Demo Presentation: Method Generation Group

Changshu Liu
Huiyan Zhang
Zhangyi Pan

What is Method Generation?

{

"signature": "def do_transform(self, v=<NUM_LIT:1>):",

"**body**": "if not self.transform: <EOL> <INDENT> return <EOL> <DEDENT> try: <EOL><INDENT>self.latest_value = utils.Transform ...",

"docstring": "Apply the transformation (if it exists) to the latest_value",

"id": "f19:c4:m1"

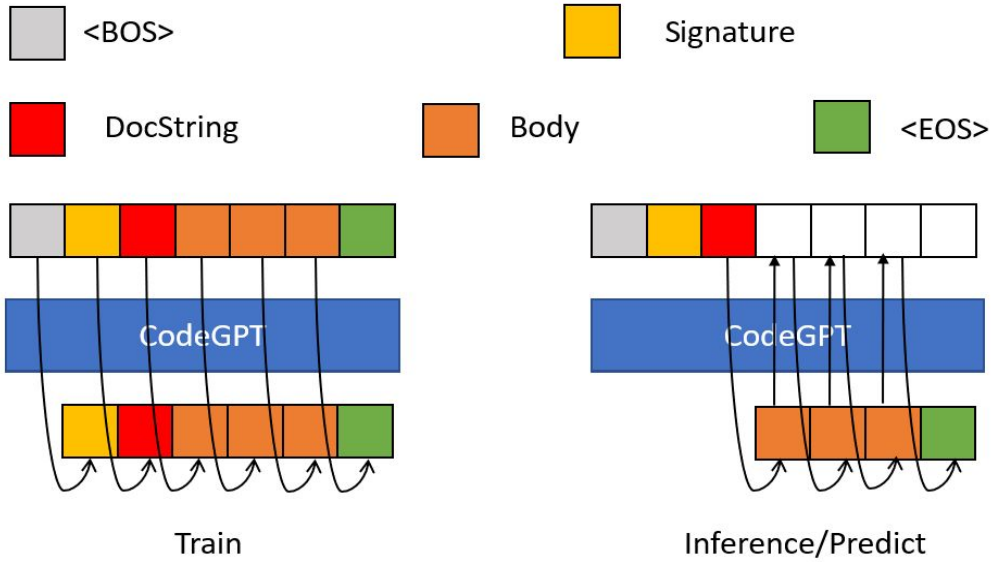
}

Given **signature** and **docstring**, generate **body**

CodeGPT for Method Generation

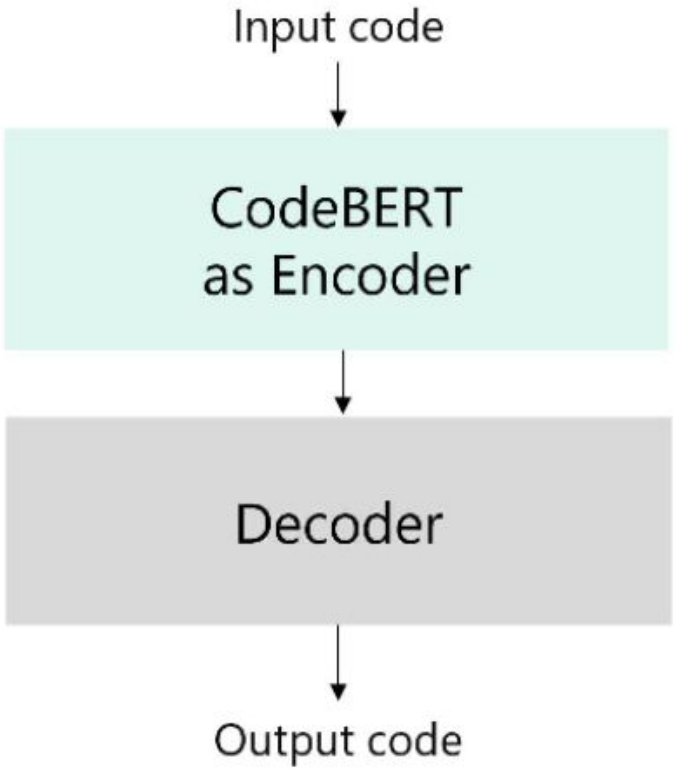
CodeXGLUE provided the implementation to fine-tune CodeGPT on method generation task.

