

Progress Report 2

1. Changes
 - a. No major changes, though I will probably stick to a minimal set of model comparisons (e.g. using one set of parameters) given time constraints.
2. Research questions
 - a. Does incorporating AST information in a Transformer model improve performance on standard metrics for documentation generation compared to a Transformer using only textual information (BLEU, METEOR, ROUGE)?
 - b. How does the performance of the baseline Transformer for code summarization model change when training on a dataset without data leakage?
3. Challenges
 - a. I am having some issues with using the 150k dataset with the model library I'm using as the baseline, but I haven't had much time to debug. I do think it's some sort of user error on my part with parsing the inputs since it's a text-only model and shouldn't require any modifications to the underlying model.
4. Demo
 - a. I can demonstrate commands to use the library, but the models themselves probably won't be able to run in a 5-minute demo. I could potentially use a very small demo dataset for this, though I'm not sure how much value there is in just running the model. I can also compare specific examples of model generated and human written documentation and potentially model vs. model results.
 - b. Sample demo
 - i. Data preparation and necessary commands
 1. I will assume the user has access to a dataset of similar format as the Python 150k dataset (ASTs extracted)
 - ii. Model training
 - iii. Model prediction
 - iv. Example on a specific function + comment
5. Deliverables (FROM PROGRESS REPORT 1)
 - a. Written work:
 - i. Courseworks
 1. This should include a table comparing performance on standard metrics (ROUGE, METEOR, etc.) for at least one variant of the text-only Transformer and one variant of the AST-augmented Transformer. Other variants include:
 - a. AST representation variants
 - b. Parameter counts/sequence lengths
 - b. Code:
 - i. github, ideally structured and documented so that using it is as easy as a model imported from some standard lib (e.g. pytorch, hugging face). I liked that CodeBERT, which I used for my midterm paper, once installed, was able to fine-tune and run using 2 (very long) lines in the terminal. I'd like to provide a similar experience (though I won't be offering a pretrained model).
 - c. Dataset:

- i. github for dataset extraction code
- ii. potentially zenodo for the actual dataset if I end up doing anything noteworthy to it, otherwise it is an existing dataset

6. Resources (FROM PROGRESS REPORT 1)

- a. Dataset:
 - i. Python 150k: <https://eth-sri.github.io/py150>
- b. Code:
 - i. Ahmad et al. 2020 model and related code:
<https://github.com/wasiahmad/NeuralCodeSum>