

67658 ወጪ ተወስኝ የሚያስተካክል

207766 u78 - 7'23 11C

2091KK013 - גראניט

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 $1 \leq i \leq n$ 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.

Bigram $S_2 \mid W$ STOP (2) START (1) $\sum_{W \in W \cup \{\text{START}\}} P(W) = 1$: $P(S_2 \mid W) > 0$ $\forall W \in W \cup \{\text{START}\}$

$$A = \{(\text{START}, w_1, \dots, w_n, \text{STOP}) \mid n \in \mathbb{N} \wedge \forall i \in \{1, \dots, n\} \quad w_i \in W\}$$

Տարբերակը $P(S^*) = 0$ է թափանցիկ պահումներում:

$w_0 = \text{START}$ \cup $\{w_i\}_{i=0}^{\infty}$ \cup $\{\text{join}\}$ \cup S^* \cup $\{\text{join}\}$ \cup B \cup $\{\text{remove}\}$

$w_i \in W$ $i \in N$ گز

$$P(S^*) = P\left(\bigwedge_{i=1}^{\infty} w_i = w_i^*\right) = \prod_{i=1}^{\infty} P(w_i^* | w_{i-1}, \dots, w_0) = \prod_{i=1}^{\infty} P(w_{i-1}^* | w_i)$$

menen δδj Bigram menen δδj
 $w_0 \equiv \text{START}$

类似地， $P()$ 为 $0 \leq P(u|w) < 1$ 的所有 $w \in W$ 和 $u \in W \cup \{\text{START}\}$ 。

$$P(h|w) \leq \sum_{v \in W} P(v|w) < \sum_{v \in W} P(v|w) + P(\text{STOP}|w) = 1$$

↓
0 ≤ P()
(1)

↓
v ∈ W
(2)

: אוניברסיטת תל אביב - מרכז מחקר אטומרי ופלוטוני

$$0 \leq P(S^*) = \prod_{i=1}^{\infty} P(\omega_{i+1}^* | \omega_i^*) \leq \prod_{i=1}^{\infty} \alpha^{i-1} = \lim_{n \rightarrow \infty} \alpha^n = 0 \Rightarrow P(S^*) = 0$$

$$P(\text{प्राप्ति } S) = P(S \in A) = \sum_{S \in A} P(S) = \sum_{S \in A} 0 = 0$$

$$P(\text{not } S) = P(S^c) = 1 - P(S) = 1 - 0.5 = 0.5$$

- (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).

Given Σ and $M = \{0, 1\}$. We want to show that $P(\text{START}, w_1, \dots, w_n, \text{STOP}) \neq 0$ for all $n \in \mathbb{N}$.

$$P(w_1=1 | \text{START}) = P(w_1=0 | \text{START}) = P(w_1=\text{STOP} | \text{START}) = \frac{1}{3} : (\text{START}, w_1, \dots, w_n, \text{STOP}) \quad n \in \mathbb{N}$$

$$\forall 1 < i \in \mathbb{N} \quad P(w_i=0 | w_{i-1}=0, \dots, w_1=0) = \frac{1}{2} \cdot \left(1 - \frac{1}{i^2}\right)$$

$$P(w_i=1 | w_{i-1}=0, \dots, w_1=0) = \frac{1}{2} \cdot \left(1 - \frac{1}{i^2}\right) \rightarrow w_{i-1}, \dots, w_1 \quad \text{using induction}$$

$$P(w_i=\text{STOP} | w_{i-1}=0, \dots, w_1=0) = \frac{1}{i^2}$$

$$0 < \frac{1}{i^2} < 1 \quad i > 1 \quad \text{thus } P(w_i=\text{STOP} | w_{i-1}=0, \dots, w_1=0) > 0 \quad 1 < i \in \mathbb{N}$$

$$\sum_{w \in W \cup \{\text{STOP}\}} P(w_1=w | \text{START}) = 1 : \text{PDI} \quad \sum_{k \in W \cup \{\text{STOP}\}} P(w_i=k | w_{i-1}, \dots, w_1) = 1 : 1 < i \in \mathbb{N}$$

Given $(w_i^*)_{i=1}^{\infty}$ such that $w_i^* \in \Sigma$ for all $i \in \mathbb{N}$.

