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.

Example: getListener (Original) ✔ get Listener (Human Understanding) ❌ get List ener (BPE)

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.

Methodology

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

Experiment Results

<table>
<thead>
<tr>
<th>Strategy</th>
<th>All Tokens</th>
<th>Identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Entropy</td>
<td>MRR</td>
</tr>
<tr>
<td>Original</td>
<td>4.46</td>
<td>64.41</td>
</tr>
<tr>
<td>Simple</td>
<td>4.45(-0.22%)</td>
<td>64.31(-0.46%)</td>
</tr>
<tr>
<td>Hybrid</td>
<td>4.37(-2.02%)</td>
<td>65.24(+0.98%)</td>
</tr>
</tbody>
</table>

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