- BERT structure: Embedding => Encoder 1 => Encoder 2 => … => Encoder 12
- Embedding: The hidden states of the last encoder layer, i.e., Encoder 12

Attn Heads	Hd Layers	Hd Size	With KW	Train Loss	Val Loss	Overfit	Sim within	Sim btw
12	12	768	×	≈ 0.2	> 2.5	√	-0.01/-0.01	-0.003
12	12	768	√	≈ 0.2	> 2.5	√	0.01/0.01	0.02
6	6	768	×	< 2	> 5	×	-0.29/-0.22	-0.26
6	6	768	√	≈ 2.3	≈ 5.5	×	-0.30/-0.22	-0.27
6	6	384	×	> 4	≈ 5.8	×	-0.13/-0.09	-0.11
6	6	384	√	> 4.5	≈ 6	×	-0.16/-0.12	-0.14
12	10 frz + 2	768	×	< 1	≈ 3	×	-0.08/-0.05	-0.07
12	8 frz + 4	768	×	< 1	≈ 3	×	-0.07/-0.05	-0.06
12	6 frz + 6	768	×	< 0.5	≈ 3	×	-0.03/-0.03	-0.03

Adjustment to the BERT model: Reduce the model capacity

- Num_attention_heads: from 12 (default) to 6
- Num_hidden_layers: from 12 (default) to 6
- Hidden_size: from 768 (default) to 384
- Hidden_dropout_prob: from 0.1 (default) to 0.2 ∵ try to prevent overfitting

Original size model

- Overfitting (in terms of BERT training obj.)
- Little similarity difference

Reduced size model

- Significant similarity difference (even within the same class)
- May also be a bad signal, since it tends to regard recommendations as far less similar to each other.

Therefore, I tried an alternative fine-tuning method

- Gradually unfreeze the hidden layers
- Top-down unfreezing
    - lower layers learn general patterns
    - higher layers learn domain-specific knowledge.
- 10 freeze + 2 unfreeze / 8 freeze + 4 unfreeze / 6 freeze + 6 unfreeze

After those fine-tuning attempts, I tried to replicate the previous workflow based on clustering, and compare the quality of the extracted recommendations with this workflow.

- Use doc #13 for evaluation, the rest 14 docs for clustering
- Clustering on the 87 recommendation exemplars in the 14 docs
    - Sentence embeddings: the mean of tokens embeddings
    - K-means: 3 clusters & centroids
- Filter sentences with extracted keywords as ‘candidate recommendations’
- Compute the minimum distance to the centroids, and sort
- Get top 25, judge by myself whether recommendation or not\
- Get the recommendation index and precision scores as follows

Original size model

- Config: default size and structure
- Overfitting (in terms of BERT training obj.)
- Little similarity difference
- 36% precision

Reduced size model

- Config: 6 heads, 6 layers, 384 hidden dims, with keyword filtering
- Significant similarity difference (even within the same class)
- 20% precision

Model with frozen layers

- Config: 10 freeze + 2 unfreeze
- Moderate similarity difference
- 40% precision

Baseline (Original BERT model)

- 1, 5, 7, 8, 10, 11, 12
- 7 / 25 = 0.28

Original size model

- 2, 3, 10, 11, 15, 19, 22, 23, 25
- 9 / 25 = 0.36

Reduced size model

- 2, 6, 13, 14, 16, 22
- 6 / 25 = 0.24 lowest

Model with frozen layers

- 1, 3, 5, 6, 11, 13, 15, 18, 20, 23
- 10 / 25 = **0.40 highest**