Modul Praktikum: Text Preprocessing dalam NLP

Tujuan:

Mempelajari teknik dasar dalam preprocessing teks untuk mempersiapkan data teks untuk analisis NLP.

Tools yang Dibutuhkan:

Python 3

Lowercasing

Lowercasing adalah mengonversi semua karakter teks menjadi huruf kecil. Ini membantu menghindari perbedaan antara huruf besar dan huruf kecil.

Coba inisialisasi variabel untuk menampung teks:

```
text = "Haii selamat pagi. Hari ini hari yang cerah, aku menemukan uang 100.000 saat hendak pulang kerumah. Apakah kamu mau aku traktir?"
text
```

Output:

```
'Haii selamat pagi. Hari ini hari yang cerah, aku menemukan uang 100.000 saat hendak pulang kerumah. Apakah kamu mau aku traktir?'
```

Implementasi

Syntax:

```
text = text.lower()
text
```

Output:

```
'haii selamat pagi. hari ini hari yang cerah, aku menemukan uang 100.000 saat hendak pulang kerumah. apakah kamu mau aku traktir?'
```

Tokenizing

Tokenizing adalah proses membagi teks menjadi token atau kata-kata individu. Kita akan menggunakan library nltk untuk melakukan tokenisasi.

Sent tokenize

-> Memisahkan berdasarkan kalimat dan menyimpan kedalam list

Syntax:

```
import nltk
from nltk import word_tokenize, sent_tokenize

sent_token = sent_tokenize(text)
sent_token
```

Output:

```
['haii selamat pagi.',
 'hari ini hari yang cerah, aku menemukan uang 100.000 saat hendak pulang kerumah.',
 'apakah kamu mau aku traktir?']
```

Word tokenize

-> Memisahkan kata berdasarkan kata per kata

Syntax:

```
word_token = word_tokenize(text)
word_token
```

Output:

```
['haii',
 'selamat',
 'pagi',
 '.',
 'hari',
 'ini',
 'hari',
 'yang',
 'cerah',
 ',',
 'aku',
 'menemukan',
```

Remove Punctuation

Tanda baca seringkali tidak relevan dalam analisis teks. Kita akan menghapus tanda baca dari teks.

Import library:

```
from string import punctuation
```

```
print(punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

Menghilangkan tanda baca:

```
token_bersih = [token for token in word_token if token not in punctuation]
token_bersih
```

Output:

```
['haii',
 'selamat',
 'pagi',
 'hari',
 'ini',
 'hari',
 'yang',
 'cerah',
 'aku',
 'menemukan',
```

Tanda komanya hilang

Stopword Removal

Stopwords adalah kata-kata umum yang sering tidak memberikan informasi berharga dalam analisis teks. Menghapus stopwords dapat menggunakan library nltk, bisa juga menggunakan sastrawi

```
from nltk.corpus import stopwords
```

Menampilkan list stopword Indonesia dari nltk

```
stopwords = stopwords.words('indonesian')
stopwords
```

Output:

```
['ada',
 'adalah',
 'adanya',
 'adapun',
 'agak',
 'agaknya',
 'agar',
 'akan',
 'akankah',
 'akhir',
 'akhiri',
```

Menggunakan Sastrawi:

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
import re

def remove_stopword(words):
    factory = StopWordRemoverFactory()
    stopword_remover = factory.create_stop_word_remover()
    processed_words = [stopword_remover.remove(word) for word in words]

    return processed_words

teks = ['saya tidak suka makan', 'saya lapar', 'kurang tahu bang']
```

```
processed_texts = remove_stopword(teks)

print(processed_texts)
```

Output:

```
['tidak suka makan', 'lapar', 'kurang tahu bang']
```

```
tujuan menghilangkan stopword:
- mengurangi noise
- menghemat memori
- meningkatkan efisiensi
- meningkatkan performa model
```

Lemmatization

Lematisasi adalah proses mengubah kata-kata menjadi bentuk dasarnya. Kita akan menggunakan library nltk untuk melakukan lemmatisasi.

Menggunakan nlp-ide:

```
pip install nlp-id
```

Syntax:

```
from nlp_id.lemmatizer import Lemmatizer

lemmatizer = Lemmatizer()
lemmatizer.lemmatize('Saya sedang mencoba')
```

Output:

```
'saya sedang coba'
```

Slang Word Handling

Penanganan kata slang bisa menjadi langkah tambahan yang tergantung pada dataset Anda. Anda bisa membuat daftar kata slang yang ingin Anda ganti dengan kata yang sesuai dalam bahasa formal. Menggunakan sastrawi

Syntax:

```
slang_words = {
    "gue": "saya",
    "lu": "kamu",
    "gak": "tidak",
    'crds' : 'cerdas'
}

input_text = input("Masukkan teks: ")

words = input_text.split()
```

```
output_words = []
for word in words:
    if word in slang_words:
        output_words.append(slang_words[word])
    else:
        output_words.append(word)

output_text = " ".join(output_words)

print("Hasil: ", output_text)
```

Output:

Menyesuaikan karena menggunakan input dari user

Emoticon Converting

Mengubah emoticon ke teks, menggunakan indoNLP

```
from indoNLP.preprocessing import emoji_to_words
emoji_to_words("emoji: 😳😄")
```

Output:

```
'emoji: !wajah_gembira!!wajah_gembira_dengan_mata_bahagia!'
```

Catatan Penting:

Modul ini hanya memberikan gambaran umum tentang langkah-langkah preprocessing teks dalam NLP. Di dunia nyata, langkah-langkah preprocessing dapat bervariasi tergantung pada dataset dan tujuan analisis Anda. Selain itu, Anda dapat menggunakan berbagai library seperti NLTK, spaCy, atau bahkan regex untuk menjalankan langkah-langkah ini. Pastikan Anda menyesuaikan modul ini dengan kebutuhan praktikum Anda.

Dokumentasi:

https://www.nltk.org/

https://pypi.org/project/indonlp/0.2.0/

https://github.com/IndoNLP

https://pypi.org/project/Sastrawi/

https://github.com/sastrawi/sastrawi