Modul Praktikum NLP: Penggunaan Regular Expression dengan Python

Tujuan

1. Memahami konsep dasar regular expression.
2. Mampu menggunakan modul **re** dalam Python untuk melakukan pencarian dan manipulasi teks.
3. Mengaplikasikan regular expression dalam pemrosesan teks dalam NLP.

Pendahuluan

Apa itu Regular Expression?

Regular expression (regex atau regexp) adalah sebuah urutan karakter yang membentuk pola pencarian. Ini digunakan untuk mencocokkan dan memanipulasi teks berdasarkan pola tertentu. Regular expression sangat berguna dalam pemrosesan teks, penggantian teks, validasi data, dan banyak lagi.

Regular expression (regex) adalah notasi standar yang digunakan untuk mendeskripsikan pola pencarian. Regex dapat digunakan untuk berbagai macam tugas, termasuk:

- Mencari pola tertentu dalam string
- Mengganti pola tertentu dalam string
- Membagi string menjadi bagian-bagian berdasarkan pola

Dalam NLP, regex digunakan untuk berbagai macam tugas, termasuk:

- Prapemrosesan teks, seperti menghapus tanda baca atau mengubah format tanggal
- Klasifikasi teks, seperti mengidentifikasi jenis teks berdasarkan pola tertentu
- Ekstraksi informasi dari teks, seperti nomor telepon atau alamat email

Tujuan Praktikum

Tujuan praktikum ini adalah untuk mempelajari dasar-dasar regex dan menerapkannya dalam NLP menggunakan Python.

Requirements:

- Jupyter Notebook / Colab / text editor lain.
- Python 3.
- NLTK.
- re

Guided

1. Pengenalan Regular Expression

1.1 Karakter Dasar

- **.**, *****, **+**, **?**, **|**, **()**, **[]**, **{}**, **^**, **$**

1.2 Karakter Khusus

- **\d**, **\D**, **\w**, **\W**, **\s**, **\S**

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

2. Penggunaan Modul re Python

2.1 Mengimpor Modul **re**

pythonCopy code

import re

2.2 Fungsi Utama dalam Modul **re**

- **re.search()**
- **re.match()**
- **re.findall()**
- **re.sub()**

3. Contoh Penggunaan

3.1 Pattern Matching

Untuk mencari sebuah pola di dalam sebuah string, maka fungsi match dan findall dapat digunakan.

Fungsi search

untuk mencari pola regular expression pertama yang cocok dengan string input

```
# search untuk mencari pola regular expression pertama yang cocok dengan string input
import re

txt = "Ronaldo dari Portugal"
x = re.search("Portugal", txt)
print(x)
```

Output:

```
<re.Match object; span=(13, 21), match='Portugal'>
```

Contoh lain:

```
import re

txt = "Ronaldo dari portugal"
x = re.search("Messi", txt)
print(x)
```

Output:

```
None
```

Fungsi match

Fungsi match akan mengembalikan objek MatchObject jika pola ditemukan di awal string, atau None jika pola tidak ditemukan.

```
# match (untuk mencari string yang berawalan kata tertentu)

import re

text = "Halo, nama saya Edward."

patterns = ["nama", "Edward", "Halo"]

for pattern in patterns:
    match = re.match(pattern, text)
    if match:
        print(f"Pola ditemukan di awal string. >>> {match}")
    else:
        print(f"Pola tidak ditemukan di awal string. >>> {match}")
```

Output:

```
Pola tidak ditemukan di awal string. >>> None
Pola tidak ditemukan di awal string. >>> None
Pola ditemukan di awal string. >>> <re.Match object; span=(0, 4), match='Halo'>
```

Fungsi findall

Fungsi findall akan mengembalikan daftar objek MatchObject yang mewakili semua pola yang ditemukan di dalam string.

```python
# findall (untuk menemukan semua kecocokan dari pola regular expression dalam sebuah string input)

import re

text = "nomor telepon steve adalah 081554469780, sedangkan jobs adalah 087465537920."
pattern = r'\d{12}'
# pattern = r'\d{12,13}'
matches = re.findall(pattern, text)

print(matches)
```

