Boosting Neural Commit Message Generation with Code Semantic Analysis

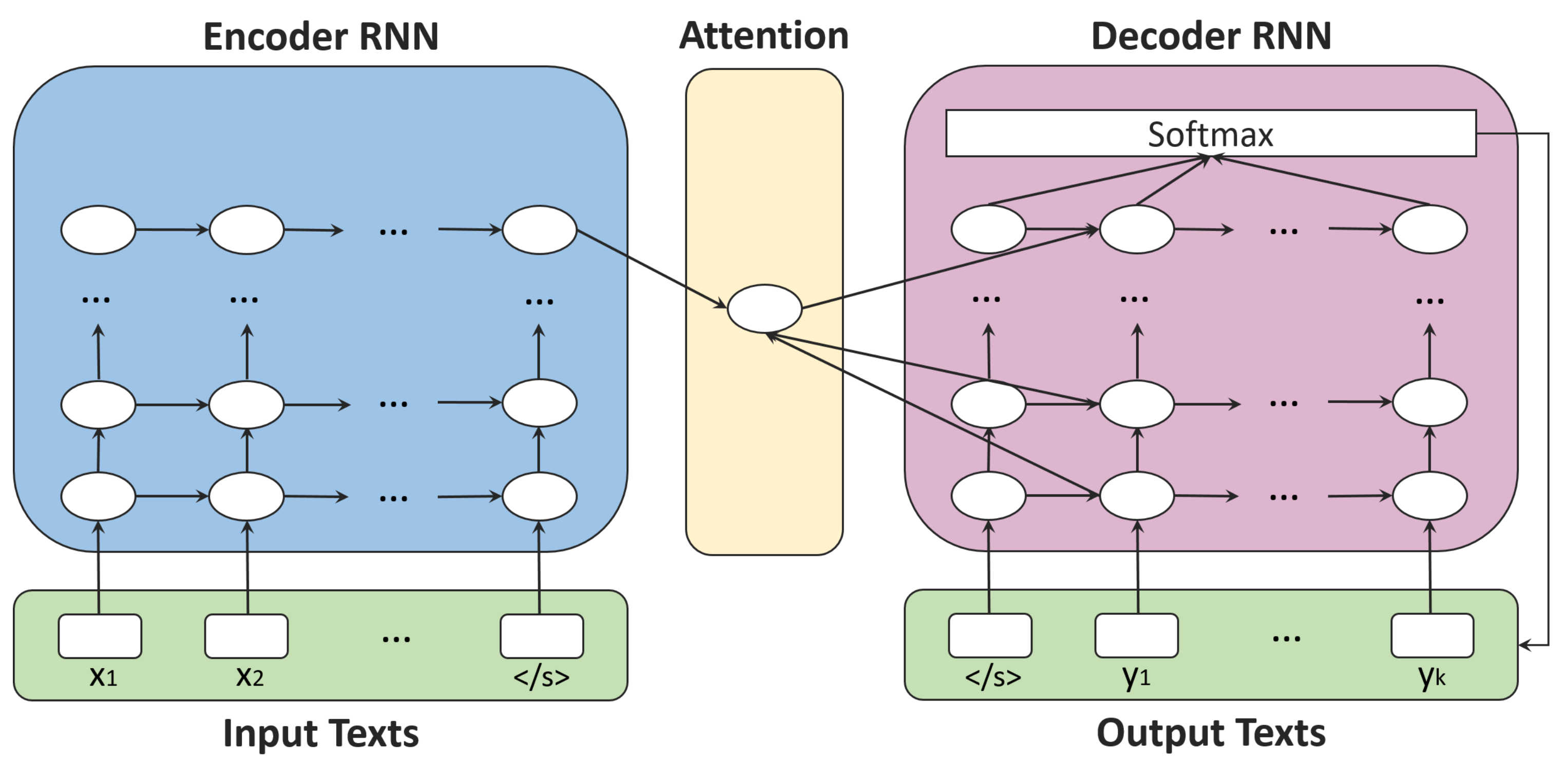
Shuyao Jiang
Fudan University, Shanghai, China *(syjiang15@fudan.edu.cn)*

