NLP - Ex1

Matan Toledano, ID: [redacted]

Daniel Levin, ID: [redacted]

## Question 1:

In order to prove that the sum of the probabilities over all finite sequences is 1, we will show that the probability of the complement event is equal to 0.
In other words, we want to show that the probability of getting an infinite sentence (sequence of words without STOP at the end) is 0. Let's call this event A. Our goal is to show that

$$P(A) = \Pi_{i=1}^{n}\left(\overbrace{1 - \mathbb{P}(STOP|w_i)}^{*}\right) = 0$$

* - the probability not to get a token STOP in sequence of n words.
Denote $x_i = 1 - \mathbb{P}(STOP|w_i)$. Notice that, since

$$\forall i \in [n] : 1 \geq \mathbb{P}(STOP|w_i) > 0$$

(from the given), it holds that

$$\forall i \in [n] : 0 \leq x_i = 1 - \mathbb{P}(STOP|w_i) < 1$$

Which is equal to:

$$0 \leq \underset{i\in[n]}{max}\{x_i\} \overset{\star}{<} 1$$

Let's aspire the quantity of the words in sentence to infinity and find the limit:

$$\lim_{n\to\infty} \mathbb{P}(A) = \lim_{n\to\infty} \Pi_{i=1}^{n}\left(1 - \mathbb{P}(STOP|w_i)\right) = \lim_{n\to\infty} \Pi_{i=1}^{n}\left(x_i\right) \leq$$

$$\leq \lim_{n\to\infty} \Pi_{i=1}^{n} \underset{j\in[n]}{max}\left\{x_j\right\} = \lim_{n\to\infty}\left(\underset{j\in[n]}{max}\left\{x_j\right\}\right)^n \overset{\star}{=} 0. \blacksquare$$

2. (a) A unigram language model for the "were-where" spelling corrector would estimate the probability for each occurrence, based on the training set, using the estimator:

$$q_{SPL}(x_m = w_i) = \frac{c(w_i)}{\sum_j c(w_j)}$$

c stands for the count of a word in the training set.

The classifier would classify a sentence containing the words "were" or "where" as more probable according to the **relative frequency** of were-where in the training corpus. That is to say the more frequent of the two will always be corrected to it in the test set.

Given the sentence "He went where there where more opportunities", the spelling corrector would give a right answer for the first instance iff $c("where") > c("were")$ in the training set.

Likewise, the second instance of "where" would be corrected to "were" iff $("were") > c("where")$ in the training set.

As the unigram model isn't sensitive to context and therefore must return the same answer to each occurrence, a unigram model cannot classify correctly the sentence as "He went *where* there *were* more opportunities".

(b) The spelling corrector would use the bigram estimator:

$$q_{SPL}(x_m = w_j | x_{m-1} = w_i) = \frac{c(w_i, w_j)}{\sum_{j'} c(w_i, w_{j'})}$$

And will classify a sentence as the answer if the occurrences of where-were are the most probable according to the context of the surrounding words. That is to say the spelling corrector will output "were" for the second instance if:

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

We expect this model to be better than the unigram because of the different contexts that we expect to encounter "where"/"were" and not the other. Which are:

where:

- At the start of a sentence.
- Using POS tagging, followed by a copula "are"/"is".

were:

- After you/we/they.

And both words we do not expect to follow themselves: *where where or *were were will probably be ungrammatical. Those different contexts and the information that a **good** training set will provide will make it better than a context-agnostic model.

However, a sentence using this model may get a zero probability: if where/were will be adjacent to a word that didn't appear near it the training set – this sentence will be disqualified, even though this sentence may be judged as an acceptable sentence by humans. In the same time, unigram will not disqualify this sentence completely because each word in the vocabulary appears at least once and therefore each sentence must have a positive probability.

This could be a problem for the model as the training set cannot cover all possible cases for word pairings, but this problem could be solved by smoothing. On the other hand, even without smoothing this model could be helpful as ungrammatical pairs cannot happen at all such as: "He were", "Why where" and so on.

