



南京大学

NANJING UNIVERSITY

数据科学基础课程大作业

—基于大语言模型代码注释自动生成

汇报人：孙伟松

汇报时间：2023.05



目录

- 一. 任务目标
- 二. 任务内容
- 三. 任务要求
- 四. 提交材料
- 五. 评分标准
- 六. 参考文献

一、任务目标

任务目标

基于大语言模型代码注释自动生成

```
private void destroy() {  
    if(log.isLoggable(Level.FINEST)) {  
        log.finest(this + _STR);  
    }  
    try {  
        _socket.forceShutdown();  
    } catch(Throwable e) {}  
    try {  
        closeConnection();  
    } catch(Throwable e) {  
        log.log(Level.FINER, e.toString(), e);  
    }  
    _port.removeConnection(this);  
}
```

(a) A Java code snippet s_1

Ex-based: destroys the connection. this implementation closes the connection.
SiT: removes this connection from the port.
CodeT5: close the connection without removing all resources.
EACS: destroy the connection (force shutdown).

(b) Comments generated by different models for s_1

```
def addAttributeWord(evaluatorWords, word):  
    if (len(word) < 2):  
        evaluatorWords.append(word)  
        return  
    if (word[0] != '.'):   
        evaluatorWords.append(word)  
        return  
    dotIndex = word.find('.', 1)  
    if (dotIndex < 0):  
        evaluatorWords.append(word)  
        return  
    evaluatorWords.append(word[:dotIndex])  
    addAttributeWord(evaluatorWords, word[dotIndex:])
```

(c) A Python code snippet s_2

Ex-based: adds a word to the given word.
SiT: add a single word to the evaluator words.
CodeT5: add the word and return a list of word.
EACS: add the word to the list of attribute words.

(d) Comments generated by different models for s_2

二、任务内容

任务内容

基于大语言模型代码注释自动生成

➤ 大语言模型 (Large Language Model)

- General LLMs: ChatGPT, Alpaca, LLaMA, ChatGLM ...
- Specialized Code LLMs: Codex, StarCoder, CodeGeeX ...

➤ 自动化代码注释生成 (Automatic Code Summarization)

```
public V remove(final K key) {  
    V oldValue = cacheMap.remove(key);  
    if (oldValue != null) {  
        LOG.debug("Removed cache entry for '{}'", key);  
    }  
    return oldValue;  
}
```

(a) A code snippet s_1

removes the mapping for the specified key from this cache if present.

(b) A comment c_1 generated by code summarization model for s_1

三、任务要求

任务要求

1. 调研哪些 LLMs 可以做代码注释生成（分类）；
2. 设计 Prompts 指导 LLMs 生成高质量的注释；
3. 分析已有的 Benchmark 数据集中注释质量；
4. 分析 LLMs 生成的注释存在的问题；
5. 结合 LLMs 生成的注释和 Benchmark 数据集中的注释，手动整理一批高质量的代码-注释数据集；
6. 用要求 5 中数据集对现有 LLMs 进行微调进一步提升性能；
7. 开发代码注释生成工具原型；
8. 系统性地报告结果。

任务要求


1. 调研哪些 LLMs 可以做代码注释生成（分类）；

Single Model

Chatbot Arena (battle)

Chatbot Arena (side-by-side)

Leaderboard

 **Chat with Open Large Language Models**

- Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. [\[Blog post\]](#)
- Koala: A Dialogue Model for Academic Research. [\[Blog post\]](#)
- [\[GitHub\]](#) [\[Twitter\]](#) [\[Discord\]](#)

Terms of use

By using this service, users are required to agree to the following terms: The service is a research preview intended for non-commercial use only. It only provides limited safety measures and may generate offensive content. It must not be used for any illegal, harmful, violent, racist, or sexual purposes. **The service collects user dialogue data and reserves the right to distribute it under a Creative Commons Attribution (CC-BY) license.**

Choose a model to chat with

Vicuna : a chat assistant fine-tuned from LLaMA on user-shared conversations by LMSYS	Koala : a dialogue model for academic research by BAIR	OpenAssistant : an Open Assistant for everyone by LAION
RMKV-4-Raven : an RNN with transformer-level LLM performance	Alpaca : a model fine-tuned from LLaMA on instruction-following demonstrations by Stanford	ChatGLM : an open bilingual dialogue language model by Tsinghua University
LLaMA : open and efficient foundation language models by Meta	Dolly : an instruction-tuned open large language model by Databricks	StableLM : Stability AI language models
FastChat-T5 : a chat assistant fine-tuned from FLAN-T5 by LMSYS		