Output:

```
['081554469780', '087465537920']
```

Contoh lain findall:

```python
# contoh lain dari findall (menemukan seluruh dokumen yang mengandung kata tertentu)

import re

text = "Halo, nama saya John. Saya adalah mantan pegulat WWE."

pattern = ".*John.*"

matches = re.findall(pattern, text)

for match in matches:
    matches = re.findall(pattern, text)
    if match:
        print(match)
    else:
        print("kata tidak ditemukan")
```

Output:

```
Halo, nama saya John. Saya adalah mantan pegulat WWE.
```

3.2 *Text Subtitusion*

Fungsi sub

Fungsi Subtitusi dapat digunakan untuk mengganti pola tertentu dalam string.

```python
# mengganti sebuah kata menggunakan regex

import re

text = "Halo, nama saya Rey. Saya adalah Petarunk MMA."
pattern = "Rey"
replacement = "Don"

new_text = re.sub(pattern, replacement, text)

print(new_text)
```

Output

```
Halo, nama saya Don. Saya adalah Petarunk MMA.
```

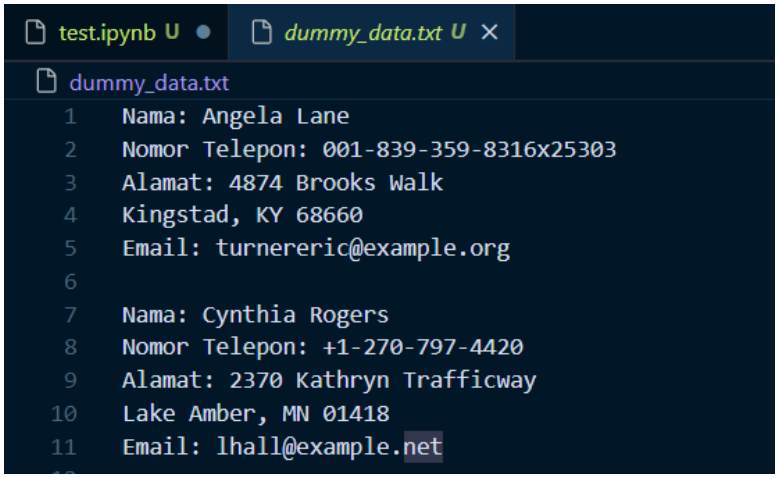
Unguided

4. Studi Kasus

Membuat data dummy dengan menggunakan library faker

```
from faker import Faker

fake = Faker()

jumlah_data = 10

dummy_data = []

for _ in range(jumlah_data):
    nama = fake.name()
    nomor_telepon = fake.phone_number()
    alamat = fake.address()
    email = fake.email()

    data = f"Nama: {nama}\nNomor Telepon: {nomor_telepon}\nAlamat:
{alamat}\nEmail: {email}\n"
    dummy_data.append(data)

file_name = "dummy_data.txt"
with open(file_name, "w") as file:
    for data in dummy_data:
        file.write(data + "\n")

print(f"{jumlah_data} data dummy telah disimpan dalam {file_name}.")
```

Akan muncul file “dummy_data.txt”



Latihan

1. Buatlah sebuah regular expression untuk mencari semua alamat email dalam teks faker.
2. Buatlah sebuah regular expression untuk mencari nomor telepon dalam teks faker.
3. Gabungkan keduanya dan simpan kedalam dataframe
4. Upload kedalam GitHub masing masing (disarankan buat folder baru prak-1)

Kesimpulan

Pada modul praktikum ini, telah dipelajari dasar-dasar regular expression dan cara menggunakannya dalam pemrosesan teks menggunakan modul **re** Python. Hal yang disampaikan pada praktikum kali ini merupakan hal dasar yang diperlukan untuk memahami dan mengaplikasikan regular expression dalam proyek-proyek NLP Anda.

Dokumentasi & Acuan Belajar:

1. w3schools https://www.w3schools.com/python/python_regex.asp

2. regexone https://regexone.com/lesson/introduction_abcs

Referensi

1. Dokumentasi resmi modul **re** Python: https://docs.python.org/3/library/re.html
2. Tutorial regular expression Python: https://docs.python.org/3/howto/regex.html

Modul di atas hanya sebagai panduan awal. Anda dapat menambahkan lebih banyak latihan dan studi kasus untuk memperdalam pemahaman mahasiswa terkait penggunaan regular expression dalam NLP. Semoga modul ini bermanfaat!