

Generování kódu z textového popisu funkcionality

Ondřej Zobal

vedoucí: doc. RNDr. Pavel Smrž Ph.D.



- I přes současný rozkvět umělé inteligence, editoru kódu postrádají jednoduché rozhraní na vykonávání primitivních úkonů vyžadujících inteligenci.
- Nástroje jako GitHub Copilot umožňují chatovat s jazykovým modelem, dotazy je však nutné pokaždé vypisovat.
- V mé práci představuji rozšíření **CodeImprove** pro VS Code, které pomocí vlastních účelových jazykových modelů integruje možnosti generování **jmen** proměnných, **in-line komentářů** a **docstringů** v souborech zdrojového kódu jazyka **Python 3**.

- Při práci je využita knihovna PyTorch, knihovny Transformers a další.
- Každá AI funkce používá svůj **vlastní model**.
- Před trénováním byly do modelů načteny váhy generického NL-PL modelu PLBART^[1].
- Modely využívají architekturu Longformer Encoder-Decoder^[2], která umožňuje prodloužení vstupní délky s lineární prostorovou komplexitou.
- Dotrénované modely CodeImprove dokážou oproti původnímu modelu PLBART pojmout **4x více tokenů** (2048).
- Modely byly dotrénovány na datové sadě CodeXGlue^[3].

- Dalším modulem pro CodeImprove byla automatická detekce a oprava chyb.
- V rámci dotrénování tohoto modulu jsem nasbíral 16 500 vzorků oprav jednoduchých chyb.
- Experimenty se dvěma architekturami: Longformer^[2] a QLoRA^[4].
- Experimenty bohužel nepřinesly uspokojivé výsledky

- V rámci klienta neběží žádné neuronové sítě, vše se odesílá na server, který provádí všechny náročné výpočty.
- Klient je ve VS Code implementován pomocí API pro rozšíření. Je napsaný v JavaScriptu. Server je implementovaný v Pythonu
- Komunikace mezi klientem a serverem probíhá přes HTTP.

```
Suggest comments | Fix error
4  def fibonacci(n):
5      fib_sequence = [0, 1]
6      while len(fib_sequence) < n:
7          next_value = fib_sequence[-1] + fib_sequence[-2]
8          fib_sequence.append(next_value)
9      return fib_sequence[:n]
10
```

- Po kliknutí na funkci se v editoru otevře boční panel, v němž jsou prezentovány výsledky.
- Uživatel si může návrh přijmout či zahodit.
- Přijetím návrh se automaticky provedou změny v textu.

Suggest a Name

Suggestions for renaming `n`:

fib_sequence_length
fib_sequence_size
fib_sequence_len

Cancel

Comment Suggestions

Suggestions for more comments in your code:

No particular formatting	Google style	Numpy style	reST
--------------------------	--------------	-------------	------

```
def fibonacci(n):
    """Return a sequence of fibonacci numbers.

    Returns a sequence with the same length as the fibonacci number.
    """
    fib_sequence = [0, 1]
    # Append the next value to the fib sequence
    while len(fib_sequence) < n:
        next_value = fib_sequence[-1] + fib_sequence[-2]
        fib_sequence.append(next_value)
    return fib_sequence[:n]
```

Accept Cancel

Generování jmen proměnných (přesnost slov, kompletní shoda)

	CodeImprove-rename	GPT-3.5-turbo
TOP3UP	35.63 %	37.74 %
TOP3EM	10 %	18 %

Generování in-line komentářů (BLEU)

	BLEU total	Unigram	Bigram	Trigram	Quadgram
CodeImprove-comment	14.99 %	72.7 %	40.0 %	5.6 %	3.1 %
GPT-3.5-turbo	12.04 %	94.1 %	37.5 %,	3.3 %	1.8 %

Generování docstringů

	BLEU total	Unigram	Bigram	Trigram	Quadgram
CodeImprove-docstring	27.92 %	67.6 %	42.4 %	21.9 %	9.7 %
GPT-3.5-turbo	38.01 %	98.3 %	78.0 % ,	31.0 %	8.8 %

Děkuji za pozornost.

Literatura

- [1] Ahmad, W. U.; Chakraborty, S.; Ray, B. and Chang, K.-W. Unified Pre-training for Program Understanding and Generation. 2021.
- [2] Beltagy, I.; Peters, M. E. and Cohan, A. Longformer: The Long-Document Transformer. 2020.
- [3] Lu, S.; Guo, D.; Ren, S.; Huang, J.; Svyatkovskiy, A. et al. CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation. CoRR, 2021, abs/2102.04664.
- [4] June, F. QLoRA: Key Quantization and Fine-tuning Techniques in the Era of Large Language Models. Medium, dec 2023. Available at: <https://ai.plainenglish.io/qlora-key-quantization-and-fine-tuning-techniques-in-the-era-of-large-language-models-0fa05a961d27>. Accessed: 2023-04-23.

Zkuste zdůvodnit, proč selhává model v úloze "Korekce chyb".

- Generování oprav je relativně komplexní 1B model nemusel být dostatečný
- I když se chyba týká jen pár řádků, logika za ní je často velmi komplexní a bývá závislá na dalších komponentách
- Doménově specifický kód může být matoucí
- **Důvodem byla pravděpodobně kombinace těchto faktorů.**