

Analysis of “Using Pre-Trained Models to Boost Code Review Automation”

Team Members: Oana-Andreea Ilie, Albert-Andrei Havarneanu, Iulia-Diana Groza

Installation, Setup and Usage

- **Documentation:** The instructions, while comprehensive, required a high degree of technical proficiency, making the setup challenging for users without extensive programming or machine learning background.
- **Complexity:** The installation process was complex and very tedious, involving multiple steps such as downloading different components (models, tokenizer, scripts) that require large amounts of space (> 2GB) and setting up the environment. On top of that, the code presented typos and nonconformities with the documentation (i.e. the source code presented old hardcoded values of files and methods that had their names modified by the time of the latest version release). One major error is given by the fact that the model .ckpt files were split in batches, but processed as a whole unit in the source code. All of these aspects forced us as users to heavily interact with the codebase and even modify the logic and hardcoded values in the implementation.
- **Dependency Issues:** Encountered issues with dependencies like tree-sitter and javalang, requiring modifying the implementation and additional installations because the tool has not been updated in years.
- **Compatibility Issues:** Faced compatibility problems with the my-languages.so file (developed on a non-MacOS machine, resulting in OS compatibility issues for MacOS users), needing detailed guidance on recompiling language grammars. This involved cloning the official tree-sitter Java repository, and writing a C script on our machine in order to generate a new compatible my-languages.so file.

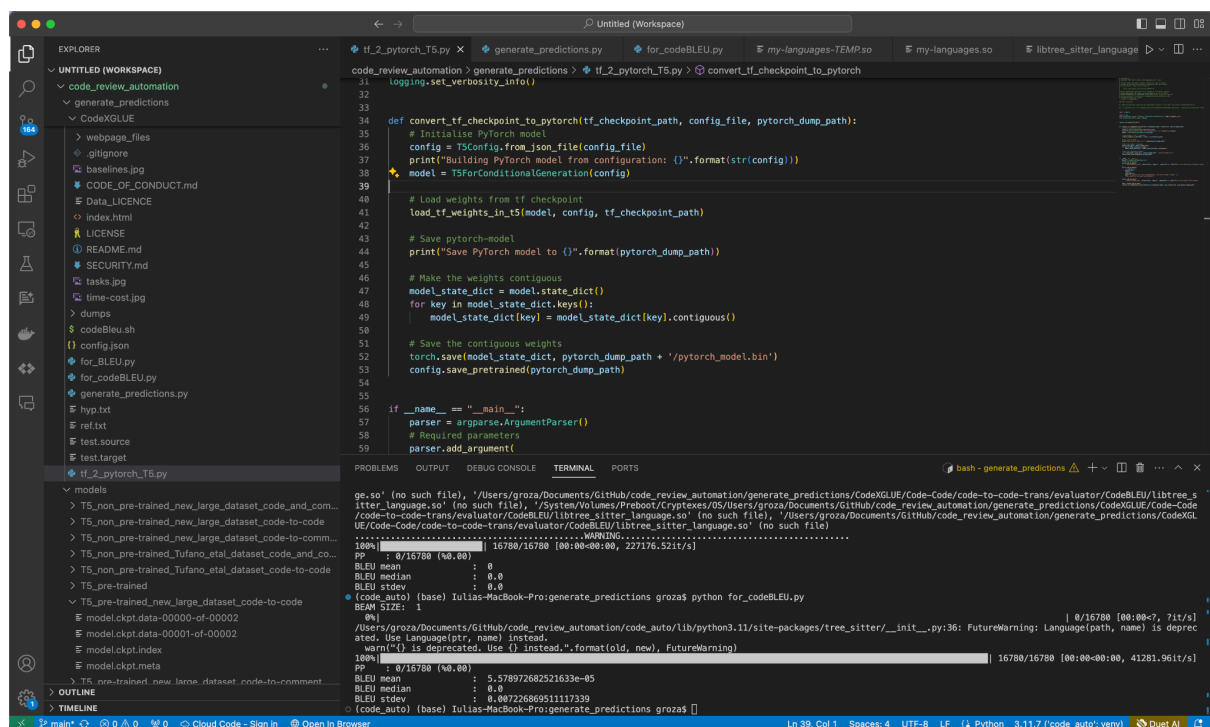
Usage

- **Data Preparation:** Preparing a suitable dataset was essential. This involved gathering examples of pre- and post-review code, which was then formatted appropriately for the tool.
- **Model Execution:** Running the model on the dataset involved executing scripts like generate_predictions.py and for_codeBLEU.py.
- **Evaluation Challenges:** Encountered errors during the evaluation phase, particularly with the for_codeBLEU.py script, which required troubleshooting around tree-sitter grammar compilation and linking.