$$P(S^*) = P\left(\bigcap_{i=1}^{\infty} (w_i^* \neq \text{STOP})\right) = P(w_1 \neq \text{STOP}) \cdot \prod_{i=2}^{\infty} P(w_i \neq \text{STOP} | w_{i-1} \neq \text{STOP}, \dots, w_1 \neq \text{STOP}) =$$

$$= (1 - P(w_1=\text{STOP})) \cdot \prod_{i=2}^{\infty} \sum_{k \in W} P(w_i=k | w_{i-1} \neq \text{STOP}, \dots, w_1 \neq \text{STOP}) = \frac{2}{3} \cdot \prod_{i=2}^{\infty} 2 \cdot \frac{1}{i^2} \cdot \left(1 - \frac{1}{i^2}\right) =$$

$$= \frac{2}{3} \cdot \prod_{i=2}^{\infty} \frac{i^2 - 1}{i^2} = \frac{2}{3} \cdot \prod_{i=2}^{\infty} \frac{(i-1)(i+1)}{i^2} = \frac{2}{3} \cdot \left(\frac{1}{2} \cdot \frac{3}{2} \cdot \frac{5}{3} \cdot \frac{7}{5} \cdot \frac{9}{7} \cdot \dots\right) = \frac{2}{3} \cdot \frac{1}{2} \cdot \lim_{k \rightarrow \infty} \frac{k+1}{k} = \frac{1}{3} \cdot 1 = \frac{1}{3} > 0$$

Thus $P(S^*) > 0$, which contradicts the fact that $P(S) = 0$.

Therefore, $P(S) \neq 0$.

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 $p(w_1, w_2, \dots, w_n)$ 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:

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

- (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?

בכדי מילוי אוסף Uni-gram נקבעו מילים יוצאות במאמר כמיון מילים. מילוי אוסף Uni-gram מושג על ידי חישוב סכום המספרים של מילים שפירושן מופיע במאמר. מילוי אוסף Uni-gram מושג על ידי חישוב סכום המספרים של מילים שפירושן מופיע במאמר.

$$P(\text{He went where there more opportunities}) < P(\text{He went were there more opportunities})$$

Unigram few & ~↑

$$P(w_2 = \text{"where"}) > P(w_2 = \text{"were"}) \Leftrightarrow \frac{c(\text{"where"})}{|V|} > \frac{c(\text{"were"})}{|V|} \Leftrightarrow c(\text{"where"}) > c(\text{"were"})$$

↗
more frequent

לעתה נזכיר מושג אחד שקיים ב邏輯 של מודלים טבילים. אם במאגר נתונים יש לנו מושג w ופונקציית $f(w)$ שפונקציית f מושג w מציין את הסיכוי שפה w תופיע כ单词 w במאגר נתונים.

ככל

- (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?

$$V \text{ נסמן כטבלה של מושגים}: C(u, w) \text{ נסמן כטבלה של מושגים}: C(u, w) = \sum_{\alpha \in W} \sum_{w' \in W} C(\alpha, w')$$

$$P(w_{i+1} = u | w_i = w) = P(u | w) = \frac{C(u, w)}{\sum_{\alpha \in W} C(\alpha, w)}$$

$$P(S) = P(\prod_{i=1}^n w_i) = \prod_{i=1}^n P(w_i | \prod_{j=i+1}^n w_j) = \prod_{i=1}^n P(w_i | w_{i+1})$$

$$\sum_{w \in W} P(w | w) = \frac{\sum_{w \in W} C(w, w)}{\sum_{\alpha \in W} C(\alpha, w)} = 1 : w \in W \text{ בסיס רצוי}$$

לכדי שפונקציית $P(w | w)$ תהיה סכום של n מושגים, נזקיף S ב-

המקרה w .

לעתה נזכיר מושג אחד שקיים ב邏輯 של מודלים טבילים. אם במאגר נתונים יש לנו מושג w ופונקציית $f(w)$ שפונקציית f מושג w מציין את הסיכוי שפה w תופיע כ单词 w במאגר נתונים.

לעתה נזכיר מושג אחד שקיים ב邏輯 של מודלים טבילים. אם במאגר נתונים יש לנו מושג w ופונקציית $f(w)$ שפונקציית f מושג w מציין את הסיכוי שפה w תופיע כ单词 w במאגר נתונים.

לעתה נזכיר מושג אחד שקיים ב邏ஜון של מודלים טבילים.

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 $p_{\text{unseen}} = \frac{N_1}{N}$.

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

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 $1 - p_{\text{unseen}}$

לעתה נוכיח ש $C \leq C_{\max}$ מגדיר N_c ככזה ש $P_{\text{unseen}} = 0$.

ההנחה היא $C < C_{\max}$ ו $N_c = 0$ עבור $c > C_{\max}$. נוכיח ש $\sum_{c=1}^{C_{\max}} N_c = N$.

