Fifth Milestone Report: Non-parametric Language Models for Code Generation and Summarization

Alex Xie
`https://axie66.github.io/07400-project/`

Mentored by Vincent Hellendoorn
Institute for Software Research

March 21, 2022

1 Major Changes

While the initial focus of my project was code generation from text, I have chosen to also explore the task of code summarization (i.e. generating documentation for a function). This task is very closely linked to code generation; switching between the tasks is simply a matter of swapping the source and target sequences. Preliminary results on code summarization using kNN retrieval show promise, in some cases yielding significant improvements over the base model.

2 Progress Report

2.1 Accomplishments

Language	w/o kNN	w/ kNN
Ruby	16.01	16.04
Javascript	16.24	17.92
Go	19.74	19.46
Python	19.43	19.46
Java	20.29	20.73
PHP	26.23	26.59
Average	19.66	20.03

Table 1: BLEU results for multilingual summarization experiments on CodeSearchNet

Taking advantage of the fact that the CodeSearchNet dataset contains examples in multiple programming langauges, I fine-tuned CodeT5 as a "multilingual" code summarization model (i.e. mapping from multiple programming languages to English); this trained model was then used as the base model in various kNN retrieval experiments, the results of which are shown in Table 1. Across five of the six languages, kNN yields improvements with minimal tuning; the gains are particularly large on Javascript. Contrary to intuition, on the two highest resource languages, Go and Python, kNN provides relatively little benefit - I plan on looking further into this.

Meanwhile, the poor performance of kNN on Ruby is understandable given that it is the lowest resource language. Inspired by kNN-MT [1], I augmented the Ruby kNN datastore with summarization examples from other programming languages. This is made possible by the multilingual nature of the model – we assume that similar code snipppets in both Ruby and, say, Python are mapped into the same space, allowing retrieved Python examples to aid in Ruby summarization. The results, shown in Table 2, validate this assumption to some extent. Interestingly, however, combining all datastores slightly hurts performance, perhaps due to excessive noise or because certain representations fail to generalize across languages.

	BLEU	Datastore size
Base Model	16.01	-
+ Ruby datastore	16.04	391270
+ Python datastore	16.24	2199034
+ Java datastore	16.31	3548944
+ all datastores	16.30	7474138

Table 2: Results of expanding the Ruby kNN datastore with examples from other PLs

2.2 Previous Milestone Goals

Due to the shift in focus, I have not devoted quite as much time to my previous experiments on code generation. However, I have continued work on some of my goals from the last milestone; I found kNN yielded marginal improvements for code generation on CodeSearchNet (2.4 to 2.5 BLEU), though the results are not particularly interesting given the model's exceedingly poor performance. I have also shelved the CoDesc experiments described in my last milestone goals for the time being, as running experiments on the dataset is computationally expensive due to its extremely large size.

2.3 Surprises

The main surprise is the relatively promising performance of kNN on code summarization, as described in the previous section.

3 Next Steps

3.1 Looking Ahead

Moving forward, I plan to further investigate the idea of improving performance on low-resource programming languages by augmenting the kNN datastore. One possible improvement, taken from machine translation [2] [3], might be to train a discriminator that attempts to discern the language of the input from the kNN representation; this would ensure that the model's representations are language-invariant, allowing for the model to make better use of other languages' datastores.

Another interesting direction might be to use kNN as a "guide" to allow for summarization in a zero-shot setting, i.e. where a model trained to summarize Python could then be expected to summarize Ruby, perhaps given only Ruby source code without corresponding natural language summaries.

3.2 Revisions to Future Milestones

My plans for the next milestones are largely dependent on the results of the experiments described in the previous sections; however, in general, I plan to obtain more detailed results on code summarization while also in parallel continuing previous work on code generation.

References

[1] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] Baptiste Roziere, Marie-Anne Lachaux, Lowik Chanussot, and Guillaume Lample. Unsupervised translation of programming languages. *Advances in Neural Information Processing Systems*, 33, 2020.