10-608: Assignment 2

Igor Labutov, Bishan Yang

January 2018

Semantic Parsing II: Email Agent

In the first assignment, you learned to build a grammar for a grammar-based semantic parser. In class, we have started building a simple email client that can be trained with examples. In this problem, your task is to complete a simple a email client that has the following functionality:

- compose a new email draft
- set the subject to some string
- set the body to some string
- set the recipient to some email address
- send the email draft
- delete the email draft

You can continue this implementation based on what we've seen in class. For this assignment, you need to do the following in SEMPRE:

- define your own grammar that can parse five of the commands listed above
- create a small dataset to ensure that your semantic parser can parse these commands, and train the semantic parser
- on every command, the email client should respond with what it's doing, e.g., *OK, I set the subject to ....*
- implement basic functionality for your email:
  - if no email is being composed, and the user tries to send it or set one of its fields, the agent should respond with some reasonable error message (e.g, *"there is no email for me to work with"*
  - if the recipient field is missing, the agent should respond with something like *"no recipient given in the email"* or something similar.
  - if the user is already composing an email, and tries to compose a new email, the agent should respond with something like *"either send or delete this draft before drafting a new one"*

You are free to decide how you will write the grammar, and the exact responses of the agent. You should aim to allow at least several phrasings for the same command to allow the user to be somewhat natural. For example, like in class, you should also aim to allow the grammar to be flexible with ignoring filler words, such as those before or after the command. In general, you have a lot of creative freedom in how to build this agent – so make it fun!

What to submit:

- the code that implements the email functions (i.e., the thing you would put in your `executors` package like we did in class)
- a sample sequence of interactions with the agent that illustrates the required functionality

Semantic Parsing III: End-to-end semantic parsing with neural networks

In this problem, your task is to build a simple sequence to sequence semantic parser to parse calculator commands (e.g., "add one to three then multiply by two"). You may re-use the pyTorch code given in class to adapt it to this problem. Your task is to train on the `calculator.dataset` to build a simple sequence to sequence model with attention (but without a copying mechanism). Please split the dataset randomly using 20% of the data held out for testing.

What to submit:

- all of your code (zip it up or store the iPython notebook).
- your training and testing accuracy.