ההנחה היא $P_{\text{unseen}} = \frac{N_1}{N}$ ו $N = \sum_{c=1}^{C_{\max}} c \cdot N_c$.

ההנחה היא $1 - P_{\text{unseen}} = 1 - \frac{N_1}{N} = \frac{N - N_1}{N} = \frac{1}{N} \cdot \sum_{c=2}^{C_{\max}} c \cdot N_c = \frac{1}{N} \cdot \sum_{c=1}^{C_{\max}-1} (c+1) \cdot N_{c+1}$.

ההנחה היא $\frac{(c+1) \cdot N_{c+1}}{N_c \cdot N} \geq 1$.

ההנחה היא $\sum_{c=1}^{C_{\max}} N_c = N$.

ההנחה היא $\sum_{c=1}^{C_{\max}} (c+1) \cdot N_{c+1} \leq N$.

ההנחה היא $\sum_{c=1}^{C_{\max}} N_c = N$.

- (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.

$$\begin{aligned} \text{ר' } \vartheta_{\text{add-one}}(w) &= \frac{C(w)+1}{\sum_{u \in W} (C(u)+1)} = \frac{C(w)+1}{N+|W|} \quad \left(|W| = \sum_{c=1}^{C_{\max}} N_c \right) \quad \text{Since } C(w) \text{ if } \\ &\quad (|V| = N) \\ \text{ר' } \vartheta_{\text{MLE}}(w) &= \frac{C(w)}{N} \quad \text{, Now we can use MLE over probabilities} \\ \vartheta_{\text{add-one}}(w) > \vartheta_{\text{MLE}}(w) &\iff \frac{C(w)+1}{N+|W|} > \frac{C(w)}{N} \iff \frac{C(w)+1}{C(w)} > \frac{N}{N+|W|} \iff \\ &\iff 1 + \frac{1}{C(w)} > 1 + \frac{|W|}{N} \iff \frac{1}{C(w)} > \frac{|W|}{N} \iff C(w) < \frac{N}{|W|} = \mu \quad \left(\text{ר' } \vartheta_{\text{MLE}} \text{ if } C(w) \leq \mu \right) \end{aligned}$$

$$q_{\text{add-one}}(w) > q_{\text{MLE}}(w) \iff C(w) < \mu \quad \text{for } \mu = \frac{|V|}{|W|}$$

$$\text{BUT} \quad q_{\text{add-one}}(w) < q_{\text{MLE}}(w) \iff C(w) > \mu$$

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

Given that q_{SGT} : smoothed Good-Turing does not always act as threshold to $C(w)$.

$N = |V| = 14$, $|W| = 10$. $C(w)$ is defined as $C(w) = \frac{N_w}{N}$.

$$N_1 = 3, N_2 = 2, N_3 = 1, N_4 = 1$$

$$\sum_{k=1}^{C_{\max}} k \cdot N_k = 14 \quad \text{so } C_{\max} = 4$$

$C(w) = 1$ for $w \in W$ such that $C(w) = 1$

$$q_{SGT}(w) = \frac{2 \cdot N_2}{N_1 \cdot N} = \frac{4}{14} = \frac{2}{7}, \quad q_{MLE}(w) = \frac{1}{14} = \frac{1}{14} \Rightarrow q_{SGT}(w) > q_{MLE}(w)$$

$C(w) = 2$ for $w \in W$ such that $C(w) = 2$

$$q_{SGT}(w) = \frac{3 \cdot N_3}{N_2 \cdot N} = \frac{3}{28}, \quad q_{MLE}(w) = \frac{2}{14} = \frac{1}{7} \Rightarrow q_{SGT}(w) < q_{MLE}(w)$$

$C(w) = 3$ for $w \in W$ such that $C(w) = 3$

$$q_{SGT}(w) = \frac{4 \cdot N_4}{N_3 \cdot N} = \frac{4}{14}, \quad q_{MLE}(w) = \frac{3}{14} \Rightarrow q_{SGT}(w) > q_{MLE}(w)$$

$$C_{\max} = 4 \Rightarrow N_5 = 0$$

$$q_{SGT}(w) = \frac{5 \cdot N_5}{N_4 \cdot N} = 0, \quad q_{MLE}(w) = \frac{4}{14} \Rightarrow q_{SGT}(w) < q_{MLE}(w)$$

$q_{SGT}(w) < q_{MLE}(w)$ because w has the same frequency as the threshold μ .

