Problem

Programming languages that use compile-time type checking, like OCaml, often give poor error messages that make debugging hard (especially for novice programmers).

Example:

To find length of a list, count the number of times we remove an element until we get an empty list

```
let rec lenlist xs =
  match xs with
  | [] -> []
  | y::ys -> 1 + lenlist ys;
```

**Error:** This expression has type 'a list but an expression was expected of type int.

- **What the compiler thinks is wrong:**
  - The program is attempting to add an int type with a list type.
  - However, the program thinks the list should be an int type.

- **What is actually wrong:**
  - The base case returns an empty list instead of an int type, 0.

Why the difference?

- The OCaml type checker inspects for valid types before a program is executed.
- As a result, when an error is found, the compiler displays the first discrepancy between the inferred type and expected type.

Approach & Design

Given broken code and its error location determined by NATE, we create another tool called RITE, a machine learning program that classifies an error into a certain clusters and determines what kind of error it is. This information allows RITE to also generate a template fix, which we apply to the broken code.

```
let rec lenlist xs =
  match xs with
  | [] -> []
  | y::ys -> 1 + lenlist ys;
```

Error Localization

- Dataset from UCSD’s undergraduate Programming Languages Course, contains pairs of student’s broken and fixed code.
- Utilize NATE to provide the location of the error.
- RITE clusters the training set of data to establish different “clusters” or groupings of errors.
- RITE uses machine learning to determine which “cluster” best represents the input error.
- A fix is generated to fit the context of the code.

NATE

Lawrence D. Lee, Vignesh Kodali, Gokul J. Shenoi, Jhala Y. Jhala

Evaluation/Conclusion

We can evaluate the accuracy of RITE’s approach in three ways.

- **(1) Does it fix the right error?** Evaluate how well RITE correctly identifies the error.
- **(2) Does it compile?**
- **(3) Does it work as intended?** Evaluate what percentage of RITE's suggested fix matches the suggested fix in our data set.

The graph below demonstrates (1), testing with a different number of clusters. However, evaluating (2) and (3) has been delegated to future work.

Future Work

Our next steps are to check whether our proposed fixes compile and run successfully. We will also compare our results from RITE to Seminal, an older research project that uses heuristics instead of machine learning to generate fixes.

We hope that RITE can help novice OCaml programmers successfully find and fix their type errors.

Acknowledgements

We would like to thank Christine Alvarado and Vignesh Kodali for leading ERSP and providing guidance and support throughout this year-long project.