Software Similarity Test System

For

<我测你们代码队>

Written by:Chen Jie(陈杰) 20301033, He Sichao(贺思超) 20301037

Date: 2022/12/18

Table of Contents

1. Introduction

Basic Points:

- The Programming Languages the system supports to test？
  python

- The Size of System it supports to test (How many lines of codes it support? How many modules it supports? etc.,)
  200MB single .py file

- The Strategy used for the system (Line by Line Character Matching? Control Flow Matching? Data Flow Matching?)

Two ways:

1. **Token block - Block by block comparison**
2. **Winnowing algorithm**

Extra Points:

- **streamlit framework**
- **Visualized Result**
- **website deployment(https://routhleck-code-different-comparision-streamlit-app-1s0mx2.streamlit.app)**

2. General Framework of the System

Basic Points:

- **res - Resource directory for storing logos and ICONS**
- **src - Source directory**
  - **categories.py - Replace python's code with the corresponding token using pyments package, matching the color of each token**
  - **levenshtein.py - Calculate levenshtein distance using numpy (DiffLib has been used instead)**
  - **plot.py - The creation of plot is used to visualize code tokens**
  - **similarity.py - The main implementation of block class and similarity calculation**
  - **winnowing.py - Implementation of winnowing algorithm**
- **test_files - The directory where the.py file for the test is stored**
- **requirements.txt - Project environment configuration information**
- **streamlit_app.py Main program**

Extra Points:

- Defined by your team

3. Overall Strategy and Approach

3.1 Similarity Testing Strategy (Basic)

1.Token block - Block by block comparison:
Use the pygments module to translate the source code from texts to tokens, and Separate tokens into blocks according to newline and indentation. Finally, compare the blocks in *code a* with the blocks in *code b* one by one using the difflib module, take the highest score as the similarity of the blocks of *code a* (between 0 and 1), and calculate the overall similarity of *code a* If the degree of similarity exceeds the threshold, it will be counted + 1, and the overall similarity is the number of blocks exceeding the threshold divided by the number of all blocks.
The similarity of *code b* is calculated analogously to the above function.

2. Winnowing algorithm
A fingerprint for an entire source code is created through the combination of a hashing process and a sliding window. *A document*'s fingerprint consists of a set of hash values. The Jaccard coefficient can be used to obtain the similarity between two source codes.
Jaccard coefficient:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

3.2 Similarity Test Data (Basic)

Ten python source code (in test_file folder).

4. Execution Plan

4.1 System Execution Explanation (Basic)

1. How to translate the source code from texts to tokens

use the *lexer* of *pygments module*

```
def __tokenizeFromText(self, text):
    lexer = PythonLexer()  # 使用lexer做词法分析, 判断属于哪种编程语言
    tokens = lexer.get_tokens(text)
    tokens = list(tokens)  # 转化为list
```

2. How to separate tokens into blocks

separate tokens into blocks according to newline and indentation

```
# 使用category简化tokens
for token in tokens:
    c = get_category(token)

    if (c is not None):
        # 换行检测,更新坐标但不加入result
        if c == 'L':
            row = row + 1
            col = 0

        # 检测到新的block, prev_c为\n且不为缩进
        elif prev_c == 'L' and c != 'I' and result:
            self.blocks.append(Block(result))
            result = []

        # 不为空行
        if c != 'L':
            # 区分函数调用和变量
            if prev_c == 'V' and token[1] == '(':
                result[-1] = 'A', result[-1][1], result[-1][2], result[-1][3]

            result.append((c, row, col, token[1]))
            col += 1
            if col > self._max_col:
                self._max_col = col
    prev_c = c
self._max_row = row  # 依照代码的行数更新最大行数

# 结果不为空则追加最后一个block
if result:
    self.blocks.append(Block(result))
```

3. How to Visualize Code Diagrams

use the *graph_objects* of *plotly* module Display the properties of each block. And Use the *streamlit* frontend to display the plot.

5. Software Similarity Reporting

5.1 Software Similarity Definitions (Basic)

Critical	90-100%
Medium	60%-90%
Low	0%-60%

6. System Installation Environment

6.1 Environment (Basic)

python3.8

install the package the requirements.txt provide↓

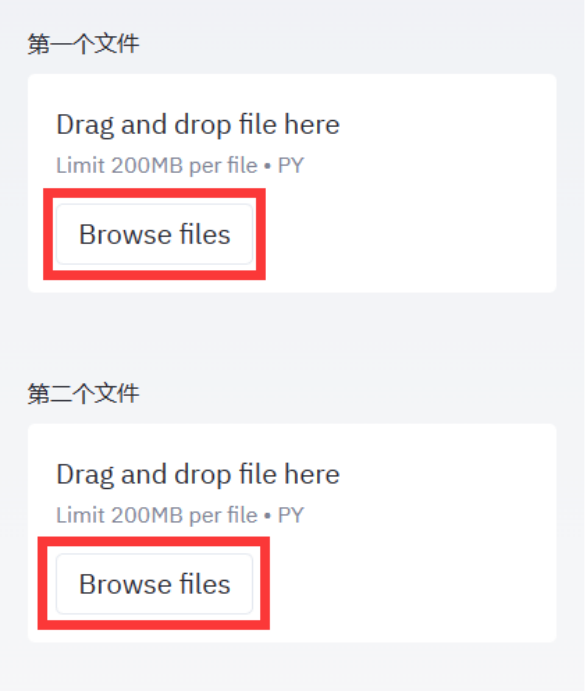
```
plotly==5.11.0
  streamlit==0.74.1
  Pygments==2.13.0
  numpy~=1.23.5
```

