


Phân tách câu nói bằng bài toán Biểu thức chính quy (Regular Expression)

Created by	 Tiến Đạt Lê Văn
Created time	@April 18, 2024 5:23 PM
Tags	

I. Giới thiệu về Bài toán Regular Expression

- Đây là 1 phần quan trọng trong việc lập trình và xử lý ngôn ngữ tự nhiên
- Regrex là 1 chuỗi kí tự đặc biệt được sử dụng để tìm kiếm và so khớp với các mẫu trong văn bản. Các mẫu này có thể là 1 chuỗi kí tự cụ thể hoặc một loạt các ký tự có cấu trúc nhất định
- Regex cho phép bạn thực hiện các thao tác như tìm kiếm, thay thế, trích xuất và kiểm tra tính hợp lệ của dữ liệu dựa trên các mẫu đã định nghĩa trước. Ví dụ, bạn có thể sử dụng regex để kiểm tra xem một địa chỉ email có hợp lệ không, tìm các từ bắt đầu bằng một chữ cái viết hoa, hoặc loại bỏ các ký tự không mong muốn khỏi một chuỗi.
- Regex có thể trở nên phức tạp với các ký tự đặc biệt và quy tắc phức tạp, nhưng khi bạn hiểu và sử dụng chúng một cách hiệu quả, bạn có thể thực hiện các tác vụ xử lý văn bản một cách nhanh chóng và mạnh mẽ.

II. Các cú pháp cơ bản của Regular Expression

Ký hiệu	Ý nghĩa	Ví dụ	Kết quả
'.'	Bất kì kí tự nào, ngoại trừ '\n'	c.t	'cat', 'cut',...
'*'	Kí tự trước nó được lặp lại ít nhất 0 lần	ab*	'a', 'ab', 'abb', ...

'+'	Kí tự trước nó được lặp lại ít nhất 1 lần	ab+	'ab', 'abb', 'abbb',... But not 'a'
'?'	Kí tự trước đó có thể được khớp 1 lần hoặc không	'ab?c'	'abc', 'ac'
'[]'	Chỉ định 1 tập hợp các ký tự có thể khớp	'[abc]'	'a', 'b', 'c'
'^'	Điểm bắt đầu của chuỗi	'^Dat'	'Dat' ở đầu chuỗi
'\$'	Điểm kết thúc của chuỗi	'man city\$'	'man city' ở cuối chuỗi
\\s	Bất kì ký tự có khoảng trắng	\\sa	[space]a, \\ta, \\na
\\S	Bất kì ký tự nào không phải khoảng trắng	\\SF	aF, bF,... But not \\tF, \\nF,...
\\b	Từ biên	ion\\b	connection, transistion,...
\\B	Bất kì vị trí nào không phải là biên	\\BX\\B	EXCAPE, EXCLUSIVE,... Bất kì từ nào có X ở giữa

III. Nguyên lý hoạt động

Dựa trên việc sử dụng các ký tự đặc biệt và các quy tắc phù hợp để tìm kiếm và xử lý các chuỗi.

- **Ký tự đặc biệt:** Regex sử dụng một số ký tự đặc biệt để biểu diễn các mẫu cụ thể. Ví dụ, dấu chấm (.) thể hiện bất kỳ ký tự nào, dấu mũ (^) thể hiện đầu chuỗi, dấu \$ thể hiện cuối chuỗi, và nhóm các ký tự trong dấu ngoặc vuông [] thể hiện bất kỳ ký tự trong nhóm đó.
- **Các quy tắc phù hợp:** Regex cho phép bạn xác định các quy tắc phù hợp với các mẫu cụ thể. Ví dụ, bạn có thể chỉ định rằng một ký tự cụ thể phải xuất hiện một hoặc nhiều lần bằng cách sử dụng dấu + hoặc *, hoặc chỉ định số lượng lần xuất hiện cụ thể bằng cách sử dụng dấu {n} hoặc {min, max}.
- **Tìm kiếm và thay thế:** Regex không chỉ giúp bạn tìm kiếm các mẫu trong văn bản, mà còn cho phép bạn thay thế các chuỗi phù hợp với mẫu đó bằng một chuỗi khác.

- **Phân nhóm và chiếm giữ kết quả:** Bằng cách sử dụng dấu ngoặc (), bạn có thể phân nhóm các phần của mẫu và chiếm giữ kết quả tìm kiếm của các nhóm đó để sử dụng sau này.

IV. Thuật toán

```
import re

def phan_tich_cau(c):
    pattern = re.compile(r'(bật|tắt|mở|đóng|tăng|giảm|đo)\s(đèn|quạt|cửa|cửa\s+sổ|nhiệt độ|độ ẩm|nhiệt độ)')

    match = pattern.search(c)

    if match:
        action = match.group(1)
        device = match.group(2)
        room = match.group(4) # Sử dụng group(4) để trích xuất phòng
        return action, device, room
    else:
        return None
```

Hàm `phan_tich_cau(c)`: Hàm này nhận đầu vào là một câu cần phân tích và thực hiện việc tìm kiếm các mẫu được định nghĩa bằng biểu thức chính quy trong câu đó.

- Trong biểu thức chính quy, mẫu được tìm kiếm bao gồm các từ khóa như "bật", "tắt", "mở", "đóng", "tăng", "giảm", "đo" và danh sách các thiết bị như "đèn", "quạt", "cửa", "cửa sổ", "nhiệt độ", "độ ẩm", "nhiệt độ độ ẩm".
- Ngoài ra, mẫu cũng có thể bao gồm thông tin về vị trí của thiết bị, như "trong", "ngoài", "bên", "ở trong".
- Cuối cùng, mẫu cũng có thể chứa thông tin về phòng như "phòng khách", "nhà bếp", "garage", "ga ra", "chính", "phòng ngủ ba mẹ", "phòng ngủ con cái", "phòng ngủ trẻ em".

Tìm kiếm mẫu trong câu: Sử dụng phương thức `search()` của đối tượng `pattern` đã được tạo ra từ biểu thức chính quy để tìm kiếm mẫu trong câu đầu vào.

Trích xuất thông tin từ kết quả tìm kiếm: Nếu một kết quả tìm kiếm được tìm

thấy, hàm sẽ trích xuất các thông tin như hành động, thiết bị và phòng từ kết quả tìm kiếm và trả về chúng.

```
chuoi = input()
# Phân tích câu
result = phan_tich_cau(chuoi)
# Hiển thị kết quả
if result:
    action, device, room = result
    print(f"(action) (device) (room)")
else:
    print("không tìm thấy.")
```

In kết quả: Nếu có kết quả trả về từ hàm `phan_tich_cau()`, đoạn mã sẽ in ra các thông tin về hành động, thiết bị và phòng. Nếu không tìm thấy kết quả, nó sẽ in ra thông báo "Không tìm thấy."

```
# Tôi muốn mở cửa trong phòng khách
# mở cửa phòng khách
PS D:\VOUT_Folder\OUT_28232824\Voc_ki_6\PBL5\DeepFake Voice Recognition> |
```