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 "b_n" ($b_i = w_i w_{i+1}$), using Markov assumption, we have

$$\begin{aligned} P(b_{1:n}) &= P(b_1)\, P(b_2 \mid b_1)\, P(b_3 \mid b_{1:2}) \ldots P(b_n \mid b_{1:n-1}) \\ &= \prod_{k=1}^{n} P(b_k \mid b_{1:k-1}) \end{aligned}$$

N-gram and Co-clustering: A Conceptual Link

From N-gram to Co-clustering

- Markov Assumption: This principle is used to calculate the co-occurrence probability for each pair of words.
- Co-occurrence Probability Matrix: The calculated probabilities form a matrix, which is conceptually similar to the compatibility matrix used in co-clustering.

Co-occurrence Probability Matrix

$$p_{ij} := P\left(b_i \mid b_j\right) = P\left(b_i|b_{i-1}\right) P\left(b_{i-1}|b_{i-2}\right) \ldots P\left(b_{j+1}|b_j\right)$$

Challenges in Training Large Language Models

Training Duration is very Long

- Model Size: Larger models require more resources and time.
- Dataset Size: Ensemble Learning, parallel computing, via partitioning the dataset, is a common solution.

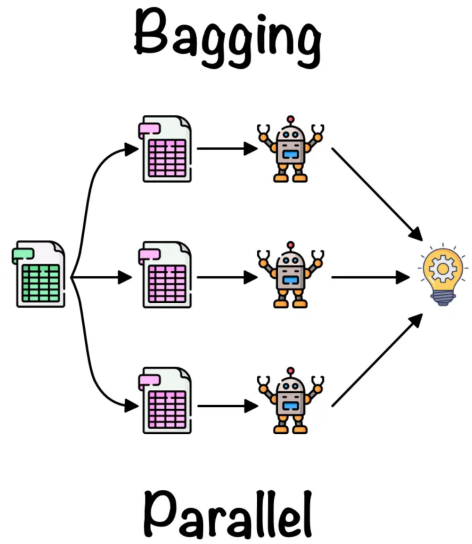

Figure: Ensemble Learning[1]

Partitioning Problem

- Random Partitioning: A common but problematic method due to it divides the related data into different parts.

[1] F. López, "Ensemble Learning: Bagging & Boosting," Medium, Jan. 18, 2021.

Proposed Solution: Co-clustering for Data Partitioning

Why Co-clustering?

- NLP Datasets as Matrices: Natural Language Processing datasets can be effectively represented as matrices, making them suitable for co-clustering.
- Improved Communication: Grouping related parts of the data together can reduce communication overhead during parallel processing.
- Reduced Training Set Size: Co-clustering similar training data can potentially decrease the size of the training set, thus speeding up the training process.