Benefits and Results

- **Automated Review:** The tool offers the potential for automated code review, which can suggest modifications or highlight areas needing attention.
- **BLEU Score Evaluation:** The tool's effectiveness was evaluated using BLEU scores. However, the initial scores were low, suggesting the model's predictions had limited alignment with the target code. Its current performance, as indicated by the low BLEU scores, suggests that additional fine-tuning and dataset optimization are necessary for more accurate and relevant results in a real-world application.
- **Manual Review Necessary:** The current state of the tool suggests a supplementary role to human code review rather than a replacement. The tool, while promising in automating code review tasks, presents significant installation and operational challenges that require advanced technical skills to overcome.

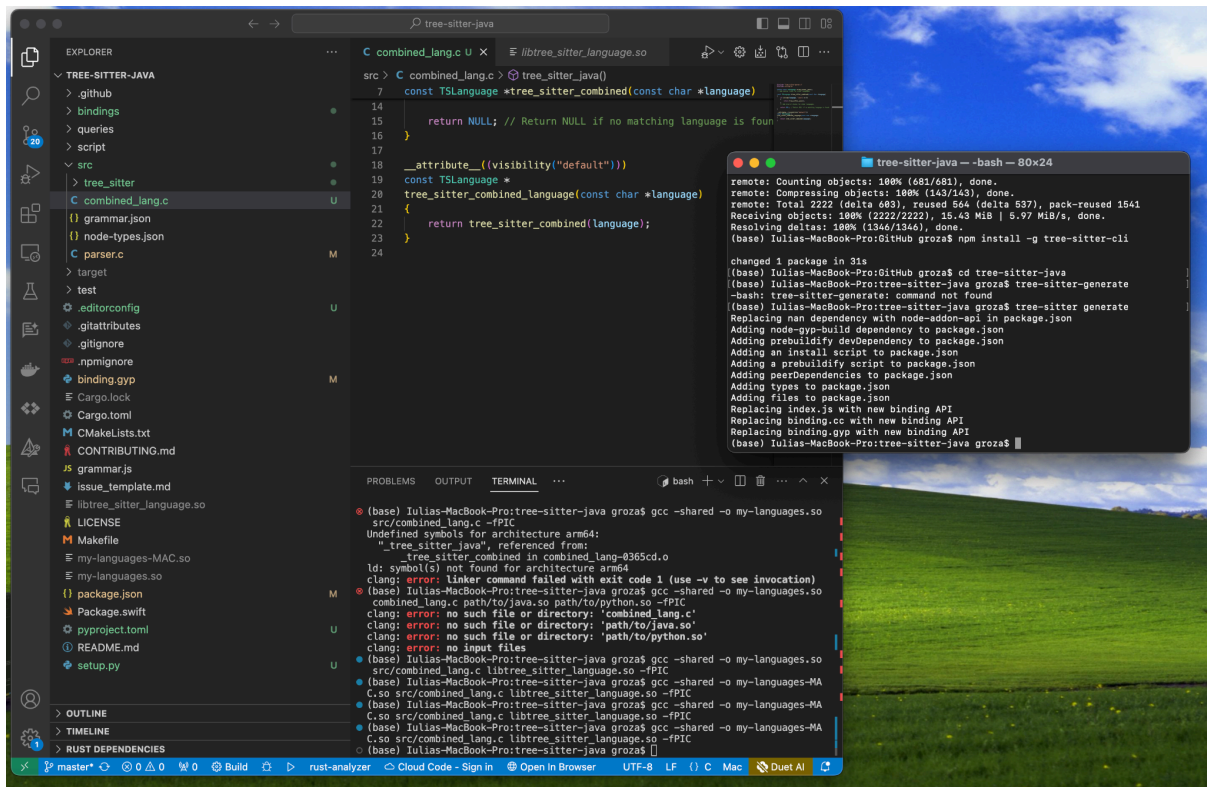
Screenshots



The screenshot displays a Visual Studio Code workspace with a Python script and a terminal window. The script, `tf_2_pytorch_T5.py`, defines a function `convert_tf_checkpoint_to_pytorch` that takes a TensorFlow checkpoint path, a config file, and a PyTorch dump path as arguments. It initializes a PyTorch model, loads weights from the TensorFlow checkpoint, and saves the resulting PyTorch model. The terminal window shows the output of the `generate_predictions` command, which includes BLEU scores for various models and a warning about deprecated language features.

```
code_review_automation > generate_predictions > tf_2_pytorch_T5.py > convert_tf_checkpoint_to_pytorch

31 logging.set_verbosity_info()
32
33
34 def convert_tf_checkpoint_to_pytorch(tf_checkpoint_path, config_file, pytorch_dump_path):
35     # Initialise PyTorch model
36     config = T5Config.from_json_file(config_file)
37     print('Building PyTorch model from configuration: {}'.format(str(config)))
38     model = T5ForConditionalGeneration(config)
39
40     # Load weights from tf checkpoint
41     load_tf_weights_in_t5(model, tf_checkpoint_path)
42
43     # Save pytorch-model
44     print("Save PyTorch model to {}".format(pytorch_dump_path))
45
46     # Make the weights contiguous
47     model_state_dict = model.state_dict()
48     for key in model.state_dict.keys():
49         model_state_dict[key] = model_state_dict[key].contiguous()
50
51     # Save the contiguous weights
52     torch.save(model_state_dict, pytorch_dump_path + 'pytorch_model.bin')
53     config.save_pretrained(pytorch_dump_path)
54
55
56 if __name__ == "__main__":
57     parser = argparse.ArgumentParser()
58     # Required parameters
59     parser.add_argument(
60
61 ge.so' (no such file), '/Users/groza/Documents/GitHub/code_review_automation/generate_predictions/CodeXGLUE/Code-Code/code-to-code-trans/evaluator/CodeBLEU/libtree_sitter_language.so' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/Users/groza/Documents/GitHub/code_review_automation/generate_predictions/CodeXGLUE/Code-Code/code-to-code-trans/evaluator/CodeBLEU/libtree_sitter_language.so' (no such file), '/Users/groza/Documents/GitHub/code_review_automation/generate_predictions/CodeXGLUE/Code-Code/code-to-code-trans/evaluator/CodeBLEU/libtree_sitter_language.so' (no such file)
