ASDL Project

Fixing syntactically incorrect code with Deep Learning

Valentin Knappich

July 3, 2021

University of Stuttgart, Analyzing Software using Deep Learning, Prof. Michael Pradel

Agenda

1. Preprocessing
2. Modeling & Training
3. Results

Preprocessing

Tokenization

- `tokenize` package
- Error handling needed for incorrect code:
  - `TokenError`
    - Thrown at the end of the sequence $\rightarrow$no tokens lost
  - `IndentationError`
    - Sometimes thrown before the end of sequence $\rightarrow$tokens lost
    - Advantage for the model
    - Occurs only 132 times in the whole dataset (50000 samples)

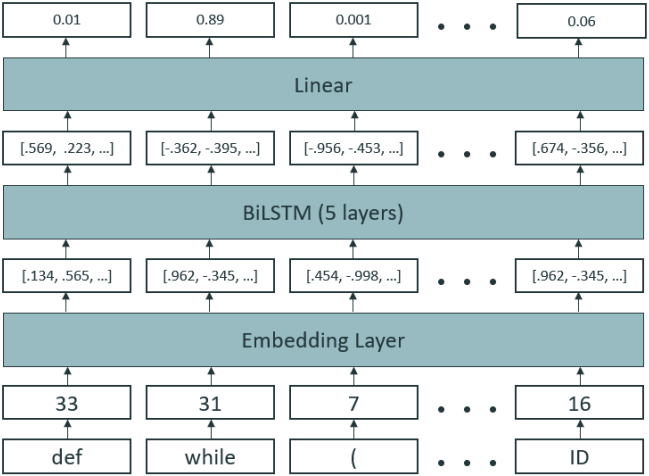
Testing Preprocessing

- Testing tokenization via reconstruction
  - Run tokenization
  - Convert character index to token index
  - Convert token index back to character index

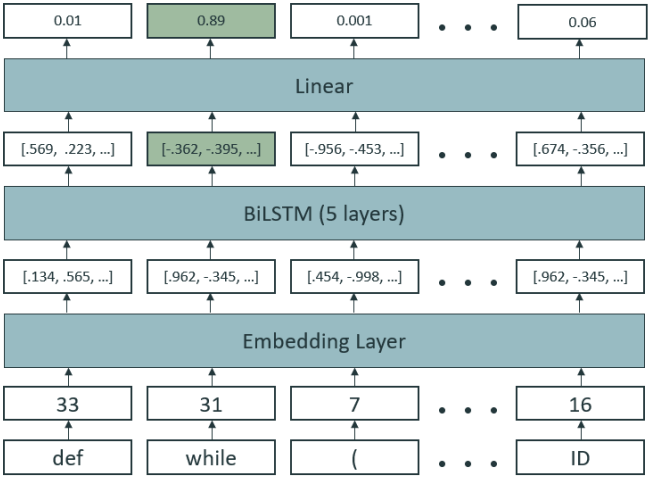
  → Misalignments?
- Walrus operator `:=` (fails 85 times in 50000 samples)
  - Was introduced in Python 3.8
  - `":=="` gives `[":=", "="]` instead of `[":", "=="]`
- Decorator `@` (fails 4 times in 50000 samples)
  - `"@=="` gives `["@=", "="]` instead of `["@", "=="]`

Modeling & Training

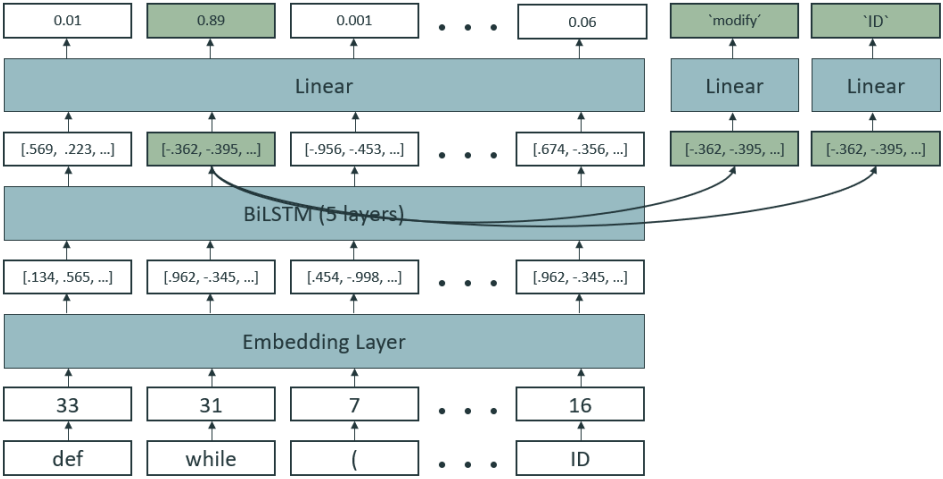
Architecture



Architecture



Architecture



Training

- `Adam` optimizer and `CrossEntropyLoss`
- Multi-task training (adding losses together)
- Problem: As long as the location prediction is bad, the signal from type and token prediction are just irritating
- Solution: Use linear loss weighting schedule
  - decrease location loss weight
  - increase type and token loss weight

```
location_weight = torch.tensor([-(x + 1) / n_epochs + 1
                                for x in range(n_epochs)])
type_weight     = torch.tensor([(x + 1) / n_epochs
                                for x in range(n_epochs)])
token_weight    = torch.tensor([(x + 1) / n_epochs
                                for x in range(n_epochs)])
```

Results

Results

- Results vary depending on random initialization and random test split
- Training on single or whole dataset implemented (single / whole)
- Evaluating on test set once per epoch
  - Location Accuracy: $85 - 95\%$ / $90 - 95\%$ (single / whole)
  - Fix Type Accuracy: $65 - 75\%$ / $80 - 85\%$ (single / whole)
  - Fix Token Accuracy: $55 - 65\%$ / $70 - 80\%$ (single / whole)
- Prediction
  - Fraction of corrected code snippets: $\approx 60\%$ / $\approx 80\%$ (single / whole)

Questions