Reimplementation of *Learning Python Code Suggestion with a Sparse Pointer Network*

Samuel Müller (sgm48)

February 22, 2019

1 Overview

Many software developers use IDEs, a feature rich editor specialized for programming in particular programming languages, for their work. One of the popular features of IDEs is code completion. Code completion lets the IDE propose what comes next in the code, this can make a developers workflow much more efficient, since she does not have to type very obvious thing herself. A particular subtask of code completion is the prediction of the next token in code, opposed to predicting the next characters on a subtoken level. In this project we aim to implement the Sparse Pointer Network (SPN) (Bhoopchand et al. 2016) model that learns to solve this problem by training on a large corpus of code, just as Bhoopchand et al. (2016) did.
Current approaches to this problem usually work by parsing the code, including imported libraries, for possible next tokens that fit into the next position. While this might work most of the time, especially for untyped languages like Python, the language trained on by Bhoopchand et al. (2016), this might fail due to having too many possible next tokens.
An approach of previous work to combat this was to lend methods from language modeling (LM) for this task, including LSTMs and n-grams. These on the other hand struggle a lot with the prediction of identifiers, because identifiers are very unique to a particular file and are not used the same way across files: Two very different objects might have the same name.
In this project, we use a SPN as an alternative to LSTMs to increase performance on identifier prediction. A SPN has three parts, a copying mechanism that computes likelihoods for copying recent identifiers based on attention weights, a simple LSTM based language model using a fixed vocabulary and a mechanism to weight the two previous.
The model will be evaluated on the dataset provided in class. This dataset

will be split three-fold into training, validation and test set. The model will be tested in terms of accuracy, accuracy among the 5 highest ranking predictions and perplexity.

2 Plan

2.1 LM with Attention on Trivially Prepared Dataset (Until Session 4)

The target for session 4 is to have an implementation of a language model, that, different from the model implemented as our practical, uses an attention mechanism to predict the next token. The knowledge gained here can be used to better understand the implementation of the paper. This also is an interesting baseline that could be further explored. While the dataset used in the paper is preprocessed in a quite complex way, especially in differentiating between different kinds of identifiers based on where they are introduced and respecting the scope of identifiers for name replacement, my dataset preparation in this and the next week should not differentiate between the different kinds of identifiers, and just replace identifiers with tokens independent of what kind of identifier they are.

2.2 Reimplementation of a 'Sparse Pointer Network' on a Trivially Prepared Dataset (Until Session 5)

The goal for session 5 is to finish a reimplementation of the SPN model (Bhoopchand et al. 2016), based on the knowledge gained from the previous week.

2.3 Proper Dataset Preparation (Until Session 6)

In this week the plan is to come as close as possible to the way the dataset was prepared in the paper. There might be cases where this is hard, for a lot of identifiers in the dataset we use in class it is not totally clear which of the types of identifiers they belong to and I might need to come up with some tricks to figure that out.

2.4 Extensions and Experiments (Until Session 7)

I want to propose two extension ideas, to be finished as much as possible until session 7. Firstly, it might be interesting to try sharing an embedding between all identifiers (of some kind). This might lead to interesting results since there already is random embedding sharing between identifiers in the SPN model. To have the same identifier and still predict usable tokens would be possible in the SPN, since the addressing of the pointers works over the output of the LSTM and not the embeddings directly. Thus I think it might lead to similar results to just replace all the embeddings with a single embedding, or at least replace all embeddings of identifiers of one type with a single embedding. Another possibility would be to test the effects on the architecture when removing the filter on the SPN making it similar to a simple combination of a Pointer Network (Vinyals et al. 2015) and a standard LSTM (with attention) like proposed by See et al. (2017) for summarization. This simpler approach, which treats every token equally, was not introduced as a baseline in the original paper.

2.5 Risks

There are the following risks to the project's success.

- The preparation of the dataset in the same manner as Bhoopchand et al. (2016) might turn out to be very difficult. It is especially likely to be hard to replace the identifiers correctly. For the case that this fails, one could go with a simplified version, e.g. just ignoring scope in files and treating all variable kinds the same and assuming only one scope per file, or as a last resort use the python dataset published by Bhoopchand et al. (2016).

- The highest risks are associated to the extensions, since these are done at the end, so if something goes wrong before they are likely to be left away, and additionally they are very ambitious.

References

Bhoopchand, A., Rocktäschel, T., Barr, E. & Riedel, S. (2016), 'Learning python code suggestion with a sparse pointer network', *arXiv preprint arXiv:1611.08307* .

See, A., Liu, P. J. & Manning, C. D. (2017), 'Get to the point: Summarization with pointer-generator networks', *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* .
URL: *http://dx.doi.org/10.18653/v1/P17-1099*

Vinyals, O., Fortunato, M. & Jaitly, N. (2015), Pointer networks, *in* C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett, eds, 'Advances in Neural Information Processing Systems 28', Curran Associates, Inc., pp. 2692–2700.
URL: *http://papers.nips.cc/paper/5866-pointer-networks.pdf*