

Algorithm: Proxy-Tuning of Language Model

/ Applies proxy-tuning to adjust logits of a target model in an NLP task */*

Input: input_ids, model_base, model_tuned, model_target

Output: generated_text, a sequence of tokens

Hyperparameters: max_length (maximum generation length), n (number of tokens to generate)

Parameters: Θ includes all parameters for the base, tuned, and target models.

1. Initialize generated_tokens as an empty list
2. Encode input_text into input_ids using tokenizer $\Theta_{\text{tokenizer}}$
- // Perform token-wise proxy-tuning and text generation
3. For t in [1, ..., n]:
 - a. Obtain base, tuned, and target model logits:
 - i. logits_base \leftarrow model_base(input_ids).logits with parameters Θ_{base}
 - ii. logits_tuned \leftarrow model_tuned(input_ids).logits with parameters Θ_{tuned}
 - iii. logits_target \leftarrow model_target(input_ids).logits with parameters Θ_{target}
 - b. Proxy-tuning adjustment:
 - i. $\Delta\text{logits} \leftarrow \text{logits_tuned} - \text{logits_base}$
 - ii. logits' $\leftarrow \text{logits_target} + \Delta\text{logits}$
 - c. Normalize the logits for next token prediction:
 - i. predictions $\leftarrow \text{softmax}(\text{logits}', \text{axis}=-1)$
 - d. Select the next token:
 - i. next_token_id $\leftarrow \text{argmax}(\text{predictions})$
 - ii. Append next_token_id to generated_tokens
 - e. Update input_ids with next_token_id for the next iteration
4. Decode the sequence of generated_tokens into text using $\Theta_{\text{tokenizer}}$
5. **Return** generated_text