

CS264 Homework 4 - ReAct Agent for Software Engineering

My final results were 8 resolved instances. The list of instances I passed were the following:
"django_django-10973", "django_django-11179", "django_django-13810",
"django_django-7530", "scikit-learn_scikit-learn-26323", "sphinx-doc_sphinx-9230",
"sympy_sympy-17655" and "sympy_sympy-24213"

I found that django, scikit and sympy instances had more successes than sphinx-dox, astropy, pytest and psf instances. My guess is that the former tools were used more often in industry/academia, so end up in a larger proportion of training data for the models, meaning that the models are more accurately able to identify the issues in the instances and resolve them. This makes sense as django is a common web framework and scikit is a common library for machine learning, while I haven't heard of the remaining instance frameworks.

The parser works by looking at the structure of the response format given in the comment at the top of the file, in the order given. The parser then looks for the indices of the tokens and uses that to split the response into its individual components and returns it as a dictionary for each of the portions(thought, name and arguments).

For the most part, the [agent.py](#) methods are fairly straight forward. The run method is the most complex, taking in the task in natural language and repeatedly prompts the LLM with the updated query and working/not working status. I did not spend as much time as I would have liked on this assignment, but I thought it was a very fun and interested homework.