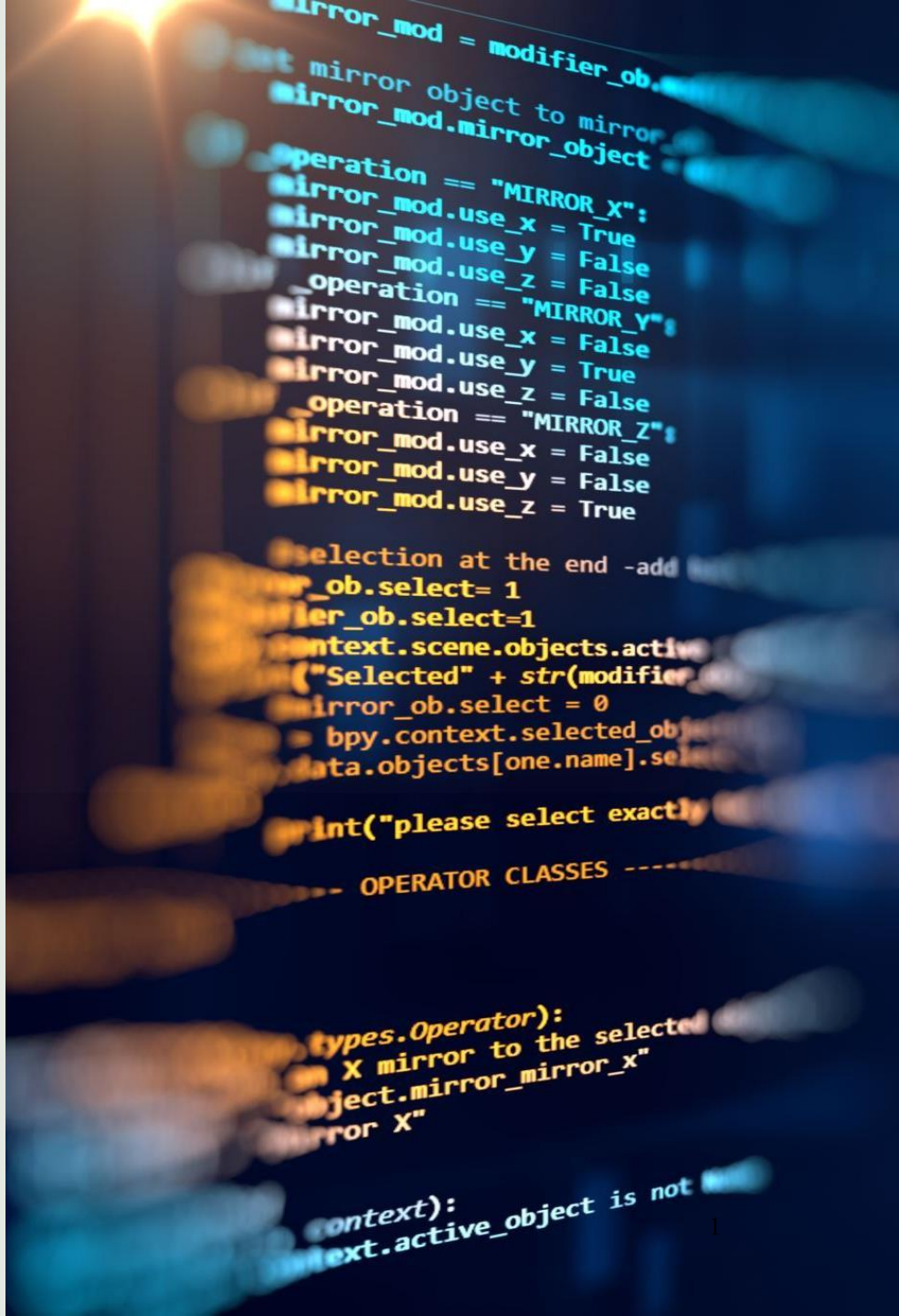
COMMIT BERT: COMMIT MESSAGE GENERATION USING PRE-TRAINED PROGRAMMING LANGUAGE MODEL