and run "***streamlit run streamlit_app.py***" in terminal on the root folder

7. User Manual (Basic)

1. Open https://routhleck-code-different-comparision-streamlit-app-1s0mx2.streamlit.app/ to enter the home page of our software and you will see the page like this.

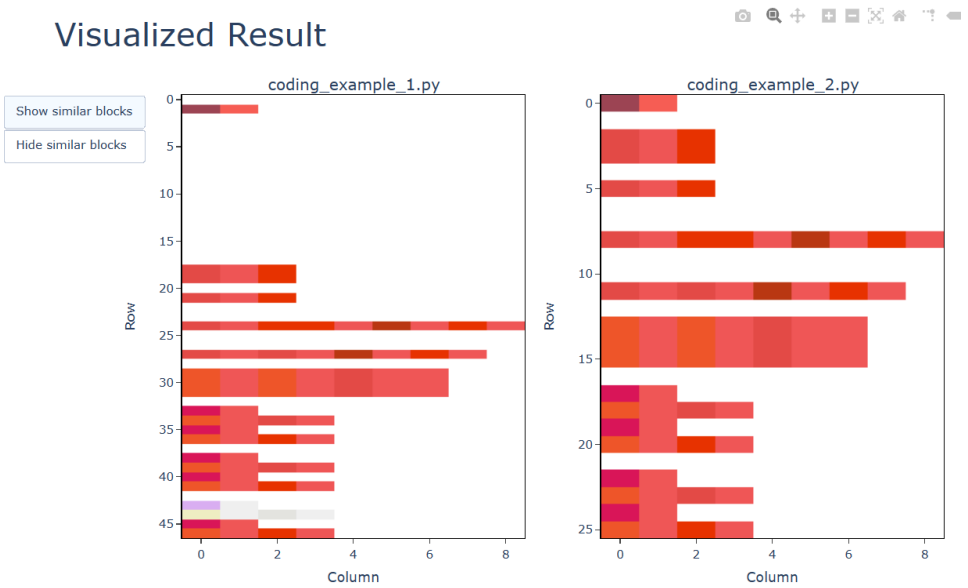
2. Then you can choose two files that you need to compare on these two buttons



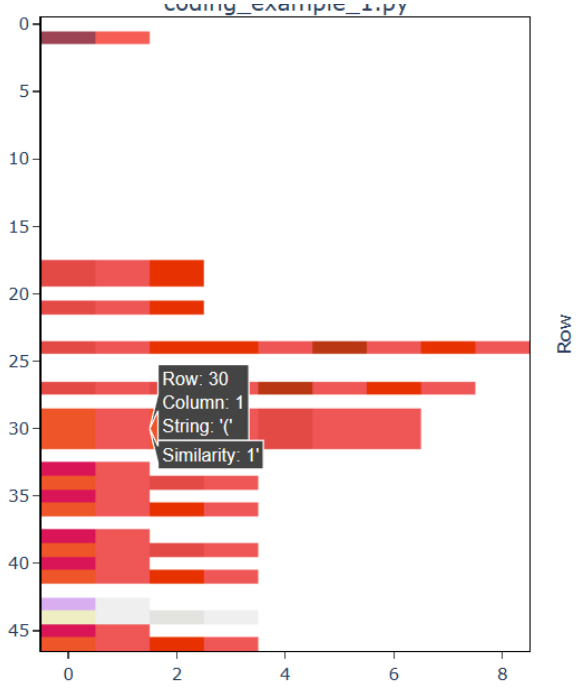
3. Finally， when the two chosen files gotten successfully, you will see the page like this

'*coding_example_1.py*' 的相似度为 **93%** 相似度高, 可以认为是抄袭

'*coding_example_2.py*' 的相似度为 **100%** 相似度高, 可以认为是抄袭



The marked red area means these column or code module are considered have high similarity，and when you put your mouse pointer on the marked area,the area will show you the detail information like this.



Besides,you can change the KGrams and the size of Sliding window on this panel.

you also can change the view mode here,the "High similarity blocks " can show you the similarity by token like this.

s

8. Appendices

References:

[1] Schleimer S , Wilkerson D S , Aiken A . Winnowing: Local Algorithms for Document Fingerprinting[C]// Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003. ACM, 2003.

[2] Hao X , Yan H , Li Z , et al. BUAA_AntiPlagiarism: A System To Detect Plagiarism for C Source Code[C]// International Conference on Computational Intelligence & Software Engineering. IEEE, 2009.