1\cdot 105} = \frac{2}{100{,}500}$$

$(N_2=1)$

$(N_1=100)$

$$\Downarrow$$

$$MLE(w) > GT(w)$$

ולכן אם קיים μ אזי צריך להתקיים $\mu \leq 1$

כעת עבור $w \in V$ כך ש: $C(w)=2$ מתקיים:

$$MLE(w) = \frac{2}{105}, \quad GT(w) = \frac{3 \cdot N_3}{N_2 \cdot N} = \frac{3}{105}$$

$$\Downarrow$$

$$GT(w) > MLE(w)$$

ולכן אם קיים μ אזי צריך להתקיים $\mu \geq 2$

בסתירה לכך ש: $\mu \leq 1$

ולכן μ כזה לא בהכרח קיים

$\blacksquare$

4. (15 pts)

(a) Write down the equation for a trigram language model (without detailing the probability estimations). Which (conditional) independence assumption is made in the model?

(b) Give an example of an English sentence and a Hebrew sentence where the phenomenon of verb-subject agreement (see below) is captured by the model in (a). That is, give an example where the model in (a) is likely to predict the correct inflection of the verb, given the subject.

(c) Give an example of an English sentence and a Hebrew sentence where subject-verb agreement is not captured. Which n (for an n-gram model) is necessary for capturing this phenomenon in your example?

Note: Subject-verb agreement is the correspondence between the morphological inflection of the verb and the type of the subject. For instance, in English where the subject is singular, verbs in present tense end with an 's', while the base form is used for plural subjects. (e.g., "a dog bark**s**", "dogs bark")

a

משוואה למודל שפת טריגמה:

מודל טריגמה מעריך את ההסתברות של מילה בהינתן שתי המילים הקודמות

המשוואה היא:

$$P(S) = P(w_1) \cdot P(w_2 | w_1) \cdot \prod_{i=3}^{n} P(w_i | w_{i-2}, w_{i-1})$$

הנחה: המודל מניח שההסתברות של מילה תלויה רק בשתי המילים
המיידיות הקודמות והיא בלתי תלויה בשאר המשפט.

b

משימה: לתת דוגמאות באנגלית ועברית בהן הסכמה בין נושא לפועל
נתפסת היטב ע"י מודל טריגרם:

* דוגמה באנגלית: "The cat sleeps"

במשפט זה הנושא "cat" והפועל "sleeps" צמודים אחד לשני
מודל טריגמה השומר את ההסתברות של מילה על סמך שתי המילים
הקודמות שלה יתפוס למעשה את הסכמה בין הנושא היחיד לבין הפועל
היחיד.

* דוגמה בעברית: "הכלב רץ"

כאן גם פה הנושא הוא "הכלב" והפועל - "רץ"

גם פה מודל הטריגמה יחזה נכון את צורת הפועל היחיד בעקבות הנושא היחיד.

(c) נספק דוגמאות בעברית ואנגלית בהם מודל הטריגרם לא יצליח לתפוס נושא-פועל ונציין את ה-n המינימלי עבור מודל n-gram

* דוגמה מאנגלית: "The cat, after being fed, sleeps"

גם פה הנושא - "cat" והפועל "sleeps" מופרדים ביותר משני מילים ("לאחר שהאכילו")
מודל טריגרם שמתחשב רק בשתי המילים המיידיות שלפני המילה הנוכחית לא יוכל לתפוס את ההסכמה בין "החתול" ל-"ישן"
המודל יפספס את הקשר בין הנושא והפועל בגלל המילים האחרות.

ה-"n" ההכרחי עבור מודל זה נצטרך מודל 5-gram (כלומר n=5)

* דוגמה בעברית: "הכלב, לאחר שאכל, רץ"

בדומה לדוגמה מאנגלית הנושא - "הכלב" והפועל "רץ" מופרדים לאורך זמן? המילים "לאחר שאכל" (כלומר לאחר אכילה)

גם פה מודל טריגמה לא יספיק כדי לתפוס את הנושא-פועל בגלל המרחק בין השניים

ה-"n" ההכרחי עבור מודל n-gram הוא n=5

5. (5 pts) Give an example of a sentence (in Hebrew or in English), where each consecutive pair of words is grammatically valid, but the sentence is not grammatically valid. Do the same with consecutive triplets and consecutive 4-tuples. You will note that doing this exercises for 4-tuples is considerably more difficult than for pairs. What does that indicate, in terms of the suitability of Markov models (of various orders) to be language models?
Note: A sequence of words is said to be *grammatically valid* in a language, if there is a grammatical sentence in that language that contains the sequence as a sub-string.

* משפט עם זוגות נכונים מבחינה דקדוקית: כל זוג מילים נכון מבחינה דקדוקית אבל המשפט בכללותו לא:

משפט באנגלית: Yesterday he and her friend went to the mall

ניתן לראות כי כל זוג מילים נכון מבחינה דקדוקית אבל המשפט בכללותו אינו תקין בגלל שלשת המילים: "he and her"

* משפט עם שלשה - שלישיות תקינות מבחינה דקדוקית:

Yesterday he and his friend is going to the mall

גם פה כל שלשה של מילים הגיונית אבל המשפט בכללותו אינו כך.
עקב המילים המסומנות

* משפט עם רביעיות נכונות:

He will go to the store and bought some milk

כל רצף של 4 מילים נכון מבחינה דקדוקית ניתן לראות, עם זאת המשפט המלא לא נכון מבחינה תחבירית (זמן עתיד ועבר) - מסומן

הסקנה לגבי מודל מרקוב מסדר n: ניתן לראות שככל שעולים בסדר המודל
קשה יותר למצוא דוגמאות לא תקינות למשפט שנכונות עבור רצפים של n

מכאן ניתן להסיק שמודלים מרובים מספר 4 והלאה (שיש תלות של 4 מילים
ויותר) יתנו תוצאות למשפטים אמינים יותר.

2 Practical

```
Question 2
The word with the highest probability to continue the sentence:'I have a house in' is: `the`

Question 3
The log-probability of the sentence: `Brad Pitt was born in Oklahoma` is -inf
The log-probability of the sentence: `The actor was born in USA` is -29.66684697358842
The perplexity of the sentences is: inf

Question 4
The log-probability using linear interpolation for: `Brad Pitt was born in Oklahoma` is -36.19159630034921
The log-probability using linear interpolation for: `The actor was born in USA` is -30.99285170428172
The perplexity of the sentences using linear interpolation is: 270.07616191453576
```