Introduction

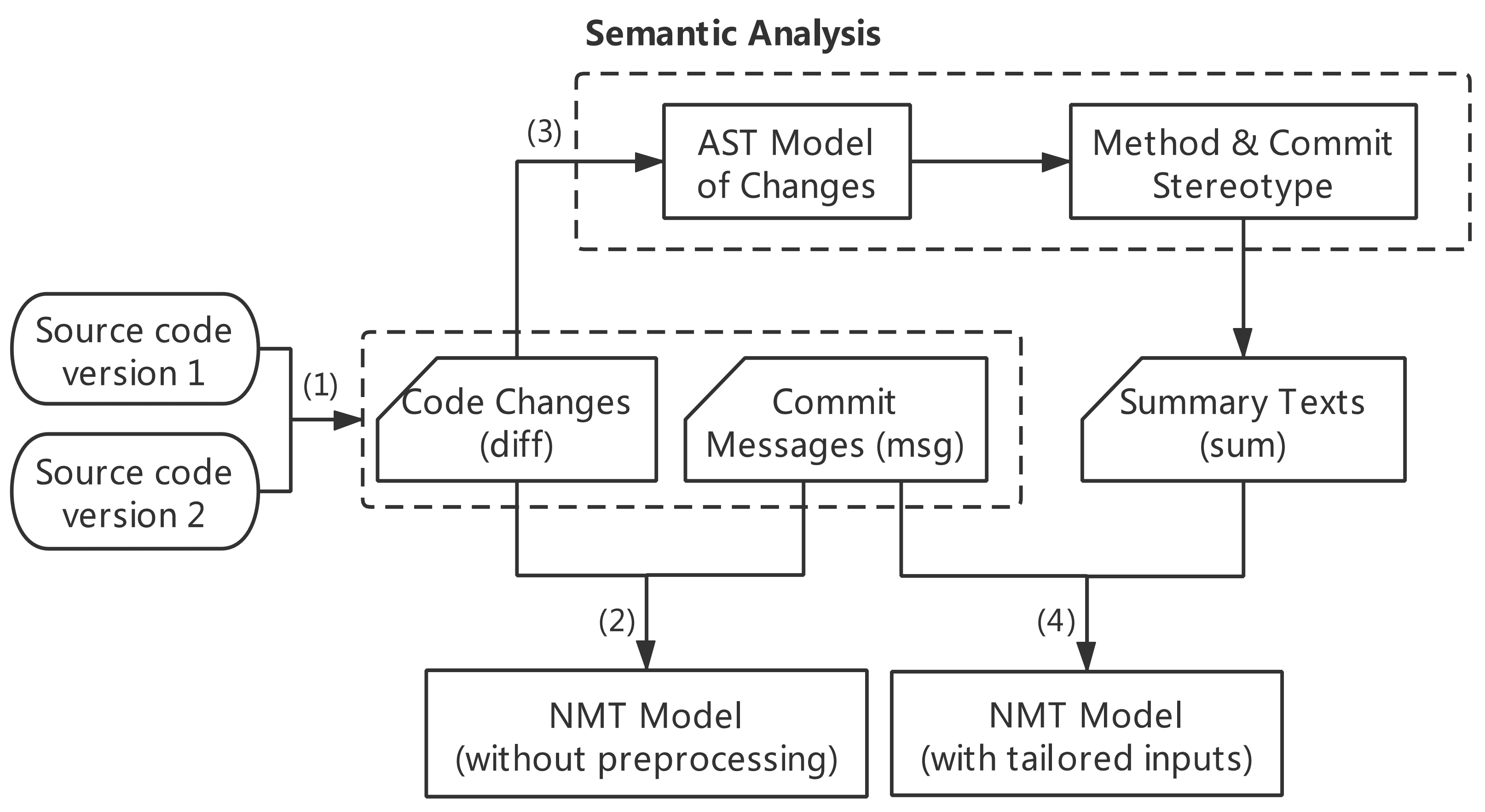
- Neural machine translation (NMT) has been suggested to automatically generate commit messages, but the quality of generated messages is not yet satisfactory.
- This work suggests that
  - Proper preprocessing of code changes is critical to improve NMT for commit message generation.
  - Code semantic analysis can be applied to tailor inputs.

Background

Neural Machine Translation Architecture



Approach



Traditional NMT-based approach: Steps (1) - (2)

- Take raw code changes (diffs) as inputs
- Diffs generally contain too much noisy information, leading to NMT performance degradation

Our approach: Steps (1) - (3) - (4)

- Apply code semantic analysis to tailor inputs
- Convert long code changes into short summary texts
- Translate summary texts into commit messages

Our approach suggests that proper preprocessing of code changes is critical to improve the performance of NMT.

