Neural Code Summarization: Experiments in Bash and Python

Benjamin N. Peloquin

Introduction

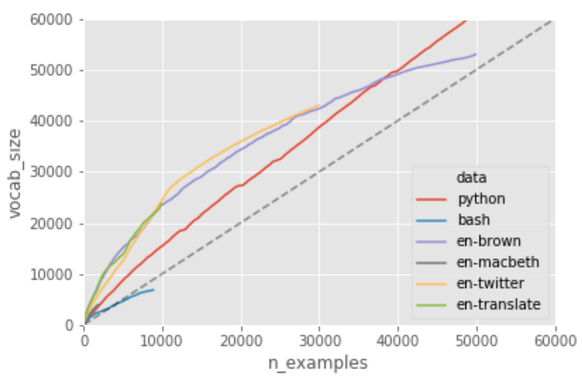
Code summarization *(CS), the task of generating natural language descriptions of code, has clear applications in domains such as code search, automated documentation, and programming pedagogy. Previous work has focused on the reverse task (program synthesis) and few projects have explored end-to-end neural approaches to CS. We cast the problem in terms of neural machine translation. We gain traction on a Bash dataset and explore problems on a more difficult Python dataset. We find character-level modeling appears particularly important in this domain.*

Models

* ***Token-level LSTM encoder/decoder*** *models*
  * *Tuning on validation set alpha=1e-04, dropout=0.5*
  * *Learned token/character embeddings from scratch*
* ***Character-level LSTM encoder/decoder*** *models*
  * *Tuning on validation set alpha=1e-04, dropout=0.5*
  * *Learned token/character embeddings from scratch*
* ***Token-level Transformer*** *model*
  * *layers=6, heads=8*
  * *Default query, key, value projection dims as in Vaswani et al. 2017*
  * *Learned token/character embeddings from scratch*

Data complexity and scaling

Python dataset displays linear growth in vocabulary size unlike natural language datasets — this makes the problem particularly difficult.



BASH

Bash *is a Unix Shell and command language used as the default login shell for most Linux distributions as well as Apple's macOS. Shell commands typically consist of three components -- a utility (e.g. find, grep), optional flags (e.g. -name, -i), and arguments (e.g. "*.java", "TODO").*

Dataset *from Lin et al. (2018)*
* *9,305 Bash script / English description pairs*
* *102 unique Bash utilities (find, grep, etc)*
* *206 option flags*

Example source / target pairs

```
In [3]: grep -l "TODO" *.java
```

Find .java files in the current directory tree that contain the pattern “TODO” and print their names

```
In [3]: find . -type f  sort -nk 5,5 | tail -5
```

Display the 5 largest files in the current directory and its subdirectories.


```
find . -name '*.pyc ' -print0 | xargs -0 rm
```

transformer-token: Delete all <unk> files under current directory
lstm-token: Remove all <unk> files from the current directory tree
lstm-character: Find all *.pyc files under current directory tree
gold: Recursively removes all files like '*.pyc ' in a current folder

python

Python *is a high-level general purpose programming language which is distinctive for its readability and use of white-space.*

Data *is a recently released corpus of python function/ docstring pairs.*
* *1.2 million docstring/function pairs*
* *310,092 pairs after pre-processing*

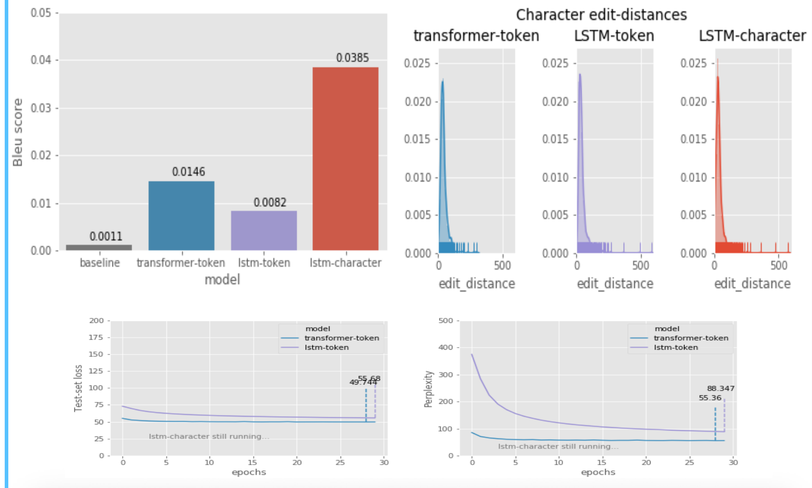
Example source / target pairs

```
def enabled():
    return _editor \
        if value_to_boolean(_editor) \
        else _editor.get_value(_extension_config)
```

Gets or sets a boolean value that describes if the extension is enabled

```
def pupil_size():
    return 19
```

Returns dummy pupil size




```
def default_transcript(self, url):
    self.language_code = url
    return self
```

transformer-token: returns the url for the given url
lstm-token: return a list of the data
lstm-character: returns the default transcript
gold: download default transcript from a video platform