Tae-Hwan Jung from Kyung Hee University

Presentation made by Martin Guillemot at Tamkang University

2023-11-22



PAPER PRESENTATION

COMMIT BERT: COMMIT MESSAGE GENERATION USING PRE-TRAINED PROGRAMMING LANGUAGE MODEL

- Written by : Tae-Hwan Jung from Kyung Hee University

CommitBERT explores the use of a pre-trained AI language model to automatically generate commit messages, bridging the gap between code changes and their natural language descriptions, aiming to enhance developer collaboration and code change comprehension.

SUMMARY

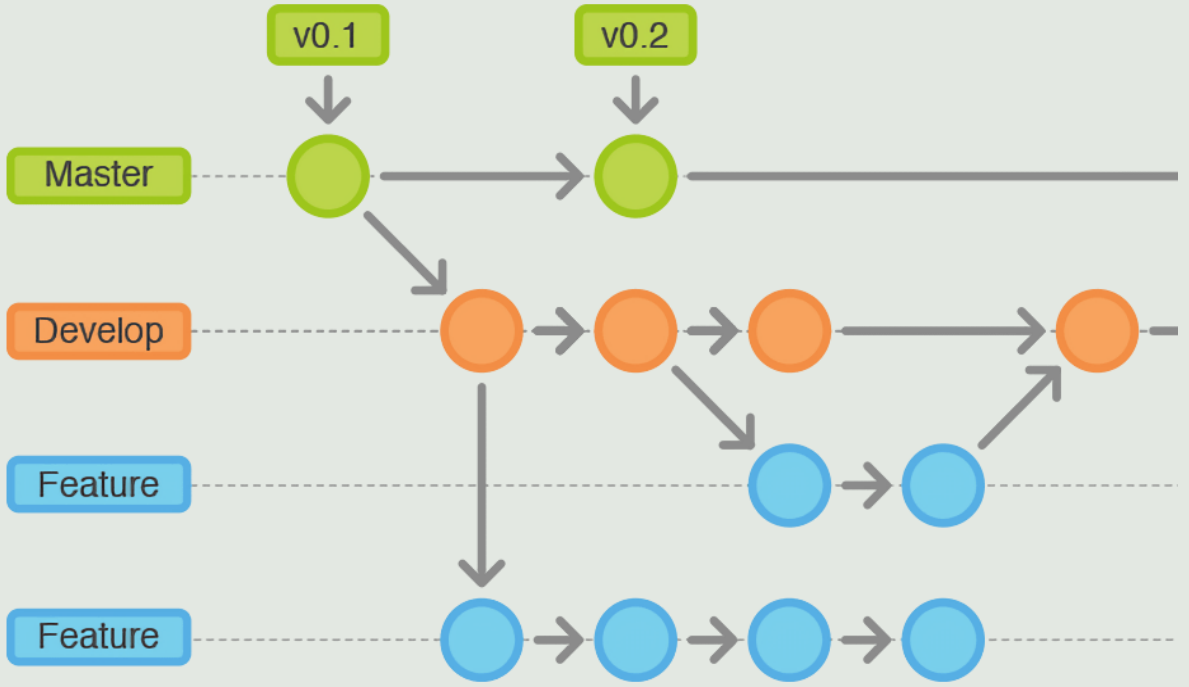
- WHAT IS GIT ?
- IMPORTANCE OF GOOD COMMIT MESSAGES
- GOOD PRACTICE FOR WRITING A MESSAGE
- PROBLEM IDENTIFY
- DATA COLLECTION
- RESOLVING THE CONTEXTUAL REPRESENTATION GAP
- PERFORMANCE MEASUREMENT
- FIRST EXPERIENCE
- SECOND EXPERIENCE
- CONCLUSION
- IMPROVEMENT PROPOSAL

WHAT IS GIT ?

Git is a decentralized version control system designed for efficient management of projects.

