From Natural Tokens to Natural Trees in Source-Code

Bringing Structure to Naturalness: On the Naturalness of ASTs

PÂRȚACHI Profir-Petru, 杉山麿人

(国立情報学研究所)

どんな研究？

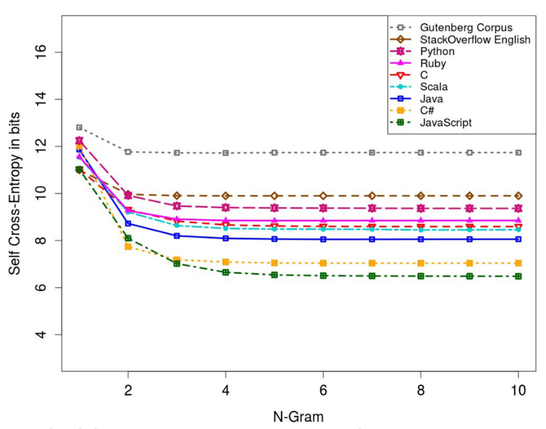
Previous research have shown source-code to be statistically predictable as a sequence of tokens. Is the same still true for structured views of source-code such as Abstract Syntax Trees?

何がわかる？

Employing a structured view of source-code can be beneficial and make the prediction task easier; however, it can also hinder by confusing the model.

背景・目的

Hindle et al.[2] showed that source-code at the token level is more predictable than standard English. It is also intuitive to assume that adding information can only simplify the prediction task, thus recent work employ tree or graph views of source-code. However, the hypothesis that such views should be “natural”, i.e. statistically predictable, has not been tested.



Self Cross-Entropy by language for n-gram models
Figure from Rahman et al.[3]

研究内容（方法・結果・結論）

方法：Self Cross-Entropy

In the original work by Hindle et al., they estimate self cross-entropy as:

$$H(D,M) = \frac{1}{n}\sum_{i=1}^{n} \log_2 \mathbb{P}(t_i|h(t_i))$$

where D is the dataset, M is the model, t_i is a token in the document, and $h(t_i)$ is a context around the token, such as other tokens just before or just after it.

In our work, we estimate the above probability as:

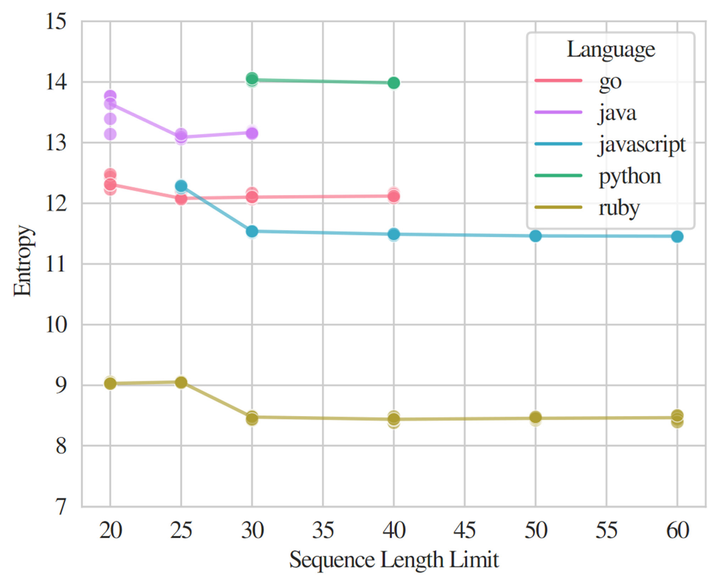
$$\mathbb{P}(t_i\,|t_1,\ldots,t_{i-1},t_{i+1},\ldots,t_n,\Delta\mathsf{T}\backslash\Delta LCA(t_i))$$

Where t_i is the *i-th* token, $\Delta\cdot$ is the sub-tree rooted at $\cdot$, T is the root of the AST, and *LCA(·)* is the least common ancestor operator.

We realise this via a TreeLSTM model trained using sub-tree masking, removing a token and the associated parse (sub-)trees.

結果：

Model performance improves similarly with context size, but (tree) structure is not always better.



While Ruby shows improvement over a raw token model, other languages are less predictable than their counterparts or indeed English.

This suggests that naively performing predictions at an Abstract Syntax Tree level may be more difficult than at a token level, although models do manage to learn.

[1] Pârțachi P-P, Sugiyama M, "Bringing Structure to Naturalness: On the Naturalness of ASTs", under revision.
[2] Hindle A, Barr ET, Gabel M, Su Z, Devanbu P. "On the naturalness of software", Communications of the ACM. 2016 Apr 26;59(5):122-31.
[3] Rahman M, Dharani P, Peter CR. "Natural software revisited." In 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), pp. 37-48. IEEE, 2019.