Introduction to NLP (CS7.401)

Spring 2022, IIIT Hyderabad
25 Jan, Tuesday (Lecture 5)

Taught by Prof. Manish Shrivastava

Smoothing (contd.)

Witten-Bell Discounting

This method has a similar idea to Kneser-Ney, but w.r.t the context rather than the central word. That is, if $P(w_i \mid w_{i-1} \dots w_{i-m}) = 0$, then $P_{\text{WB}}(w_i \mid w_{i-1} \dots w_{i-m})$ is higher if $w_{i-1} \dots w_{i-m}$ occurs with many different words w_i.

For example, consider bigrams. Let $T(w)$ be the number of different types that occur to the right of w; $N(w)$ the number of word occurrences to the right of w, also represented as $c(w, \cdot)$; and $Z(w)$ is the number of bigrams in the dataset starting with w that do not occur in the training data.

Mathematically,

$$P_{\text{WB}}(w_i \mid w_{i-1}) = \begin{cases} \frac{T(w_{i-1})}{Z(w_{i-1})(N+T(w_{i-1}))} & c(w_{i-1}, w_i) = 0 \\ \frac{c(w_{i-1}w_i)}{N+T(w_{i-1})} & c(w_{i-1}, w_i) > 0. \end{cases}$$

This method is more conservative in subtracting probability mass. However, it has the disadvantage that if w_{i-1} and w_i do not occur in the training data, the smoothed probability remains zero.

Part-of-Speech Tagging

Part-of-speech tagging consists of assigning grammatical categories to every word in a sentence. The complexity of this task lies in the fact that a certain word can have different parts of speech depending on its context.

Formally, POS tagging is a sequence labelling task – given a sequence of observations $W = [w_1, \dots, w_n]$, we need to produce a tag sequence $T = [t_1, \dots, t_n]$, such that $P(T \mid W)$ or $P(W, T)$ is maximised. Stated concisely, we need to compute

$$\underset{T}{\text{argmax}}\, P(T \mid W).$$

However, it is not possible to directly compute $P(T \mid W)$ from the data. Thus, we use Bayes' Theorem

$$P(T \mid W) = \frac{P(W \mid T) \cdot P(T)}{P(W)},$$

and then get

$$\begin{aligned} T^* &= \operatorname*{argmax}_T P(T \mid W) \\ &= \operatorname*{argmax}_T P(W \mid T) \cdot P(T) \\ &= \operatorname*{argmax}_T P(W, T). \end{aligned}$$

since $P(W)$ is independent of T.

$P(W \mid T)$ is reliably computable, given the limited number of grammatical categories.
Another aspect of the comparison is that W and T are *sequences of elements*. In computing $P(T \mid W)$, we would have computed

$$\prod_i P(t_i \mid w_i),$$

which is entirely context-independent for each word w_i. This defeats the purpose of the sequence-based formalism of the problem.
Computing $P(W \mid T)P(T)$ is more accurate since, even if we split the $P(W \mid T)$ as above, the $P(T)$ term provides the necessary context-sensitivity. We can therefore compute it as

$$\operatorname*{argmax}_T \prod_i P(w_i \mid t_i) P(t_i \mid t_{i-1}).$$

Hidden Markov Models

This process can be visualised as a Markov state machine, where the states are POS tags and a word is produced as output at every transition.
As an analogy, consider three urns, one consisting of only red balls, one of only blue, and one of only green. We have the transition probability matrix between urns. Since each urn can only give one type of ball, any sequence of balls implies an unique sequence of urns.
However, if each urn had a mixture of balls of different colours, it is no longer unique. We now need to find the *most probable* sequence of urns that could give rise to a given sequence of balls. This is extremely difficult to compute.

Hidden Markov models are formalised with the following parameters:

- a set of states S
- an output alphabet V

- transition probabilities $A = [a_{ij}]$
- emission probabilities $B = [b_j(o_k)]$
- initial state probabilities π

These parameters constitute an HMM $\lambda = (A, B, \pi)$.