Prof. Ryan Cotterell

Johanna Ribas: Assignment 4

jribas@student.ethz.ch

12/01/2023 - 19:10h

Question 1: Calculating Prefix Probabilities

a)

$$\sum_{\mathbf{w}\in\Sigma^*} \tilde{p}(\mathbf{w}) = \sum_{\mathbf{w}\in\Sigma^*} \prod_{n=1}^{|\mathbf{w}|} p\left(w_n\right) =$$

$$\sum_{\mathbf{w}\in\Sigma^1} p\left(w_n\right) + \sum_{\mathbf{w}\in\Sigma^2} \prod_{n=1}^{2} p\left(w_n\right) + \sum_{\mathbf{w}\in\Sigma^3} \prod_{n=1}^{3} p\left(w_n\right) + \cdots = \tag{1}$$

$$1 + \left(\sum_{w\in\Sigma} p(w)\right)^2 + \left(\sum_{w\in\Sigma} p(w)\right)^3 + \cdots = 1 + 1 + 1 + \cdots$$

This is an infinite addition of 1, i.e., it tends to infinity.

b) Now we have that $\sum_{w\in\Sigma} p(w) < 1$, i.e, equation(1) becomes a geometric series and the infinite sum is: $\frac{1}{1-\sum_{\mathbf{w}\in\Sigma^*}}$.
With these results:

$$\sum_{\mathbf{w}\in\Sigma^*} p(\mathbf{w}) = p(EOS) \cdot \frac{1}{1-\sum_{\mathbf{w}\in\Sigma^*} p(w)} = \left(1 - \sum_{w\in\Sigma} p(w)\right) \cdot \frac{1}{1-\sum_{w\in\Sigma} p(w)} = 1 \tag{2}$$

c)

$$p_{pre}(\mathbf{w}) = \sum_{u\in\Sigma^*} p(\text{EOS} \mid \mathbf{wu}) \prod_{n=1}^{N} p\left(w_n \mid w_0 \cdots w_{n-1}\right) \tag{3}$$

$$
\begin{aligned}
\sum_{u\in\Sigma} p(wu) &= \sum_{u} p(EOS \mid \omega u) \prod_{n=1}^{|wu|} p\left((wu)_n \mid ((wu)_0, \ldots, ((wu)_{n-1}\right) \\
= \sum_{u} p(EOS \mid \omega u) \prod_{n=1}^{|\omega|} p\left(w_n \mid \omega_1, \omega_2, \ldots, \omega_{n-1}\right) &\prod_{m=1}^{|\omega|} p\left(u_m \mid \omega, u_1, \ldots, u_{m-1}\right) \\
&= p_{pre}(\omega) \sum_{u\in\Sigma} p(\text{ EOS} \mid \omega u) \prod_{m=1}^{|\omega|} p\left(u_m \mid w, u_1, \ldots, u_{m-1}\right) = \\
&= p_{pre}(\omega)\left(\sum_{u\in\Sigma} p(EOS \mid \omega) + \sum_{u\in\Sigma} p\left(EOS \mid \omega u_1\right) p(u_1 \mid \omega) + \ldots\right) = \\
(*) &= p_{pre}(\omega)(p(EOS \mid \omega) + \sum p\left(u_1 \mid \omega\right) - \sum_{u_1, u_2} p\left(u_1 \mid \omega\right) p\left(u_2 \mid \omega u_1\right) + \\
&+ \sum_{u_1 u_2}\left(1 - \sum_{u_3} p\left(u_3 \mid w u_1 u_2\right)\right) p\left(u_1 \mid w\right) p\left(u_2 \mid w u_1\right) + \cdots) = \\
&= p_{pre}(\omega)\left(1 + \sum \prod_{n=1}^{|u|} p\left(u_n \mid w, u_1, \ldots, u_{n-1}\right) - \right. \\
&\left. - \sum \prod_{n=1}^{|u|} p\left(u_n \mid w, u_1, \ldots, u_{n-1}\right)\right) = \\
&= P_{\text{pre}}(\omega)
\end{aligned} \tag{4}
$$

In (*), we can only perform this step because of local normalization.

d) The CKY algorithm can be used to compute the probability of a sentence w under a PCFG by constructing a parse tree for the sentence and then using the probabilities associated with the grammar rules used in the tree to compute the overall probability of the sentence. Specifically, for each rule $X \to YZ$, multiply the probability of the rule by the probabilities of the spans generated by B and C. The overall probability of the sentence w is the product of the probabilities of all the rules used in the parse tree.

   To compute the probability of a sentence w using the CKY algorithm, we follow the following steps:

   Initialize a 2-dimensional table T with the same number of rows and columns as the length of the sentence w. Each cell T[i][j] in the table will contain a set of nonterminals that can generate the substring of w from i to j.

   Fill in the table T by considering each possible split of the sentence w into two substrings. For each split k, compute the set of nonterminals that can generate the substring from i to k and the substring from k+1 to j, and then use the rules of the PCFG to combine these sets of nonterminals to create the set of nonterminals that can generate the substring from i to j.

   Once the table is filled in, the probability of the sentence w can be computed by looking at the probability of the start symbol generating the entire sentence. This probability is given by the product of the probabilities of the rules used to derive the sentence, as specified by the PCFG.

e)

$$p(S \stackrel{*}{\Longrightarrow} wv) = \sum_{\mathbf{t} \in T_s(w_i:w_k)} p(\mathbf{t}) = \sum_{\mathbf{u} \in \Sigma^*} p(\mathbf{wu}) \tag{5}$$

f) Initialize a two-dimensional array P with dimensions $|N|x|N|$, where $|N|$ is the number of non-terminal symbols, and set all elements of P to 0.

Iterate over all non-terminal symbols X, Y and Z in the grammar. For each pair (X, Y), compute the probability $p_{lc}(Y|X)$ as given by equation 13 and set the element P[X][Y] in the array P to this value.

For each triple (X, Y, Z), compute the probability $p_{lc}(YZ|X)$ as given by equation 14, and set the element P[X][Y][Z] in the array P to this value.

The runtime of this algorithm is $O(|N|^2)$ for computing the left-corner probabilities $p_{lc}(Y|X)$ and $O(|N|^3)$ for computing the left-corner probabilities $p_{lc}(YZ|X)$.

g) To compute the prefix probability in a tree we can factorize the probability of every split up to the terminals wi,..., wk. From X, we reach YZ along the left corner of the tree, and the probability of this is given by the left corner probability: $X \rightarrow YZ\alpha$. Y yields the derivation trees producing $w_i, ..., w_j$ and the probability of this is given by the inside probability : $p_{inside}(w_i, ..., w_j|Y)$. Last, the right derivation from X' (Z), yields the last words $w_{j+1}, ..., w_k$ but not the EOS and therefore this is also a prefix probability given by $p_{pre}(w_{j+1}...w_k|Z)$. To compute $p_{pre}(w_i...w_k|X)$ we will have to sum this computed joint probability for each word between indices i and k and sum over all non-terminals Y,Z children of X that yield these words.

Question 2:

Link to colab notebook:

`https://colab.research.google.com/drive/1ZyIbOvNyH_t8WcEcmd3XiOd4p3WxsPc9?usp=sharing`