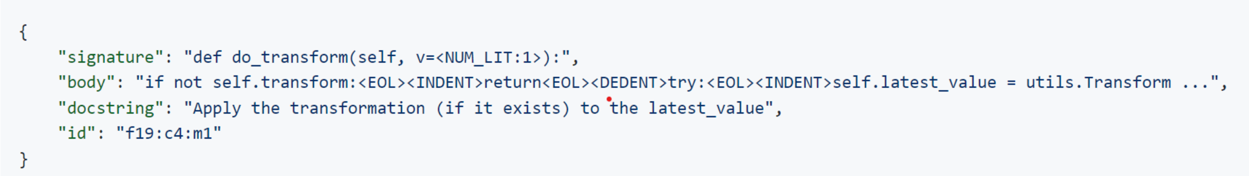
Project Proposal

Changshu Liu (cl4062), Zhangyi Pan (zp2223), Huiyan Zhang (hz2757)

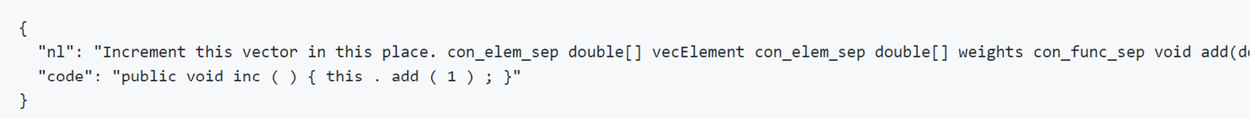
1. Method Generation

Our project focuses on a relatively new task: method generation. The definition of this task is given a docstring and signature (method name), and predicts the method body. In figure 1 we show an example from the dataset. Method generation is different from code generation, which also uses natural language description (docstring) to generate a piece of code. As is shown in figure 2, in code generation both method name and variable name are anonymized.



```
{
    "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"
}
```

Figure 1. An example from CodeSearchNet Python methods generation dataset



```
{
  "nl": "Increment this vector in this place. con_elem_sep double[] vecElement con_elem_sep double[] weights con_func_sep void add(d
  "code": "public void inc ( ) { this . add ( 1 ) ; }"
}
```

Figure 2. An example from Concode code generation dataset

2. Existing Method and its limitation

CodeXGLUE(https://github.com/microsoft/CodeXGLUE/tree/main/Code-Code/Method-Generation) provided a GPT-based baseline, which is shown in figure 3. It simply followed the pipeline of code generation and concatenated signature and docstring together as a new input. We argue that a concatenation may not fully exploit the connection between signature and docstring.

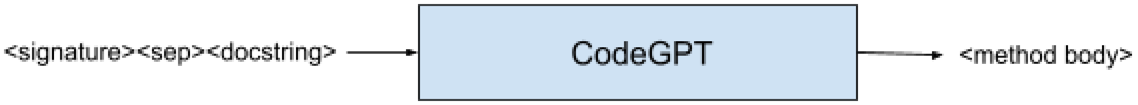


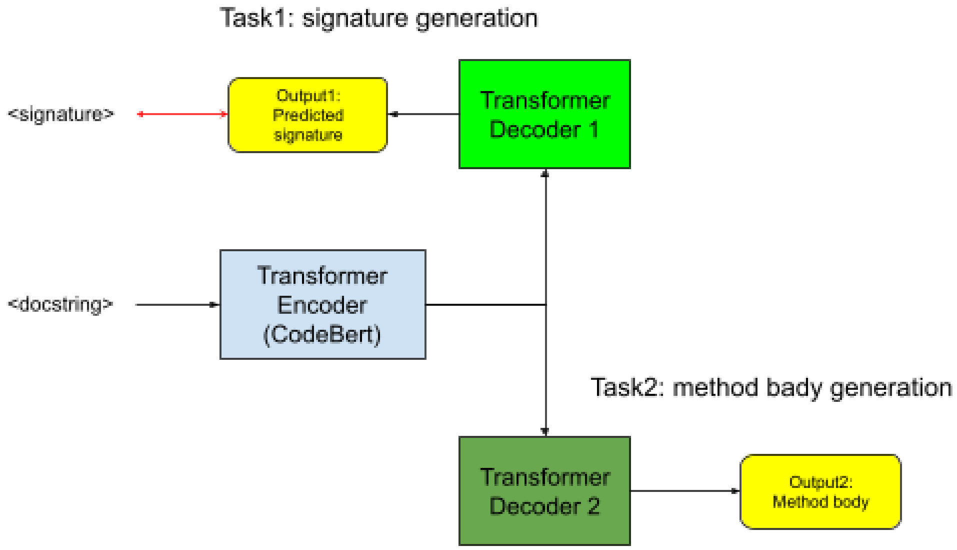
Figure 3. Baseline provided by CodeXGLUE

3. Our Design

Our hypothesis is that signature(function name) can be a high-level summarization of docstring, for example 'Apply the transformation ……' can be concluded as 'do transform'. Based on this, we set up our multi-task learning framework as below. Our model has a common encoder and two different decoders. The first encoder is based on our hypothesis and iit will generate the function name. We assume that this task will help the common encoder learn a better understanding of developers' intention. The second decoder, which will generate the method body will also benefit from a better encoder.

4. Dataset

CodeXGLUE(https://github.com/microsoft/CodeXGLUE/tree/main/Code-Code/Method-Generation) released a huge dataset which consists of 893,538 training data,20,000 validation data and 20,000 test data. We plan to sample a smaller dataset at first to see if our method work, then we can try to train our model on the full dataset.


Task1: signature generation
<signature>
Output1: Predicted signature
Transformer Decoder 1
<docstring>
Transformer Encoder (CodeBert)
Task2: method bady generation
Transformer Decoder 2
Output2: Method body