Enabling developers to :

- Track changes
- Collaborate seamlessly
- Maintain a comprehensive history of their code evolution.



IMPORTANCE OF GOOD COMMIT MESSAGES

- Clarity in Code History
- Facilitates Team Collaboration
- Eases Problem Troubleshooting
- Improves Code Review Efficiency
- Supports Project Documentation

Don't do this :

- *"Fixed stuff" x42*
- *"I don't know what I did but it works"*
- *"Oops, broke it"*
- *"Something changed, I guess"*
- *"Committing changes before leaving for lunch 🍔"*

GOOD PRACTICE FOR WRITING A MESSAGE

- Be Concise but Descriptive
- Use the Imperative way:

Verb	Description
Add Fix Use Update – Upgrade Remove Change Improve	• the #3025 bug • margin to nav items to prevent them from overlapping the logo • npm dependency to latest version • bug preventing users from submitting the subscribe form

PROBLEM IDENTIFY

There are two main problems when generating commits :

- No Dataset Available:
  - Examples of changes made in the code (like things added or removed) and the messages that explain these changes, for different kinds of computer languages.

- Contextual Learning Difficulty:
  - Significant challenge for AI models to have a big difference between their input and their output.

DATA COLLECTION

- **Commit Limitation:** Collection capped at 50 commits per repository to ensure format diversity.
- **File Change Filter:** Only commits with one or two file changes are selected.
- **Issue Number Exclusion:** Commits with issue numbers removed for clarity.
- **Language Filter:** Non-English commit messages excluded.
- **First Line Only:** For long commit messages, only the first line is used.
- **Code Token Limit:** Commits with code tokens over 32 characters, parsed by tree-sitter, are excluded.
- **Verb Type Focus:** Focus on 13 most frequent verb types.

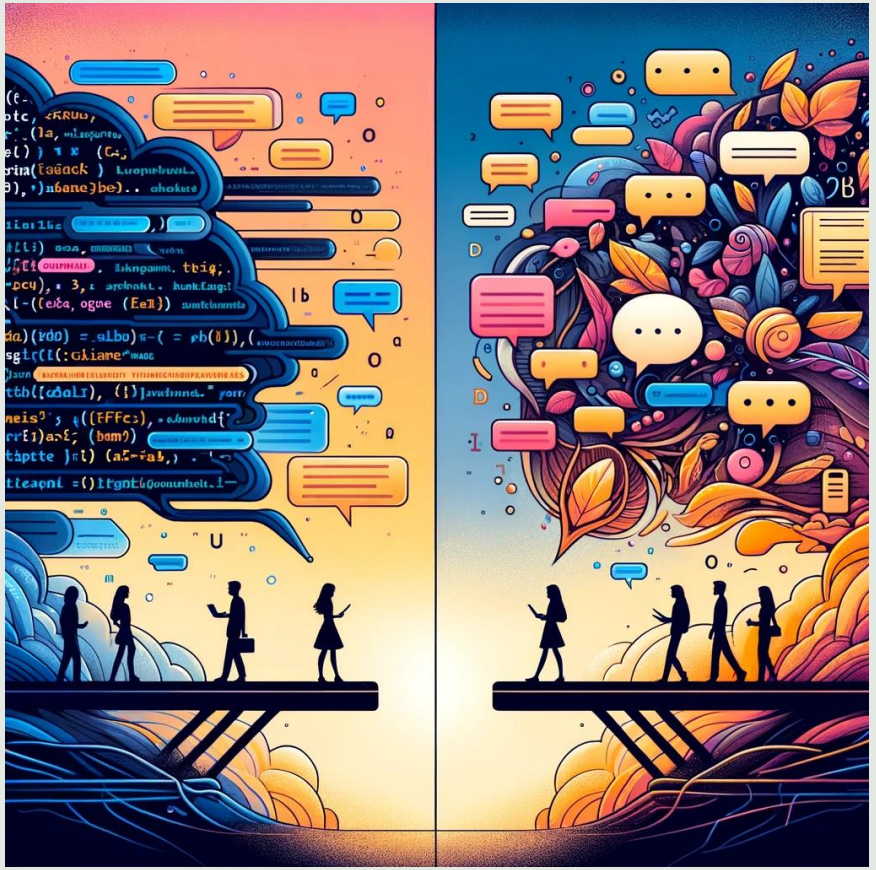
We obtain :