$q_{SGT}(w) > q_{MLE}(w)$ because w has a higher frequency than the threshold μ .

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?

Trigram Sun W W ən l'sn ʃələ m S l'ɒt ʃən 722 unzəʊn
 $\begin{matrix} W_1 \\ ||| \end{matrix}$ $\begin{matrix} W_0 \\ ||| \end{matrix}$: S əll e

$$P(S) = P\left(\bigcap_{i=1}^n W_i\right) = \prod_{i=1}^n P(W_i | \bigcap_{j=i-1}^{i-1} W_j) = \prod_{i=1}^n P(W_i | W_{i-1}, W_{i-2})$$

- (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.

ՀԱՅՐ ՀԱՅՐ : "ԵՐԵՎԱՆ ՏԵՇԵ ՀԵՄ"

"Mark is running : IS he not

גנץ מזכיר כי החלטת הממשלה לחייב את כל תושבי הארץ ללבוש מסכת בימי שישי ושבת נזקפת עליה.

לעתות אחדים מושג ייחודי אחד בודד כנושא של דפוס נקרא בשם "singular subject 'Mark'" ו- "singular subject 'John'"

"Mark" - 1 בזבזת פסנתר מס' 821 Erigram Seite PRA. 105

የዚህ የወጪ በዚህ የወጪ በዚህ የወጪ በዚህ የወጪ በዚህ የወጪ በዚህ የወጪ በዚህ የወጪ

גּוֹלָן וְרִגְלָס

- (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?

הילדר רוזנברג: "כונקציון רצינית הפלורנטינה רמי"

"The dog of my neighbour is white": *redundant*

"is" -> "n2" subjects -> "dog" -> "run" *verb*

የፍቅር በአዲስ አበባ የትራንስፖርት ስራው የተደረገው ተግባር ነው.

— n -gram များ ဖြစ်တဲ့ အကြောင်း ဘုရား ပါ ဘူး။ မြန်မာ ရွှေမြန်မာ ပြောတဲ့ အကြောင်း ဘုရား ပါ ဘူး။

Re^proves Subject-Verb agreement \rightarrow no own vs 5-gram size issues

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.

Pairs: { "name": "John", "age": 25 } { "name": "Jane", "age": 28 }

Triplets: N_2 O_2 F_2 Cl_2 Br_2 I_2

א-טְּוָלֵפֶס: תְּזִקְנַתְּנָה בְּגַדְגֹּלָה וְבְגַדְגֹּלָה

କାହିଁ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

বাসিন্দা করে জোড় কু. মুসলি পুরুষের নাম গ্রন্থ হচ্ছে এবং তার বাসিন্দা হচ্ছে আর কুমুদী।

and beyond the two most recent years have been added. So far, 1975 through 1981 are available as n-gram statistics.

מִתְבָּאֵשׁ בְּלֹאֶתֶן וְלֹא־בְּלֹאֶתֶן, אֲבָא אַבְּבָאֶתֶן וְבְּלֹאֶתֶן

... בְּאַתָּה תִּשְׁמַח אֶת־בְּנֵי־יִשְׂרָאֵל וְעַל־בְּנֵי־עֲמָקָם.

8. **השאלה ה' (הנושאים)** – מילוי תבניות ופונקציות.

2 Practical (40 points)

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

The next most probable word for the sentence "I have a house in..."

according to the bigram model is: "the"

3. Using the bigram model:

- compute the probability of the following two sentences (for each sentence separately).
- 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 "

a) The log probability for the sentence " Brad Pitt was born in Oklahoma " according to the bigram model is: $-\infty$ (i.e. the probability is 0)

The log probability for the sentence " The actor was born in USA " according to the bigram model is: -29.6668

b) The perplexity of the bigram model according to the above sentences is: ∞

4. Now estimate a new model using linear interpolation smoothing between the bigram model and unigram model with $\lambda_{\text{bigram}} = 2/3$ and $\lambda_{\text{unigram}} = 1/3$ and using the same training data. Given this new model, compute the probability and the perplexity of the two sentences above.

The log probability for the sentence " Brad Pitt was born in Oklahoma " according to the interpolated model is: -36.1916

The log probability for the sentence " The actor was born in USA " according to the interpolated model is: -30.9928

The perplexity of the interpolated model according to the above sentences is: 270.0773

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 $w_1 w_2 w_3 \dots w_n$ STOP (where w_i for $1 \leq i \leq n$ 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 $p(w_1, w_2, \dots, w_n)$ 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?
- (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 barks", "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)
```

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.