GPT: use the previous token to generate the next token.



CodeBert?

- We can use a CodeBert(Transformer Encoder-Decoder) to do the same thing!
- Input Sequence: <BOS><Signature><Docstring><EOS>



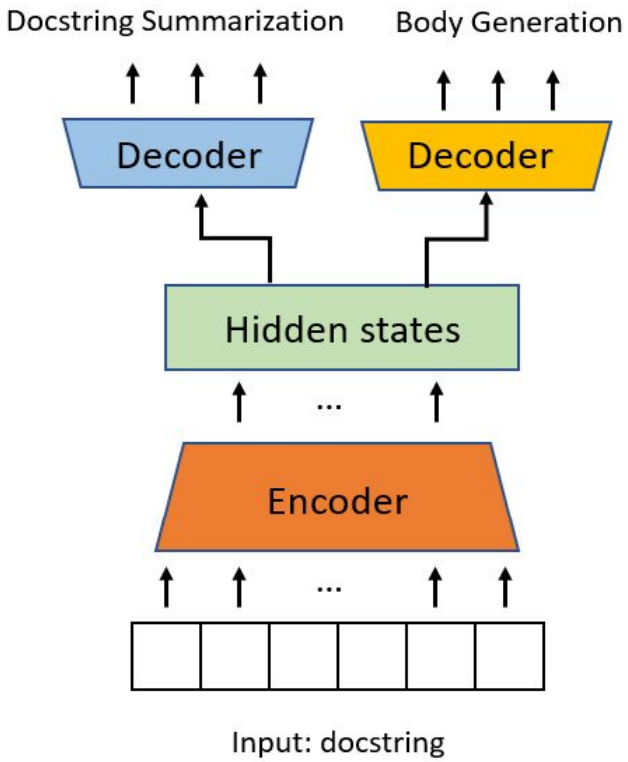
A Better Way to Use the Input information

- concat(signature, docstring) vs docstring

  BLEU-4: 19.42 vs. 19.03

- Our hypothesis:
  - Signature can play the role of the ‘abstract’ or ‘summarization’ of the docstring.
    - E.G. "def _get_syslog_secure(self):" → get syslog secure
  - The task of generating the summarization of docstring may help the encoder have a better understanding of the input sequence.

Multi-task learning



- Task1: Generating the Summarization of the docstring
- Task2: Generating the method body.
- Loss = loss_{task2} + α*loss_{task1}

Experiment Setups and Performances

- Dataset: train/valid/test: 22,338/500/500

  (original dataset:893,538 /20,000/20,000)
- Baseline: CodeGPT(*batch_size=2)

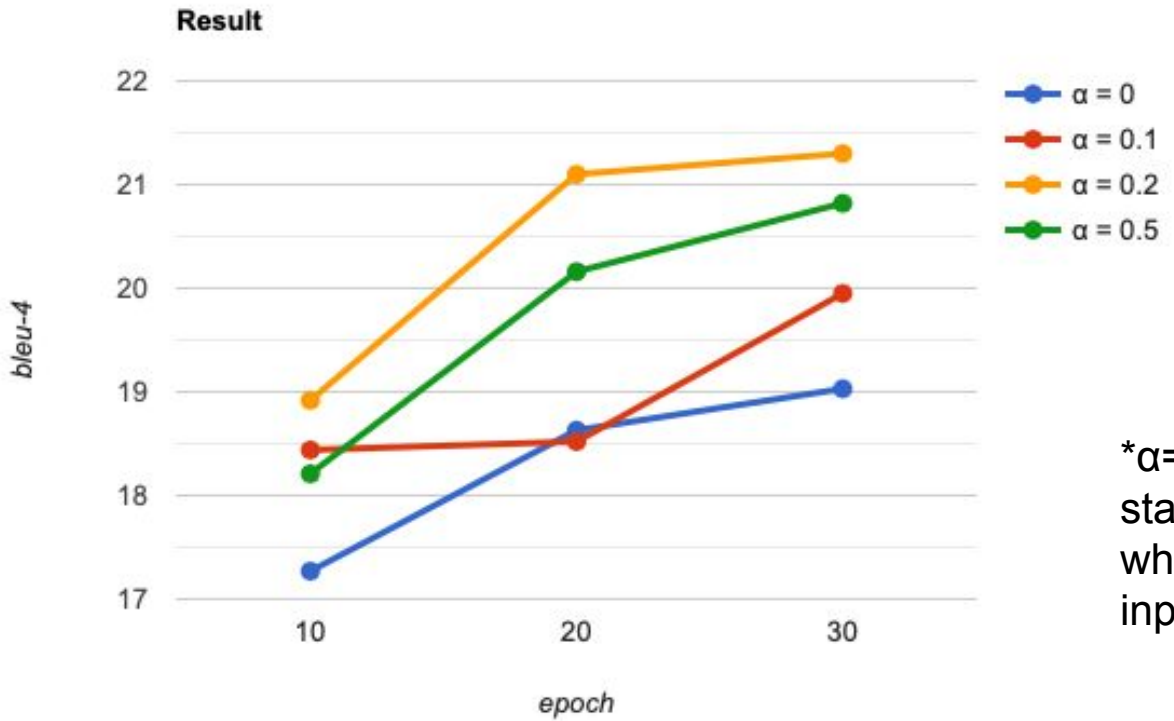
  Encoder-Decoder_concat (batch_size=12)