## Question 3:

**(a)** The following complement has got the highest probability based on the given corpus and on bigram language model:

"He likes NLP"

**(b)** (1) START John likes Mary STOP
(2) START Mary likes John STOP

**i.** (1) -

$$\mathbb{P}(s_1) = \mathbb{P}(John|START) \cdot \mathbb{P}(likes|John) \cdot \mathbb{P}(Mary|likes) \cdot \mathbb{P}(STOP|Mary) =$$

$$= \frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{64}$$

(2) -

$$\mathbb{P}(s_2) = \mathbb{P}(Mary|START) \cdot \mathbb{P}(likes|Mary) \cdot \mathbb{P}(John|likes) \cdot \mathbb{P}(STOP|John) =$$

$$= \frac{1}{6} \cdot \frac{1}{2} \cdot \frac{0}{4} \cdot \frac{1}{4} = 0$$

**ii.** (1) -

$$Perplexity = 2^{-l} = 2^{-\frac{1}{M} \cdot \Sigma_{i=1}^{m} \log p(s_i)} = 2^{-\frac{1}{5} \cdot \Sigma_{i=1}^{2} \log p(s_i)} =$$

$$= 2^{-\frac{1}{5} \cdot (\log\left(\frac{1}{64}\right) + \log(0))} = 2^{-\frac{1}{5} \cdot (-6 + (-\infty))} = 2^{-(-\infty)} = \infty$$

Based on the given data the language model is infinitely perplexed.

**(c)** Recall the general formula of linear interpolation using bigram and unigram distributions and maximum likelihoods:

$$q(w_i|w_{i-1}) = \lambda_1 \cdot q_{ML}(w_i|w_{i-1}) + \lambda_2 \cdot q_{ML}(w_i) =$$

$$= \frac{2}{3} \cdot q_{ML}(w_i|w_{i-1}) + \frac{1}{3} \cdot q_{ML}(w_i)$$

**i.** (1) - first let us assume that unigram model doesn't take START and STOP keywords into account, since they appear in every sentence and they don't carry any probablistic importance rather than denote borders of sentences in data, that is to say $q_{ML}(STOP) = q_{ML}(START) = 0$. Thus number of tokens in the training data equals to 18.

$$\mathbb{P}(s_1) = \mathbb{P}(John|START) \cdot \mathbb{P}(likes|John) \cdot \mathbb{P}(Mary|likes) \cdot \mathbb{P}(STOP|Mary) =$$

$$= \left(\frac{2}{3} \cdot q_{ML}(John|START) + \frac{1}{3} \cdot q_{ML}(John)\right) \cdot \left(\frac{2}{3} \cdot q_{ML}(likes|John) + \frac{1}{3} \cdot q_{ML}(likes)\right) \cdot$$

$$\cdot\left(\frac{2}{3}\cdot q_{ML}(Mary|likes)+\frac{1}{3}\cdot q_{ML}(Mary)\right)\cdot\left(\frac{2}{3}\cdot q_{ML}(STOP|Mary)+\frac{1}{3}\cdot q_{ML}(STOP)\right)=$$

$$=\left(\frac{2}{3}\cdot\frac{1}{2}+\frac{1}{3}\cdot\frac{4}{18}\right)\cdot\left(\frac{2}{3}\cdot\frac{1}{4}+\frac{1}{3}\cdot\frac{4}{18}\right)\cdot$$

$$\cdot\left(\frac{2}{3}\cdot\frac{1}{4}+\frac{1}{3}\cdot\frac{2}{18}\right)\cdot\left(\frac{2}{3}\cdot\frac{1}{2}+\frac{1}{3}\cdot 0\right)=\frac{220}{19683}\approx 0.0112$$

(2) -

$$\mathbb{P}(s_2)=\mathbb{P}(Mary|START)\cdot\mathbb{P}(likes|Mary)\cdot\mathbb{P}(John|likes)\cdot\mathbb{P}(STOP|John)=$$

$$=\left(\frac{2}{3}\cdot q_{ML}(Mary|START)+\frac{1}{3}\cdot q_{ML}(Mary)\right)\cdot\left(\frac{2}{3}\cdot q_{ML}(likes|Mary)+\frac{1}{3}\cdot q_{ML}(likes)\right)\cdot$$

$$\cdot\left(\frac{2}{3}\cdot q_{ML}(John|likes)+\frac{1}{3}\cdot q_{ML}(John)\right)\cdot\left(\frac{2}{3}\cdot q_{ML}(STOP|John)+\frac{1}{3}\cdot q_{ML}(STOP)\right)=$$