Experimental Study

Dataset: 18 popular Java projects from GitHub

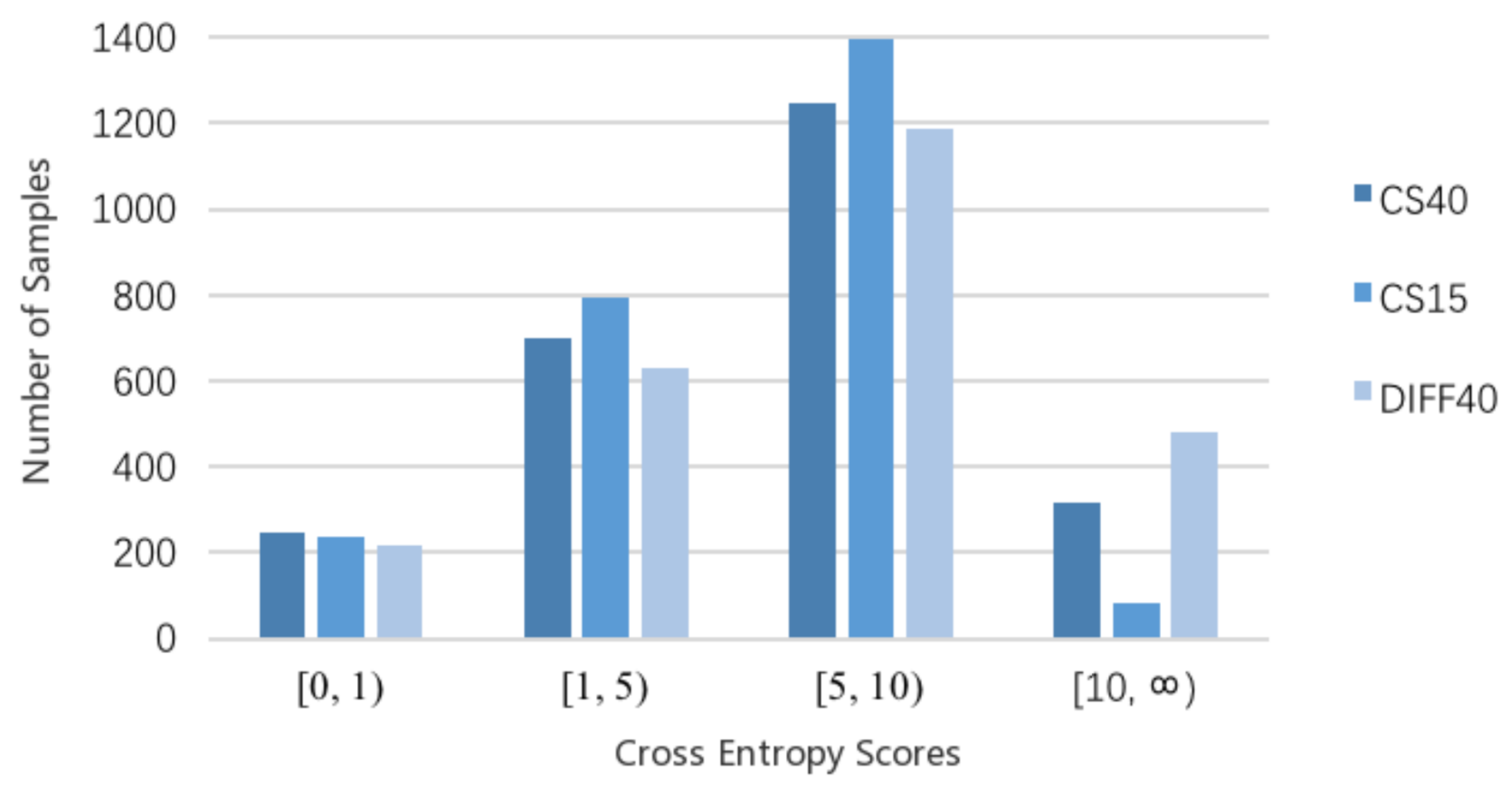
- **Each:** 20k+ stars, 100k+ code lines, 3k+ commits
- **In total:** 50k+ commits
- **Release:** https://github.com/ShuyaoJiang/CommitDataset

Models: Attentional RNN Encoder-Decoder

- **CS40:** with tailored inputs, batch size 40
- **CS15:** with tailored inputs, batch size 15
- **DIFF40:** without tailored inputs, batch size 40

Quantitative Evaluation (2.5k commits)

➢ Cross Entropy



Our approach can obtain more commit messages with low cross entropy loss (i.e., more similar to the reference messages).

➢ BLEU Scores

Model	BLEU-4	p_1	p_2	p_3	p_4
CS40	1.10	4.7	1.7	0.5	0.4
CS15	0.44	9.1	3.2	0.1	0.0
DIFF40	0.41	3.9	0.9	0.1	0.1

* p_n is the modified n-gram precision used to calculate BLEU-4

CS40 has the highest BLEU scores (i.e., has higher translation quality), indicating it outperforms traditional approach.

Translation Example

Diff:	**Summary Text:**
+++ b/guava-tests/test/com/google/common/base/EquivalenceTest.java, + import com.google.common. testing.NullPointerTester;, + public void testEquals() { …	Changes to package com.google.common.base: Add a class for package sanity tests. It allows to: Instantiate package sanity tests.
DIFF40 Translation: Add <UNK>	**CS40 Translation:** Add support for task properties

Reference Commit Message:
Applied package sanity tests to common/base

Conclusion

- We suggest that **data preprocessing** is critical to commit message generation with NMT.
- We apply **code semantic analysis** to tailor NMT inputs in commit message generation.
- We conduct a **comprehensive evaluation** to prove that data preprocessing does improve NMT.
- This work sheds light to how to **properly apply existing DNN models** in software engineering tasks.