  Encoder-Decoder_multitask(batch_size=12)
- Train/Finetune for 30 epoches

Model	**BLEU-4**
CodeGPT	17.82
Encoder-Decoder_concat	19.42
Encoder-Decoder_multitask	**21.30**

Parameter Analysis and Ablation Study

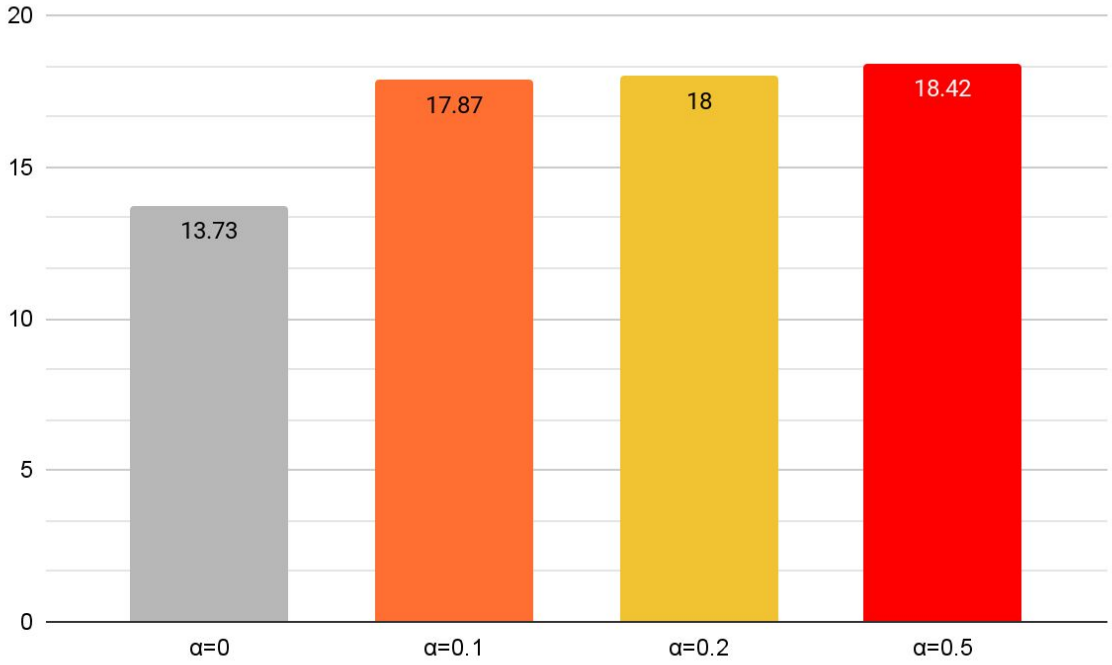
Impact of α



*α=0: only have one task (a standard encoder-decoder model which only uses docstring as the input)

Parameter Analysis and Ablation Study

Impact of pretrained weights (models are trained from scratch)



*α=0: only have one task (a standard encoder-decoder model which only uses docstring as the input)

A TEST:

{"signature": "def get_frame(self):",

"docstring": "return the Rectangle instance used to frame the legend",

"id": "f17172:c0:m12"}

GOLD : "return self.legendPatch<EOL>",

Thank you!