**Experimental Setup**

The primary objective of this study is to compare various methods for embedding process trees, including techniques such as Word2Vec, TF-IDF, Word Embedding, and Pre-trained Models. Specifically, the goal was to identify the method that most clearly highlights the differences between the groups. To achieve this, fine-tuning was also performed alongside the comparisons. Below, we detail the dataset generation process, the fine-tuning approach, and the similarity metric used for model evaluation.

* **Dataset**

To conduct this experiment, we generated six distinct groups of process trees using the pm4py library. These groups were designed with varying structural properties by adjusting parameters such as tree length (min/mode/max) and gateway types (sequence, choice, parallel, loop), as detailed in Table~\ref{Experimental Setting}. The activities within these trees were represented by random alphabetic characters, ensuring that the focus remained solely on the structural complexity and variability of the trees when evaluating the different embedding methods.

+ method 개별 소개

* **Fine-Tuning**

To enhance the performance of the embedding models, we conducted fine-tuning on SBERT (Sentence-BERT). The fine-tuning approach was inspired by SBERT's regression objective function. In a similar manner, we calculated our own similarity measure, which we used as a label for fine-tuning. This fine-tuning process aimed to refine the model's ability to capture the nuances of process tree similarities more effectively.

* **Graph Edit Similarity**

The similarity metric employed for the fine-tuning process was based on Graph Edit Distance. This metric provides a measure of similarity between process trees by calculating the minimum number of edit operations required to transform one tree into another. The specifics of how this metric was implemented and its impact on the fine-tuning process will be discussed in detail later in the paper.