Learning of Python Source Code From Structure and Context

Adam Stecklov
ID
adamkutak@gmail.com

Ling Fei Zhang
260985358
lzhang133@gmail.com

Noah El Rimawi-Fine
260914339
noah.elrimawi-fine@mail.mcgill.ca

Abstract

In this paper, we introduce a neural model designed to encapsulate the semantics of Python functions and to predict function names based on code snippets. The core idea revolves around representing a code snippet as a fixed length vector embedding. This is achieved through a process involving the decomposition of code into an abstract syntax tree, extraction of path contexts, and feeding this set of path contexts to our encoder transformer. Then, the decoder processes this vector embedding and predicts a method name. We demonstrated the effectiveness of our approach by achieving x% accuracy on our testing set, which consists of 2500 samples.

1 Introduction

Vector representations of language, whether applied to individual words, sentences, or paragraphs, have greatly contributed to the works of natural language processing (NLP). For example, the distributed representation of words with "word2vec" showed that large language models (LLM) can learn the relationship between words with the concept of "king - man + queen = woman". However, we believe that the reach of LLMs should not be restricted to spoken language alone, but rather extended to all forms of language that convey actions, including coding languages. Specifically, our goal is to capture the semantic meaning of the Python language as actions.

Building upon the success of "word2vec", we want to distill Python functions into fixed length vectors, which encapsulates the semantic nuances of actions. In other words, we want to obtain fixed vector embeddings which represent the unique semantics of Python. For instance, for the two functions in Table 1, we would expect their vector embeddings to be similar.

It's worth noting that in our examples, we've given different functions and variable names. This is because, ideally, the neural model should be able to recognize actions, which is independent of variable or function names.

Function 1	**Function 2**

`def f(x)`
`  x = x + 10`
`  x = x**2`
`  return x`
	`def g(y)`
	`  y = y + 10`
	`  y = y**2`
	`  return y`

Table 1: Two functions that perform the same *action*.

Command	**Output**	**Command**	**Output**
`{\"a}`	ä	`{\c c}`	ç
`{\^e}`	ê	`{\u g}`	ğ
``{\`i}``	ì	`{\l}`	ł
`{\.I}`	İ	`{\~n}`	ñ
`{\o}`	ø	`{\H o}`	ő
`{\'u}`	ú	`{\v r}`	ř
`{\aa}`	å	`{\ss}`	ß

Table 2: Example commands for accented characters, to be used in, *e.g.*, BibTEX entries.

2 Engines

To produce a PDF file, pdfLATEX is strongly recommended (over original LATEX plus dvips+ps2pdf or dvipdf). XeLATEX also produces PDF files, and is especially suitable for text in non-Latin scripts.

3 Preamble

The first line of the file must be

```
\documentclass[11pt]{article}
```

To load the style file in the review version:

```
\usepackage[review]{ACL2023}
```

For the final version, omit the `review` option:

```
\usepackage{ACL2023}
```

Output	natbib command	Old ACL-style command
(Cooley and Tukey, 1965)	`\citep`	`\cite`
Cooley and Tukey, 1965	`\citealp`	no equivalent
Cooley and Tukey (1965)	`\citet`	`\newcite`
(1965)	`\citeyearpar`	`\shortcite`
Cooley and Tukey's (1965)	`\citeposs`	no equivalent
(FFT; Cooley and Tukey, 1965)	`\citep[FFT;][]`	no equivalent

Table 3: Citation commands supported by the style file. The style is based on the natbib package and supports all natbib citation commands. It also supports commands defined in previous ACL style files for compatibility.

To use Times Roman, put the following in the preamble:

```
\usepackage{times}
```

(Alternatives like txfonts or newtx are also acceptable.) Please see the LaTeX source of this document for comments on other packages that may be useful. Set the title and author using `\title` and `\author`. Within the author list, format multiple authors using `\and` and `\And` and `\AND`; please see the LaTeX source for examples. By default, the box containing the title and author names is set to the minimum of 5 cm. If you need more space, include the following in the preamble:

```
\setlength\titlebox{<dim>}
```

where `<dim>` is replaced with a length. Do not set this length smaller than 5 cm.

4 Document Body

4.1 Footnotes

Footnotes are inserted with the `\footnote` command.[1]

4.2 Tables and figures

See Table 2 for an example of a table and its caption. **Do not override the default caption sizes.**

4.3 Hyperlinks

Users of older versions of LaTeX may encounter the following error during compilation:

```
\pdfendlink ended up in different
nesting level than \pdfstartlink.
```

This happens when pdfLaTeX is used and a citation splits across a page boundary. The best way to fix this is to upgrade LaTeX to 2018-12-01 or later.

4.4 Citations

Table 3 shows the syntax supported by the style files. We encourage you to use the natbib styles. You can use the command `\citet` (cite in text) to get "author (year)" citations, like this citation to a paper by Gusfield (1997). You can use the command `\citep` (cite in parentheses) to get "(author, year)" citations (Gusfield, 1997). You can use the command `\citealp` (alternative cite without parentheses) to get "author, year" citations, which is useful for using citations within parentheses (e.g. Gusfield, 1997).

4.5 References

The LaTeX and BibTeX style files provided roughly follow the American Psychological Association format. If your own bib file is named `custom.bib`, then placing the following before any appendices in your LaTeX file will generate the references section for you:

```
\bibliographystyle{acl_natbib}
\bibliography{custom}
```

You can obtain the complete ACL Anthology as a BibTeX file from `https://aclweb.org/anthology/anthology.bib.gz`. To include both the Anthology and your own .bib file, use the following instead of the above.

```
\bibliographystyle{acl_natbib}
\bibliography{anthology,custom}
```

Please see Section 5 for information on preparing BibTeX files.

4.6 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

[1] This is a footnote.

5 BibTEX Files

Unicode cannot be used in BibTEX entries, and some ways of typing special characters can disrupt BibTEX's alphabetization. The recommended way of typing special characters is shown in Table 2.

Please ensure that BibTEX records contain DOIs or URLs when possible, and for all the ACL materials that you reference. Use the `doi` field for DOIs and the `url` field for URLs. If a BibTEX entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref LATEX package.

Limitations

ACL 2023 requires all submissions to have a section titled "Limitations", for discussing the limitations of the paper as a complement to the discussion of strengths in the main text. This section should occur after the conclusion, but before the references. It will not count towards the page limit. The discussion of limitations is mandatory. Papers without a limitation section will be desk-rejected without review.

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

Ethics Statement

Scientific work published at ACL 2023 must comply with the ACL Ethics Policy.[2] We encourage all authors to include an explicit ethics statement on the broader impact of the work, or other ethical considerations after the conclusion but before the references. The ethics statement will not count toward the page limit (8 pages for long, 4 pages for short papers).

Acknowledgements

This document has been adapted by Jordan Boyd-Graber, Naoaki Okazaki, Anna Rogers from the style files used for earlier ACL, EMNLP and NAACL proceedings, including those for EACL 2023 by Isabelle Augenstein and Andreas Vlachos, EMNLP 2022 by Yue Zhang, Ryan Cotterell and Lea Frermann, ACL 2020 by Steven Bethard, Ryan Cotterell and Rui Yan, ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTEX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

[2] `https://www.aclweb.org/portal/content/acl-code-ethics`

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Example Appendix

This is a section in the appendix.