N-gram

n-gram is a choice to represent the word, the sentence, or the document. It is originally used to predict the next word in a sentence.

Example for Bi-gram

$w_1 \boxed{w_2 \ w_3} \ w_4$
where the "$w_2 \ w_3$" is the bi-gram.

To calculate the probability of "w_4" after "$w_1 \ w_2 \ w_3$", using Markov assumption, we have

$$\begin{aligned} P\left(w_{1:n}\right) &= P\left(w_1\right) P\left(w_2 \mid w_1\right) P\left(w_3 \mid w_{1:2}\right) \ldots P\left(w_n \mid w_{1:n-1}\right) \\ &= \prod_{k=1}^{n} P\left(w_k \mid w_{1:k-1}\right) \end{aligned}$$

Large Language Model

Problem

- The model is too big: GPT-3 comes in eight sizes, ranging from 125M to 175B parameters.
- The training process is too slow: 355 GPU-years and cost $4.6M for a single training run.

Solution

Partition the training data

- Partition the training data into several parts.
- Use co-clustering to partition the training data.
- Make the ensemble process more efficient.

Notes

- Some literature research is done and no one pays attention to partition the training data in this way.
- The dataset for NLP can be formed as a matrix and those that can be formalized as a tensor can also be partitioned in this way.