<https://chat.lmsys.org>

<https://huggingface.co/blog/starcoder>

任务要求

2. 设计 Prompts 指导 LLMs 生成高质量的注释；

vicuna-13b

× ▾

Scroll down and start chatting

Please generate comments for the following function:

```
def mostFrequentEven(self, nums: List[int]) -> int:
```

```
    ans = -1
```

```
    cnt = 0
```

```
    for k, v in Counter(nums).items():
```

```
        if k % 2 == 0 and (v > cnt or v == cnt and k < ans):
```

```
            ans = k
```

```
            cnt = v
```

```
    return ans
```

Prompt

This function takes a list of integers and returns the most frequent even number in the list. It uses a counter object to count the number of occurrences of each integer in the list and compares the count of even numbers to the count of all numbers. If the count of even numbers is greater than or equal to the count of all numbers and the most frequent even number has a lower index than the previous most frequent even number, the function updates the most frequent even number.

The function uses a variable `cnt` to keep track of the count of each even number and a variable `ans` to store the most frequent even number. It iterates through the items in the counter object, which are the tuples `(k, v)` where `k` is the key and `v` is the value of the counter. For each tuple, the function checks if the key `k` is even (`k % 2 == 0`) and if the count of the key `v` is greater than or equal to the current count of even numbers (`cnt`) and the count of all numbers (`k < ans`). If all conditions are met, the function updates the most frequent even number by setting `ans` to `k`.

Answer

任务要求

3. 分析已有的 Benchmark 数据集中注释质量；

现有代码摘要模型在 CodeSearchNet 数据集上的性能

Methods	Go			Java			JavaScript			PHP			Python			Ruby		
	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}
LSTM	17.8	15.1	35.6	12.2	10.1	24.6	10.4	6.2	17.2	19.5	12.2	29.8	13.9	9.1	23.3	9.4	5.3	16.3
Transformer	19.8	16.2	38.4	15.3	11.8	30.6	11.2	7.4	20.5	21.5	13.9	34.2	15.8	10.6	31.3	10.3	6.4	18.3
CodeBERT	21.1	17.5	43.6	18.0	12.4	35.5	13.3	8.7	24.3	24.6	15.3	39.4	18.7	12.4	34.8	11.2	7.1	20.6
CodeT5	22.1	18.5	44.8	20.4	14.5	38.1	15.8	11.2	28.9	25.9	18.0	43.0	20.0	15.1	37.8	14.9	10.8	27.9
TR-based	5.1	4.1	9.3	7.4	5.1	10.2	6.1	4.2	8.4	11.0	6.8	12.7	5.8	3.9	7.6	6.3	4.1	8.5
EX-based	21.6	18.9	43.6	19.8	14.6	37.5	14.9	10.5	27.7	24.7	17.2	41.1	18.6	13.6	35.5	14.9	10.4	27.9
ICS	23.4	20.4	46.9	20.9	15.2	39.1	15.9	11.4	29.1	25.9	18.2	43.2	20.4	15.1	38.6	15.2	11.1	28.7

注： \mathcal{B} , BLEU; \mathcal{M} , METEOR; \mathcal{R} : ROUGE-L.

Ground-truth 注释本身可能是低质量的

任务要求

3. 分析已有的 Benchmark 数据集中注释质量;

```
def vimeo_download_by_channel(url, output_dir='.'
    , merge=False, info_only=False, **kwargs):
    channel_id = match1(url, r'http://vimeo.com/channels/(\w+)')
    vimeo_download_by_channel_id(channel_id, output_dir, merge,
    info_only, **kwargs)
```

(a) A code snippet s_1

str - > None.

(b) The ground-truth comment of s_1

Ground-truth 注释本身可能是低质量的

任务要求

4. 分析 LLMs 生成的注释存在的问题;



```
Please generate a short comment for the following function:  
def validate_value(key, value, expected_type):  
    if not isinstance(value, expected_type):  
        raise TypeError("{} argument must have a type {} not {}".format(  
            key, expected_type, type(value)))
```



This is a Python function that validates the type of a given value against an expected type. The function takes three arguments: key, value, and expected_type. The key argument is a string that identifies the value being validated, the value argument is the actual value being validated, and the expected_type argument is the type that the value is expected to be.

The function works by using the isinstance() function to check if the value argument is an instance of the expected_type. If the value is not an instance of the expected type, the function raises a TypeError with a message that describes the error.

The error message includes the key argument, the expected_type argument and the type of the value argument. This provides a clear message to the caller of the function about what went wrong and what they need to do to fix the error.

Overall, this function is useful for enforcing type constraints in Python code and can help catch errors early in the development process.