- ***345,000*** pairs of changes.
- On ***6*** language : Python, PHP, Go, Java, JavaScript et Ruby.

```
procedure REPOPARSER(Repos)
  for Repo in Repos do
    commits = get_commits(Repo)
    for commit in commits do
      mods = get_modifications(commit)
      for mod in mods do
        if filtering(mod, commit) then
          break
        end if
        Save (mod.add, mod.del) to dataset.
      end for
    end for
  end for
end procedure
```

RESOLVING THE CONTEXTUAL REPRESENTATION GAP

- *We use CodeBERT !*
- What is CodeBERT?
  - A state-of-the-art machine learning model designed by researchers at Microsoft.
  - Specializes in understanding and processing programming languages and natural language.
- Purpose of CodeBERT
  - Built to bridge the gap between human language and programming languages.
  - Aims to improve automated tasks like code generation, documentation, and translation between programming languages.



PERFORMANCE MEASUREMENT

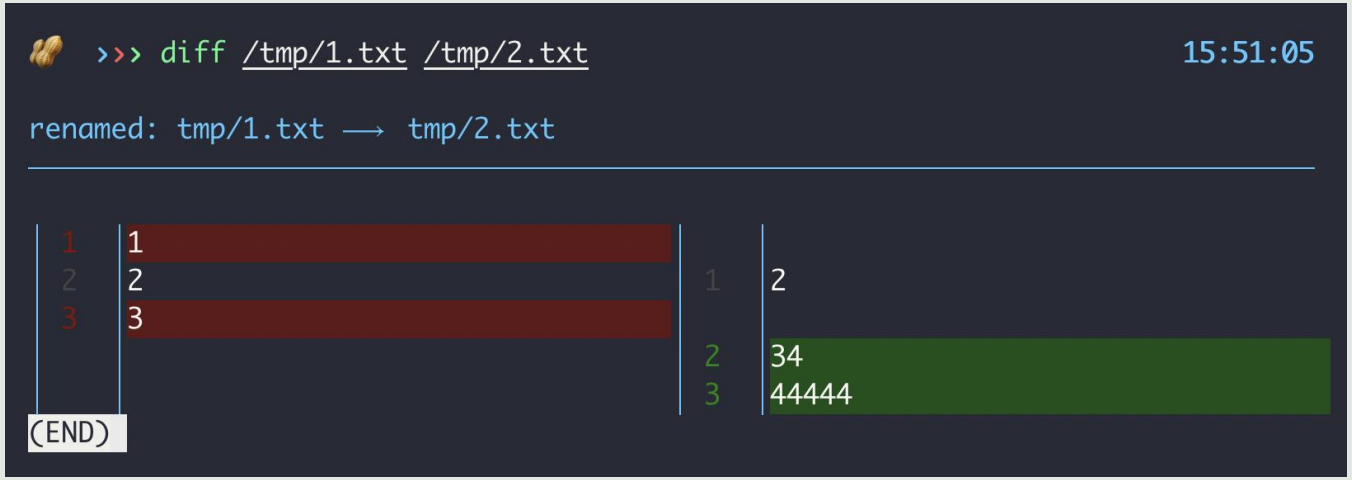
- BLEU-4:
  - Purpose: BLEU-4 is a way to check how good automatic translations by AI are.
  - How it works: BLEU-4 compares AI-created text to one or more human-created texts. It looks at how similar the sentences are.

- PPL (Perplexity):
  - Purpose: Perplexity measures how well AI language models understand language.
  - How it works: A low perplexity score means the AI is good at predicting words.

