01/13/2020 - 01/27/2020

E E/Cpr E/S E 492 Weekly Report 1

Intelligent Code Editor

Client & Advisor: Ali Jannesari

sdmay20-46

John Jago – Software Lead

Keaton Johnson – Systems Lead

Jon Novak – Machine Learning Lead

Matthew Orth – Meeting Facilitator

Garet Phelps – Report Manager

Isaac Spanier – Test Lead

Weekly Summary

This first two weeks were mostly getting back in the groove after the hiatus we took over break. We established what things we need to work on, and broke into groups to get started with the next sprint.

Past week accomplishments

John Jago

- Research for custom POS tagging
    - Investigated how well existing parts-of-speech taggers work on English statements with a lot of programming terminology
        - For the most part, the tagging is correct
        - Need to follow up to understand why we might need to do additional work for the words in the programming domain
- Custom POS tagger
    - NLTK allows the existing taggers to be extended
    - Created a Python script to demonstrate a custom unigram tagger that considers one word at a time
        - This can be incorporated into the preprocessing to improve the tagging of terms typically used when describing method invocations

Keaton Johnson

- Looked into methods of data mining to create a method invocation dataset.
    - Looked into using C#/.NET Core application to generate data.
    - Started looking at how to parse Github Repos.

Jon Novak

- Looking into ways to extend the dataset to invoke methods.

Matthew Orth

- OpenNMT-py <unk> token issue research
    - Posted question on OpenNMT forum to determine potential solutions: http://forum.opennmt.net/t/unk-token-in-target-file-during-translation/3271/3
    - Conducted update of OpenNMT on GPU server to confirm previous versions did not cause this issue
    - Considered dataset construction and how that could affect the <unk> token results
- Stack Overflow API Research
    - Learned about searching and filtering methods to obtain relevant questions and answers for automatic dataset generation
        - This issue will be addressed when implementing automatic dataset generation
- Preprocessing Research:
    - Brainstormed potential methods of preprocessing using NLTK
        - Segmentation, tagging, summarizing
    - Determined how GloVe can be used to find the similarity between multiple words
    - Researched more in depth about AnyCode's implementation
- System Design Research:
    - Completed an in-depth review of AnyCode and what aspects we could use in our system
    - Outlined a system design that preprocessed the input and natural language statement into Verb, Noun format where each of the parameters are converted to their type

Garet Phelps

- Researched POS tagging

Isaac Spanier

- OpenNMT-py <unk> token issue research
    - Running script to discern the actual error
- Researching the Noun Verb Preprocessing

Individual contributions

Name	**Contributions**	**Hours this week**	**Hours cumulative**

John Jago	● Research for custom POS tagging ● Custom POS tagger	3	41
Keaton Johnson	● Looked into methods of data mining to create a method invocation dataset.	2	28
Jon Novak	● Looking into ways to extend the dataset to invoke methods.	2	29
Matthew Orth	● OpenNMT-py <unk> token issue research ● StackOverflow API Research ● Preprocessing Research ● System Design Research	15	102
Garet Phelps	● Researched custom POS tagging	2	21
Isaac Spanier	● OpenNMT-py <unk> token research ● Noun Verb Preprossing	2	26

Plans for the upcoming week

John Jago

- Research Creating or Modifying a POS Tagger for Programming Domain

Keaton Johnson

- Mine Java Method Invocation from GitHub

Jon Novak

- Mine Java Method Invocation from GitHub

Matthew Orth

- Research Verb-Noun Preprocessing in Programming Domain

Garet Phelps:

- Research Creating or Modifying a POS Tagger for Programming Domain

Isaac Spanier

- Research Verb-Noun Preprocessing in Programming Domain

Summary of weekly client/advisor meeting

Meeting with Prof. Jannesari on 2020-01-23 at 2:00 pm

During our first meeting this semester, we discussed feedback given by the faculty panel at the final presentation last semester. A few takeaways were that we need to be more specific with the scope of the project so that it isn't too small (only print statement translations) or too broad (translating any arbitrary English into working Java code) and that we need to compare our results to some existing baseline before making claims of better translation accuracy. We decided to focus on method invocations, as we had mentioned at various times last semester, and we also agreed that the user of our translation tool should know programming concepts. Three initial steps were identified: researching how nouns and verbs are used when describing actions in programming, creating or modifying a part-of-speech tagger to include words from the programming domain, and mining method invocation samples.