62 .....WARNING.....
63 .....
64 PP : 0/16788 (%0.00)
65 BLEU mean : 0
66 BLEU median : 0.0
67 BLEU stdev : 0.0
68 (code_auto) (base) Iulias-MacBook-Pro:generate_predictions groza$ python for_codeBLEU.py
69 0%
70 /Users/groza/Documents/GitHub/code_review_automation/code_auto/lib/python3.11/site-packages/tree_sitter/__init__.py:36: FutureWarning: Language(path, name) is deprecated. Use Language(path, name) instead.
71 warn("{} is deprecated. Use {} instead.".format(old, new), FutureWarning)
72 PP : 0/16788 (%0.00)
73 BLEU mean : 5.578972682521633e-05
74 BLEU median : 0.0
75 BLEU stdev : 0.00726869511117339
76 (code_auto) (base) Iulias-MacBook-Pro:generate_predictions groza$
```



```
(code_auto) (base) Iulias-MacBook-Pro:code_review_automation groza$ python3 ./generate_predictions/tf_2_pytorch.T5.py --tf_checkpoint_path ./models/T5_pre-trained_n
ew_large_dataset_code_and_comment-to-code/model.ckpt --config_file ./generate_predictions/config.json --pytorch_dump_path ./generate_predictions/dumps
Building PyTorch model from configuration: T5Config {
  "architectures": [
    "T5ForConditionalGeneration"
  ],
  "classifier_dropout": 0.0,
  "d_ff": 2048,
  "d_kv": 64,
  "d_model": 512,
  "decoder_start_token_id": 0,
  "dense_act_fn": "relu",
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "relu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "is_gated_act": false,
  "layer_norm_epsilon": 1e-06,
  "model_type": "t5",
  "n_positions": 512,
  "num_decoder_layers": 6,
  "num_heads": 8,
  "num_layers": 6,
  "output_past": true,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
  "transformers_version": "4.39.2",
  "use_cache": true,
  "vocab_size": 32128
}
Generate config GenerationConfig {
  "decoder_start_token_id": 0,
  "eos_token_id": 1,
  "pad_token_id": 0
}
```

```
(code_auto) (base) Iulias-MacBook-Pro:generate_predictions groza$ python for_codeBLEU.py
BEAM SIZE: 1
0% | 0/16780 [00:00<?, ?it/s]
/Users/groza/Documents/GitHub/code_review_automation/code_auto/lib/python3.11/site-packages/tree_sitter/_init__.py:36: FutureWarning: Language(path, name) is deprec
ated. Use Language(ptr, name) instead.
warn("{} is deprecated. Use {} instead.".format(old, new), FutureWarning)
100% | 16780/16780 [00:00<00:00, 41281.96it/s]
PP : 0/16780 (%0.00)
BLEU mean : 5.578972682521633e-05
BLEU median : 0.0
BLEU stdev : 0.007226869511117339
```

```
45 beam_size = 1
46 batch_size = 64
47 task = 'code&comment2code: ' # possible options: 'code2code: ', 'code&comment2code: ', 'code2comment: '
48 data_dir = "../dataset/fine-tuning/new_large/code&comment-to-code/" # change the path if needed
49 tokenizer_name = "../tokenizer/TokenizerModel.model"
50 model_name_or_path = "../dumps/pytorch_model.bin"
51 config_name = "./config.json"
```