$$=\left(\frac{2}{3}\cdot\frac{1}{6}+\frac{1}{3}\cdot\frac{2}{18}\right)\cdot\left(\frac{2}{3}\cdot\frac{1}{2}+\frac{1}{3}\cdot\frac{4}{18}\right)\cdot$$

$$\cdot\left(\frac{2}{3}\cdot\frac{0}{4}+\frac{1}{3}\cdot\frac{4}{18}\right)\cdot\left(\frac{2}{3}\cdot\frac{1}{4}+\frac{1}{3}\cdot 0\right)=\frac{20}{59049}\approx 0.0003$$

**ii.**

$$Perplexity=2^{-l}=2^{-\frac{1}{M}\cdot\Sigma_{i=1}^{m}\log p(s_i)}=2^{-\frac{1}{5}\cdot\Sigma_{i=1}^{2}\log p(s_i)}=$$

$$=2^{-\frac{1}{5}\cdot\left(\log\left(\frac{220}{19683}\right)+\log\left(\frac{20}{59049}\right)\right)}\approx 12.14423$$

Now we see that the perplexity of current model for given test sentences is higher than that computed in previous section. It means that this model is more precise in estimating the probabilities of the given test sentences based on the given training data.

## Question 4:

Let's denote the maximal frequency in the given corpus by $c_{max}$.
Denote a set of all the existing word types in the corpus by $W$.
For word type $w \in W$ denote the frequency estimate of smoothed Good-Turing by $\widetilde{p}(w)$.
For frequency $c$ denote the probability of a word occuring given its frequency in training data by $\hat{p}(c)$.
Denote Good-Turing by $GT$.
Recall that $p_{unseen} = \frac{N_1}{N}$. We want to show that $\sum_{w \in W} \widetilde{p}(w) = 1 - p_{unseen}$.
Proof:

$$\sum_{w \in W} \widetilde{p}(w) \overset{*}{=} \sum_{c=1}^{c_{max}-1} \hat{p}(c) \cdot N_c \overbrace{=}^{GT\,definition} \frac{1}{N} \sum_{c=1}^{c_{max}-1} \frac{(c+1) \cdot N_{c+1}}{N_c} \cdot N_c =$$

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

$$= \frac{1}{N} \sum_{c=1}^{c_{max}-1} (c+1) \cdot N_{c+1} + \frac{N_1}{N} - \frac{N_1}{N} = \frac{1}{N} \sum_{c=0}^{c_{max}-1} (c+1) \cdot N_{c+1} - \frac{N_1}{N} =$$

$$= \frac{1}{N} \sum_{c=1}^{c_{max}} c \cdot N_c - \frac{N_1}{N} \overbrace{=}^{\star} \frac{1}{N} \cdot N - \frac{N_1}{N} = 1 - p_{unseen} \blacksquare$$

$*$ - $\widetilde{p}(w)$ stands for frequency of each word type presenting in the corpus, therefore summing them up over all the existing tokens in the corpus is equal to summing up products of probability of words occured $c$ times with quantity of such words (that occured exactly $c$ times).
Additionally $\hat{p}(c) = \frac{(c+1) \cdot N_{c+1}}{N \cdot N_c} = 0$ for all $c \in \{0, c_{max}, c_{max}+1, ...\}$ resulting from the **Note**. Consequently, the only indexes that contribute to the sum expression are those varying between 1 and $c_{max} - 1$.
$\star$ - $\sum_{c=1}^{c_{max}} c \cdot N_c$ - is exactly a counter of size of the corpus - sum of all the possible tokens frequencies multiplied by number of tokens occuring corresponding quantity of times.

**(b)**

The equation of the smoothed Add-One estimate of a frequency of a word that appears c times in the training corpus:

Where $|V|$ - the size of the vocabulary, $c(w)$ - count of the word $w$ in the training corpus,

$N$ - overall number of tokens in the corpus. Recall the MLE estimate: $q_{MLE}(w) = \frac{c(w)}{N}$.

