Understanding N-Gram Models Summary

Tatev Aslanyan

2023-11-29

N-Gram Models

- $P(w_n|w_{1:n-1})$: The probability of the word w_n given the entire history of words $w_{1:n-1}$.
- $P(w_n|w_{n-N+1:n-1})$: The approximated probability of w_n, considering only the last $N-1$ words.
- N: The size of the n-gram (e.g., 2 for bigrams, 3 for trigrams).

General N-Gram Model

The general formula for an N-Gram model is given by:

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

where N is the size of the n-gram.

Estimating Probabilities in N-Gram Models

In N-Gram models, we need to estimate the probability of a word w given a history h, or the probability of an entire word sequence W. This involves formalizing our notation and approach.

Notation and Probability of Sequences

- To represent the probability of a random variable X_i taking a specific value, say "the", we use $P(X_i = \text{"the"})$ or simply $P(\text{"the"})$.
- A sequence of n words is represented as $w_1 \ldots w_n$ or $w_{1:n}$. The notation $w_{1:n-1}$ represents the sequence $w_1, w_2, \ldots, w_{n-1}$.
- The joint probability of each word in a sequence having a particular value is denoted as $P(w_1, w_2, \ldots, w_n)$.

Chain Rule of Probability

The chain rule of probability allows us to decompose the probability of a sequence of words. For a sequence $w_{1:n}$, this is given by:

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \ldots P(w_n|w_{1:n-1}) \tag{2}$$

In a more general form, this can be expressed as:

$$P(w_{1:n}) = \prod_{k=1}^{n} P(w_k|w_{1:k-1}) \tag{3}$$

Applying Chain Rule to N-Gram Models

In N-Gram models, we approximate the probability of a word given its entire context by considering only the last few words. This simplification is crucial for practical computations and applications in language processing.

The chain rule of probability provides a foundational approach for computing probabilities in N-Gram models. By approximating the context of a word with a limited history, N-Gram models balance computational efficiency with linguistic relevance, making them widely used in various natural language processing tasks.

$$P(X_1 \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_{1:2}) \ldots P(X_n|X_{1:n-1}) \tag{4}$$

$$= \prod_{k=1}^{n} P(X_k|X_{1:k-1}) \tag{5}$$

Bigram Probability Calculation Using MLE

Given a hypothetical corpus with sentences including the phrase “the quick brown fox jumps over the lazy dog”, we calculate some bigram probabilities using Maximum Likelihood Estimation (MLE).

Incorporating Start and End of Sentence Tokens:

In language modeling, especially when dealing with n-grams, it is crucial to consider the beginning and end of sentences. To this end, we introduce special tokens: $< SOS >$ (Start Of Sentence) and $< EOS >$ (End Of Sentence). These tokens help in defining the boundaries of a sentence, allowing the model to learn the probability of a word occurring at the beginning or the end of a sentence. For instance, in our corpus, each sentence would start with $< SOS >$ and end with $< EOS >$. This inclusion ensures that transitions into the first word and from the last word of a sentence are appropriately modeled, enhancing the accuracy and effectiveness of the n-gram model.

Hypothetical Corpus:

1. $< SOS >$ The quick brown fox jumps over the lazy dog $< EOS >$
2. $< SOS >$ The lazy dog sleeps $< EOS >$
3. $< SOS >$ The fox and the dog are friends $< EOS >$

Calculating Bigram Probabilities:

1. **Probability of 'quick' given 'The':**

   $$P(\text{quick}|\text{The}) = \frac{C(\text{The quick})}{C(\text{The})}$$

   Assuming 'The quick' appears once and 'The' appears three times in the corpus:

   $$P(\text{quick}|\text{The}) = \frac{1}{3} \approx 0.33$$

2. **Probability of 'lazy' given 'the':**

   $$P(\text{lazy}|\text{the}) = \frac{C(\text{the lazy})}{C(\text{the})}$$

   Assuming 'the lazy' appears once and 'the' appears twice:

   $$P(\text{lazy}|\text{the}) = \frac{1}{2} = 0.5$$

3. **Probability of 'dog' given 'lazy':**

   $$P(\text{dog}|\text{lazy}) = \frac{C(\text{lazy dog})}{C(\text{lazy})}$$

   Assuming 'lazy dog' appears once and 'lazy' appears once:

   $$P(\text{dog}|\text{lazy}) = \frac{1}{1} = 1$$

General MLE Formula for Bigrams:

For any bigram w_n and w_{n-1}, the MLE probability is calculated as:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Maximum Likelihood Estimation in N-Gram Models:

For the general case of MLE n-gram parameter estimation, the probability of a word w_n given its preceding context of $N-1$ words is estimated as follows:

$$P(w_n|w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}w_n)}{C(w_{n-N+1:n-1})} \tag{6}$$

This equation is similar to previous one, estimates the n-gram probability by dividing the observed frequency of a particular sequence by the observed frequency of its prefix. This ratio is referred to as a relative frequency. As mentioned earlier, this approach of using relative frequencies to estimate probabilities is an example of Maximum Likelihood Estimation (MLE). In MLE, the resulting parameter set is the one that maximizes the likelihood of the training set T given the model M, i.e., $P(T|M)$.