An example of rich but lengthy comments generated by ChatGPT

任务要求

5. 结合 LLMs 生成的注释和 Benchmark 数据集中的注释，手动整理一批高质量的代码-注释数据集；

```
def vimeo_download_by_channel(url, output_dir='.'
    , merge=False, info_only=False, **kwargs):
    channel_id = match1(url, r'http://vimeo.com/channels/(\w+)')
    vimeo_download_by_channel_id(channel_id, output_dir, merge,
    info_only, **kwargs)
```

(a) A code snippet s_1

GT

str -> None.

(b) The ground-truth comment of s_1



This Python function downloads Vimeo videos by channel ID.

(c) The comment generated by ChatGPT for s_1

任务要求

6. 结合要求 5 中数据集对现有 LLMs 进行微调进一步提升性能；
现有代码摘要模型在 CodeSearchNet 数据集上的性能

Methods	Go			Java			JavaScript			PHP			Python			Ruby		
	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}	\mathcal{B}	\mathcal{M}	\mathcal{R}
LSTM	17.8	15.1	35.6	12.2	10.1	24.6	10.4	6.2	17.2	19.5	12.2	29.8	13.9	9.1	23.3	9.4	5.3	16.3
Transformer	19.8	16.2	38.4	15.3	11.8	30.6	11.2	7.4	20.5	21.5	13.9	34.2	15.8	10.6	31.3	10.3	6.4	18.3
CodeBERT	21.1	17.5	43.6	18.0	12.4	35.5	13.3	8.7	24.3	24.6	15.3	39.4	18.7	12.4	34.8	11.2	7.1	20.6
CodeT5	22.1	18.5	44.8	20.4	14.5	38.1	15.8	11.2	28.9	25.9	18.0	43.0	20.0	15.1	37.8	14.9	10.8	27.9
TR-based	5.1	4.1	9.3	7.4	5.1	10.2	6.1	4.2	8.4	11.0	6.8	12.7	5.8	3.9	7.6	6.3	4.1	8.5
EX-based	21.6	18.9	43.6	19.8	14.6	37.5	14.9	10.5	27.7	24.7	17.2	41.1	18.6	13.6	35.5	14.9	10.4	27.9
ICS	23.4	20.4	46.9	20.9	15.2	39.1	15.9	11.4	29.1	25.9	18.2	43.2	20.4	15.1	38.6	15.2	11.1	28.7

注： \mathcal{B} , BLEU; \mathcal{M} , METEOR; \mathcal{R} : ROUGE-L.

选一个 LLM（例如 CodeT5）进行微调

任务要求

7. 开发代码注释生成工具原型；

- IDEA 插件；
- Web 页面；
- ...

任务要求

8. 系统性地报告结果；

- 报告哪些 LLMs 能生成注释，这些模型能不能分类；
- 报告良好 Prompts 设计/探索过程；
- 报告已有 Benchmark 数据集中注释存在哪些问题；
- 报告 LLMs 生成的注释存在哪些问题；
- 报告手动构建的高质量的代码-注释数据集；
- 报告微调 LLMs 后的到的结果；
- 报告原型开发情况；

四、提交材料

提交材料

- 文档
 - 任务要求 8 的报告（6月1日）
- 开源链接（7月1日）
 - 微调的代码注释生成模型
 - 高质量代码-注释数据集
- 演示录像（7月1日）
 - 代码注释生成工具原型演示

五、评分标准

评分标准

1. 调研哪些 LLMs 可以做代码注释生成（分类）；
 - 调研的 LLMs 全面性
2. 设计 Prompts 指导 LLMs 生成高质量的注释；
 - Prompt 设计的合理性、系统性
3. 分析已有的 Benchmark 数据集中注释质量；
 - 发现质量问题的种类、数量
4. 分析 LLMs 生成的注释存在的问题；
 - 发现质量问题的种类、数量

评分标准

5. 结合 LLMs 生成的注释和 Benchmark 数据集中的注释，手动整理一批高质量的代码-注释数据集；
 - 手动构建数据集的数量、涉及语言种类
6. 用要求 5 中数据集对现有 LLMs 进行微调进一步提升性能；
 - 提升效果的明显程度
7. 开发代码注释生成工具原型；
 - 工具的可用性、易用性
8. 系统性地报告结果。
 - 报告的完整性、丰富性

六、参考文献

参考文献

1. Cross-Lingual Summarization via ChatGPT
2. Exploring the Limits of ChatGPT for Query or Aspect-based Text Summarization
3. On the Evaluation of Neural Code Summarization
4. Learning code summarization from a small and local dataset
5. Recommendations for Datasets for Source Code Summarization
6. UniXcoder: Unified Cross-Modal Pre-training for Code Representation
7. CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation
8. CodeBERT: A Pre-Trained Model for Programming and Natural Languages
9. StarCoder: A State-of-the-Art LLM for Code



南京大学
NANJING UNIVERSITY

谢谢

汇报人：孙伟松
汇报时间：2023.05