We want to find a threshold $\mu$ such that

$$\forall w \in V, c(w) < \mu : q_{add-1}(w) > q_{MLE}(w)$$

and also

$$\forall w \in V, c(w) > \mu : q_{add-1}(w) < q_{MLE}(w)$$

Let's perform a sequence of equal transitions for the inequality we've got:

$$q_{add-1}(w) > q_{MLE}(w) \iff$$

$$\frac{c(w)+1}{N+|V|} > \frac{c(w)}{N} \iff$$

$$N \cdot c(w) + N > N \cdot c(w) + |V| \cdot c(w) \iff$$

$$N > |V| \cdot c(w) \iff$$

$$c(w) < \frac{N}{|V|}$$

All the transitions are legal since we assume that the training corpus is not empty which implies that $|V|, N, |V| + N > 0$.

Denote:

$$\mu = \frac{N}{|V|}$$

Notice that the chosen value of $\mu$ that depends exclusively on metadata of the training corpus, satisfies the first requirement. Similarly, by means of reversing the inequality signs we ensure that the second requirement

fulfilled as well. So for every training corpus

$$\mu = \frac{N}{|V|}$$

will uphold both requirements.

Let's bring a counter example. Let's define a training corpus consisting of 1 word, let's denote it A. Then $c(A) = c_{max} = 1$. Respectively $N_0 = N_1 = 1$ and otherwise $N_i = 0$. We will show that for every $\mu \geq 2$ (otherwise there are no words with frequency less than $\mu$) there exists $w \in V$ with $c(w) < \mu$ such that it holds that

$$\frac{(c(w)+1) \cdot N_{c(w)+1}}{N \cdot N_{c(w)}} = q_{GT}(w) \leq q_{MLE}(w) = \frac{c(w)}{N}$$

and thus we contradict the first requirement of the claim of the previous section which is satisfying because of AND statement.
Let be $\mu \geq 2$, take $w = A$, then:

$$q_{GT}(A) = \frac{(1+1) \cdot N_{1+1}}{N \cdot N_1} = \frac{2 \cdot 0}{1 \cdot 1} = 0 \leq \frac{1}{1} = \frac{c(B)}{N} = q_{MLE}(B)$$

as needed.

5. (a) Given a sentence with words $x_1, \dots, x_n$:

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

Using the assumption that:

$$p(x_i | x_{i-1}, \dots, x_0) = p(x_i | x_{i-1}, x_{i-2})$$

(b) The **girl jumps**.

**היא** לא **הלכה**.

(c) The **boy** that lived across the street **likes** the girl.

For this example, a 7-gram would be necessary to capture correctly.

**הכבשה** של המלך הצרפתי הראשון **גולחה** כל ראש חודש.

As for this, a 6-gram would be necessary.

# Question 6:

## (1)

An example in Hebrew of grammatical invalidity consisted of consecutive grammatically valid pairs: "*האיש רוקד התהפך". Obviously, the pairs האיש רוקד and רוקד התהפך exist either independently or as a part of another sentences, but the entire sentence is ungrammatical in Hebrew since the participle form doesn't show agreement in definiteness to the nucleus of the nominal phrase.

## (2)

An example in Hebrew of grammatical invalidity consisted of consecutive grammatically valid triplets: "*משענת הכיסא הזה מיועד לישיבה נוחה". Again, all the subsequences of the sentence of length 3 are legal but the sentence itself is grammatically wrong since the gender agreement between subject and predicate has been violated contradicting a plain rule of Hebrew grammar.

## (3)

An example in Hebrew of grammatical invalidity consisted of consecutive grammatically valid 4-tuples: "*נתתי עיתון של אישה מבוגרת".

Again the tuples נתתי עיתון של אישה and עיתון של אישה מבוגרת are very regular in Hebrew as parts of another sentences, however the entire sentence is not grammatical as long as it stands independently, i.e. without any context this sentence is ungrammatical since the verb “to give” in Hebrew is ditransitive and it necessarily requires two objects - direct and indirect. In this example the latter is missing.

Conclusion:

As we could ensure while making up these examples, the last one appeared to be the most challenging. In context of Markov model suitability, it means that the more consequent words the model consider, less mistakes it will do on estimating the likelihood of a sentence. For example, bigram model may infer that every consequent pair is grammatical, which as we saw is indeed possible with high probability, although the entire sentence may easily be a complete nonsense both semantically and grammatically.