PNet - ParseNet

Progress Report

Sai Krishna Karanam karanam.s@husky.neu.edu
Nischal Mahaveer Chand mahaveerchand.n@husky.neu.edu
Varun Sundar Rabindranath rabindranath.v@husky.neu.edu

I. Changes

We have also changed the datasets used for our task. As mentioned in the Edinburd paper (cite), the Hearthstone and Djanjo datasets, are not generalizable and would not be the right data to use for the task (Discussed more elaborately in Section X). As a result, we decided to scrape our own data from the repositories mentioned in the Edinburd paper.

II. Project Plan

Due to the important task of gathering our own data for the task, the data creation and processing phase of our original plan has taken up significant time, resulting in the following new plan.

1) Phase 1: Literature Survey [Done]
2) Phase 2: Data Preprocessing [In-progress]
3) Phase 3: Model training [Future work]

III. Data

A. Problems with existing datasets

The comments in the Django dataset are human annotated to explicitly describe what the following line of code is. Because of this, the BLEU scores reported by the research community on the Django dataset is high. However, this is not the case in general production environment and the dataset is less useful for our task of converting natural language comments to code.

Marceli Barone et. al. (cite) also critiques about the django dataset on similar lines.

Consider the following example from the Django dataset:

```
comment:
  raise and exception InvalidCacheBackendError
  with string "Could not find config for '\%s'
  in settings.CACHES" as argument , replace '
  \%s ' with alias .

code:
  araise InvalidCacheBackendError ( "Could not
  find config for '\%s' in settings.CACHES" \%
  alias )
```

Example 1

As we can see, programmers are very unlikely to write such elaborate comments, and learning from such input pairs, while making it feasible to achieve high BLEU and accuracy scores, is not very useful for automatic code generation in production environments

B. Data gathering

Scrape from web.

C. Data preprocessing

processing...

IV. Methods

The two methods used:

1) Base paper placeholder
2) Vanilla seq2seq: To further illustrate the problems with HS and DJ data, we train a vanilla seq2seq model in TensorFlow. Google separately provides a high-level API for the same at google/seq2seq [?], under the Apache 2.0 license. We run the same model without modification.
   As TensorFlow provides the ability to see the training process via TensorBoard, we stopped model training when we saw no visible improvement on the dev corpus. The model trained was a BiDirectional LSTM model, with attention as described in [?].

V. Experiments

DO you even experiment?

VI. Results

End of humanity is near.

VII. Future Word

Our current focus is on following two major tasks:

A. Adding context to data

We plan to add context to the data

B. Creation of dynamic top-down recursive networks

Not sure if we should do this.