Lecture Notes: N-gram Language Models

CSCI 375: NLP / Katie Keith /Williams College / Fall 2024

1 Probability Review

- A *random variable* maps outcomes of a random process to real numbers. We will use capital letters (e.g., X) to refer to random variable and lowercase letters (e.g., x) to refer to specific values the random variable takes. For example, if the event is a coin flip, X, we can use $X = 1$ to indicate when the coin comes up heads and $X = 0$ to indicate the coin comes up tails and $X = x$ to refer to either.
- We can estimate probabilities using a **relative frequency estimator**, calculating the ratio of how many times an event occurred in a series of trials, divided by the total number of trials conducted. For example, let's flip a coin 1000 times and count the number of times it lands heads, 498 times. Then

$$P(X = \text{heads}) = P(X = 1) = \frac{\text{Count(heads)}}{\text{Total trials}} = \frac{498}{1000} \tag{1}$$

  As the number of trials approaches infinity, the probability estimated by the relative frequency estimator will approach the true probability (it is *unbiased*).
- We denote the *joint probability* of random variables X and Y as $P(X, Y)$. In other classes, you may have used different notation such as $P(X \cap Y)$.
- We define the *conditional probability* of a random variable X given random variable Y as $P(X|Y) = P(X, Y)/P(Y)$
- We define the *marginal probability* of random variable (also called "the law of total probaiblity") X as $P(X) = \sum_{y \in \text{domain}(Y)} P(X, Y = y)$

2 Language Models

We formally define a **language model** as a model that, given a discrete vocabulary $\mathcal{V}$ (created after tokenization), computes the probability of a sequence of n words. *Note,* here we are dropping the capital random variables for simplicity,

$$P(w_1, w_2, \ldots, w_n) \tag{2}$$

For example, we may want the probability of the sentence attributed to Pablo Picasso "*Computers are useless, they only give you answers*".

Language modeling can also refer to the task of computing the probability of an upcoming word given its context (the preceding $n - 1$ words)

$$P(w_n|w_1, w_2, \ldots, w_{n-1}) \tag{3}$$

which we also sometimes abbreviate as

$$P(w_n|w_{1:n-1}). \tag{4}$$

Returning to our running example, a first approach is to use a **relative frequency estimator** to calculate

$$P(\textit{Computers are useless, they can only give you answers}) \tag{5}$$

$$= \frac{\text{Count}(\textit{Computers are useless, they can only give you answers})}{\text{Count(all sentences)}} \tag{6}$$

This estimator is:

- **Unbiased.** With infitite data, this estimated probability will be corrected. However, the estimator has
- **High variance.** Suppose we cap sentences as $n = 20$ words long and we only have a vocabulary size of $|\mathcal{V}| = 10^5$. Then we have $|\mathcal{V}|^{20} = (10^5)^{20} = 10^{100}$ (a googol!) possible sequences. So even gramatical sentences will have probability zero if we never saw them in our data.

Thus, we need to make a simplifying assumption, which we call **Markov assumption**, that the probability of the nth word only depends on the previous $N-1$ words

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-(N-1):n-1}) \tag{7}$$
$$= P(w_n|w_{n-N+1:n-1}) \tag{8}$$

If $N = 2$ this is called a **bigram** Markov assumption and the equation above simplifies to

$$P(w_n|w_{1:n-1})) \approx P(w_n|w_{n-2+1:n-1}) \tag{9}$$
$$= P(w_n|w_{n-1:n-1}) \tag{10}$$
$$= P(w_n|w_{n-1}) \tag{11}$$

Using this bigram assumption with a **relative frequency estimator** we get

$$P(w_n|w_{n-1}) = \frac{\text{Count}(w_{n-1}, w_n)}{\sum_{x \in \mathcal{V}} \text{Count}(w_{n-1}, x)} \tag{12}$$

The denominator makes this a valid probability; we need $\sum_{x \in \mathcal{V}} P(x|w_{n-1}) = 1$. Now, we can simplify the denominator (since the sum of all bigram counts that start with a given word must be equal to the unigram counts of that word), and arrive at

$$P(w_n|w_{n-1}) = \frac{\text{Count}(w_{n-1}, w_n)}{\text{Count}(w_{n-1})} \tag{13}$$