	Number of Pair Dataset			
	Train	Validation	Test	Number of Repositories
Python	81517	10318	10258	12361
PHP	64458	8079	8100	16143
JavaScript	50561	6296	6252	11294
Ruby	29842	3772	3680	4581
Java	28069	3549	3552	4123
Go	21945	2699	2812	3960
Total : 345759				52462

FIRST EXPERIENCE :

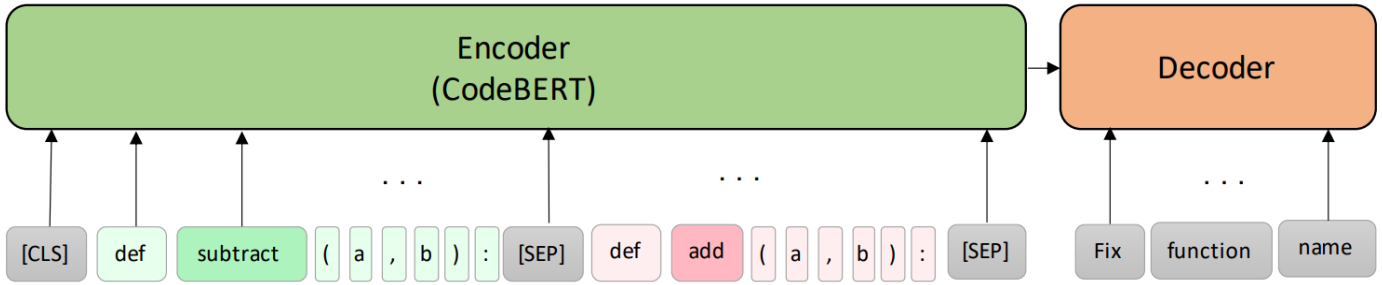
- Types of Inputs Compared
  - Complete Code Modifications.
  - Only Added or Deleted Lines.
- Results
  - Better Performance with Selective Inputs.



Initial Weight	Input Type	BLEU-4
RoBERTa	(a) All code modification	10.91
	(b) Only changed lines (**Ours**)	**12.52**
CodeBERT	(a) All code modification	11.77
	(b) Only changed lines (**Ours**)	**13.32**

SECOND EXPERIENCE

```
- def add(a, b):
+ def subtract(a, b):
      return a - b
```



Metric	Initial Weight	Python	PHP	JavaScript	Java	Go	Ruby
BLEU-4 (Test)	(a) Random	7.95	7.01	8.41	7.60	10.38	7.17
	(b) RoBERTa	10.94	9.71	9.50	6.40	10.21	8.95
	(c) CodeBERT	12.05	13.06	10.47	8.91	11.19	10.33
	(d) CodeBERT + Code-to-NL	**12.93**	**14.30**	**11.49**	**9.81**	**12.76**	**10.56**
PPL (Dev)	(a) Random	144.60	138.39	195.98	275.84	257.29	207.67
	(b) RoBERTa	76.02	81.97	103.48	164.32	122.70	104.68
	(c) CodeBERT	68.18	63.90	94.62	116.50	109.43	91.50
	(d) CodeBERT + Code-to-NL	**49.29**	**47.89**	**75.53**	**77.80**	**64.43**	**82.82**

CONCLUSION

Summary of Work:

- The study presents a model designed to summarize code modifications, addressing the challenge of manually writing commit messages.

Methodological Contributions:

Proposed methods include :

- A data collection technique.
- A process for integrating data into the model.
- Strategies for enhancing model performance.

Successful Outcomes:

- Demonstrated successful generation of commit messages using the proposed methods.

Practical Implications:

- The model offers assistance to developers who struggle with writing commit messages.

IMPROVEMENT PROPOSAL

Future Research Directions:

- While high-quality commit messages can be generated with a pre-trained model, further studies are needed to understand code syntax structure.
- Proposed Future Enhancement: Convert CommitBERT to use AST (Abstract Syntax Tree).

QUESTIONS ?