SMU
SINGAPORE MANAGEMENT UNIVERSITY
School of Computing and Information Systems

Can Identifier Splitting Improve Open-Vocabulary Language Model of Code?

Jieke Shi, Zhou Yang, Junda He, Bowen Xu, David Lo

Overview

Motivation:

- Karampatsis et al. [1] applied the Byte Pair Encoding (BPE) algorithm [2] to construct open-vocabulary LMs, which have outstanding performance.
- A drawback of BPE is that it cannot split the identifiers in a way that preserves the meaningful semantics (As the example).
- Prior researchers show that splitting compound identifiers into sub-words that reflect the semantics can benefit software development tools.

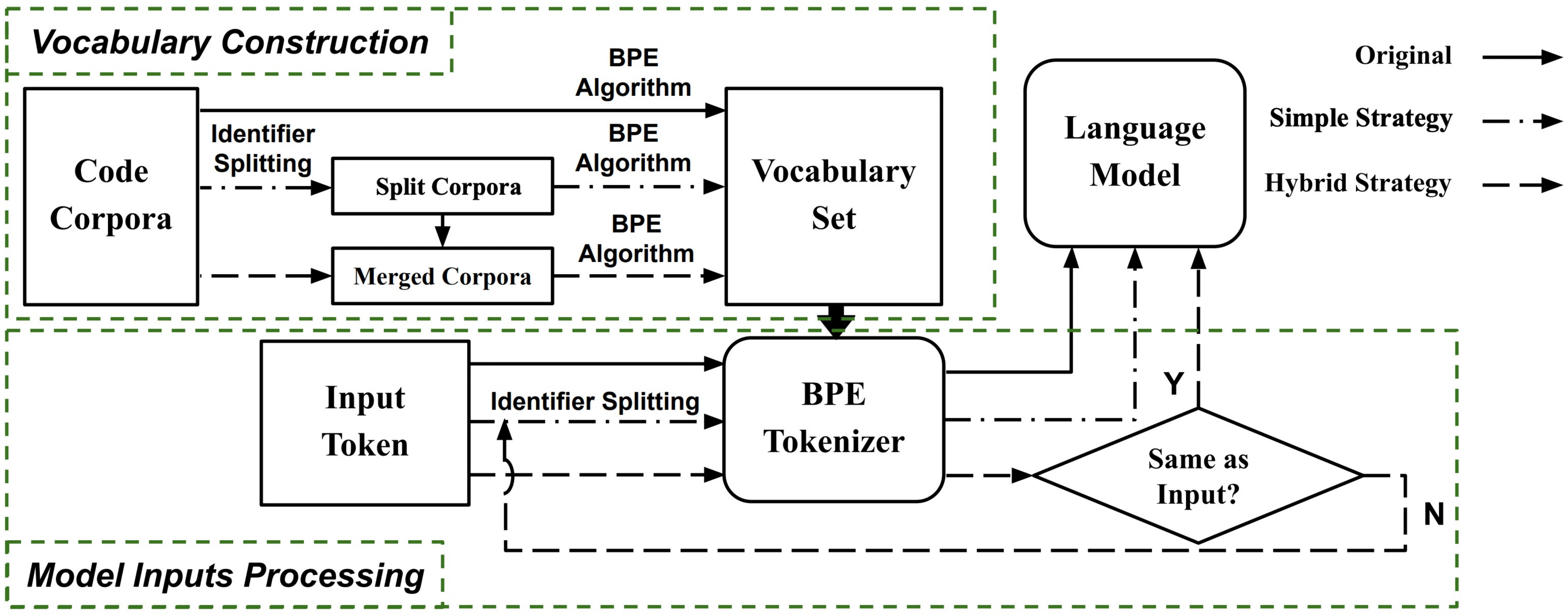
Contributions:

- We are the first to propose to apply identifier splitting to language models of code.
- We contrast the performance of LMs under different settings and find that:
  - Simply inserting identifier splitting into the pipeline hurts the model performance;
  - A hybrid strategy combining identifier splitting and BPE algorithm can improve the original open-vocabulary LMs.

Example:

getListener (Original) ✓ get Listener (Human Understanding) ✗ get List ener (BPE)

Methodology



Vocabulary Construction
Code Corpora
Identifier Splitting
Split Corpora
Merged Corpora
BPE Algorithm
BPE Algorithm
BPE Algorithm
Vocabulary Set
Language Model
Original
Simple Strategy
Hybrid Strategy
Input Token
Identifier Splitting
BPE Tokenizer
Same as Input?
Y
N
Model Inputs Processing

The overview of how to combine BPE and identifier splitting in the LMs of code

Original:

- using BPE and not splitting identifiers

Simple Strategy:

- splitting all identifiers in corpora then use BPE to construct a vocabulary
- splitting all identifiers in model inputs

Hybrid Strategy:

- splitting identifiers and merging them with original corpora for BPE vocabulary construction
- splitting identifiers in model inputs only when BPE fails to tokenize them as the original forms

Experiment Results

Strategy	All Tokens		Identifiers	
	Entropy	MRR	R@10	MRR
Original	4.46	64.41	37.55	21.83
Simple	4.45(-0.22%)	64.31(-0.46%)	36.26(-3.44%)	20.59(-5.68)
Hybrid	**4.37(-2.02%)**	**65.24(+0.98%)**	**38.93(+3.68%)**	**23.19(+6.23%)**

Evaluation results on C language [1] dataset.

Analysis:

- Simply performing identifier splitting into preprocessing procedures does not suffice and degrades the performance of LMs.
- By following the hybrid strategy, identifier splitting boosts the performance of open-vocabulary LMs of code by a decent margin.

Conclusion and Future Work

Conclusion:

- Provide an evidence that the benefits of identifier splitting methods on open-vocabulary language models for C language.

Future Work:

- Validate our findings on more programming languages beyond C.
- Investigate more language models with different architectures.

Reference

[1] Karampatsis, Rafael-Michael, Hlib Babii, et al. "Big code!= big vocabulary: Open-vocabulary models for source code." ICSE 2020.
[2] Sennrich, Rico, Barry Haddow, et al. "Neural Machine Translation of Rare Words with Subword Units." ACL 2016.

Artifacts:

Preprint:

Presentation Video:

Our Group:

SOAR
SOFTWARE ANALYTICS RESEARCH GROUP