Python ile Metin Normalizasyon (Text Normalization) Uygulaması

Hazırlayan

Fırat Kaan Bitmez

Öğrenci Numarası

23281855

Dersin Hocası

Asst.Prof.Dr. İsmail İşeri

Giriş

Text Normalization, Doğal dil işleme (NLP) alanında önemli bir adımdır. Bu süreç, metin verisinin farklı türlerdeki gürültülerden arındırılmasını ve standart bir forma getirilmesini sağlar. Bu rapor, Python dilinde metin normalizasyonu için geliştirilmiş bir dizi kodlama örneğinin kullanımı ve test edilmesini detaylı bir şekilde açıklamaktadır.

Amacımız

Bu Python uygulamasında amacımız belirli Doğal Dil İşleme Kütüphanelerini kullanarak, Metin Normalleştirme yöntemlerini incelemek ve bu yöntemleri birleştirerek Yeni bir Normalization yöntemi olarak bir metni normalize etmeyi hedefliyoruz.

Kullanılan Kütüphaneler ve Araçlar

NLTK (Natural Language Toolkit): Metin normalizasyonunda yaygın olarak kullanılan bir NLP kütüphanesi. NLTK kütüphanesini dahil etmek için Terminal komutuyla "**Pip intall nltk"** bilgisayarımıza dahil ediyoruz. Daha sonra ihtiyacımız olan diğer modülleri kodlamanın başlangıcında import ederek projeye dahil edeceğiz.

NLTK(wordnet**): Wordnet** NLTK kütüphanesine bağlı bir dil veritabanıdır. Wordnet kelime dağarcıklarını ve bu kelimeler arasındaki ilişkileri içeren bir veritabanıdır.

NLTK(punkt**): Punkt** NLTK'nin cümle ve kelime düzeyinde metinleri bölme(token) için kullandığı veri kaynağını indirir.

string: Python dilinde bulunan string işlemleri için yerleşik kütüphane.

re (Regular Expressions): Düzenli ifadelerin kullanılmasını sağlayan Python modülü.

Kullanılan Normalizasyon Teknikleri

Case Normalization (Büyük/Küçük Harf Normalleştirme): Metin içindeki tüm karakterlerin küçük harfe dönüştürülmesi.

Punctuation Removal (Noktalama İşaretlerinin Kaldırılması): Metinden noktalama işaretlerinin kaldırılması.

Stop Word Removal (Durak Kelime Kaldırma): Metinden yaygın durak kelimelerin (stop words) kaldırılması.

Stemming (Kök Çıkarma): Kelimelerin köklerinin çıkarılması.

Lemmatization: Kelimelerin sözlükteki köklerine dönüştürülmesi.

Tokenization (Belirteçleme): Metnin kelimelere veya belirteçlere bölünmesi.

Synonyms and Abbreviation Replacement (Eşanlamlılar ve Kısaltmaların Değiştirilmesi): Metindeki eşanlamlıların ve kısaltmaların tam hâllerine dönüştürülmesi.

Removing Numbers and Symbols (Sayıların ve Sembollerin Kaldırılması): Metinden sayıların ve sembollerin kaldırılması.

Removing Remaining Non-Textual Elements (Kalan Metin Dışı Unsurların Kaldırılması): Metindeki kalan HTML etiketleri veya URL'lerin kaldırılması.

Kodlama

```
#Bu Proje FIRAT KAAN BİTMEZ tarafından hazırlanmıştır.
#Projede Amacımız 9 farklı normalizasyon tekniklerini öğrenerek bir text
normalization uygulaması yazmaktır.

# pip install nltk komutuyla termail üzerinden kütüphanemizi yükeldikten sonra
#nltk kütüphanesinden gerekli modülleri import ediyoruz
from nltk.stem import PorterStemmer, WordNetLemmatizer
import string
import re
import nltk
import nltk
nltk.download('wordnet')
import nltk
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
# 1) Case Normalization (Büyük/Küçük Harf Normalleştirme)
#Case normalization'da amacımız bütün harfleri küçük hale getirmek
text_case = "The quick BROWN Fox Jumps OVER the lazy dog."
text_case = text_case.lower()
```

Bu kodlama sonucunda aldığımız Çıktı:

the quick brown fox jumps over the lazy dog.

```
# 2) Punctuation Removal (Noktalama İşaretlerinin Kaldırılması)
#Noktalama işaretlerinin kaldırılması, metinden özel karakterleri ve noktalama
işaretlerini kaldırma
text_punctuation = "The quick BROWN Fox Jumps OVER the lazy dog!!!"
text_punctuation = text_punctuation.translate(str.maketrans("", "",
string.punctuation))
```

Bu kodlama sonucunda aldığımız Çıktı:

The quick BROWN Fox Jumps OVER the lazy dog

```
# 3) Stop Word Removal (Durak Kelime Kaldırma)
#Durak kelime kaldırma, “the” ve “a” gibi anlamı çok az olan yaygın kelimelerin
kaldırılmasıdır.
nltk.download('stopwords')
text_stopwords = "The quick BROWN Fox Jumps OVER the lazy dog."
stop_words = set(stopwords.words("english"))
words = text_stopwords.split()
filtered_words = [word for word in words if word.lower() not in stop_words]
text_stopwords = " ".join(filtered_words)
```

Bu kodlama sonucunda aldığımız Çıktı:

quick BROWN Fox Jumps lazy dog.

```
# 4) Stemming (Kök Çıkarma)
#kelimeleri kök haline getirme işlemidir
stemmer = PorterStemmer()
text_stemming = "running,runner,ran"
words = text_stemming.split(",")
stemmed_words = [stemmer.stem(word) for word in words]
text_stemming = ",".join(stemmed_words)
```

Bu kodlama sonucunda aldığımız Çıktı:

run,runner,ran

```
# 5) Lemmatization
#Lemmatizasyon, “koşuyor” kelimesini “koş” olarak kök haline getirirken kelimenin
kullanıldığı bağlamı
#dikkate alarak kelimeleri kök haline getirme işlemidir
lemmatizer = WordNetLemmatizer()
text_lemmatization = "running,runner,ran"
```

```
lemmatized_words = [lemmatizer.lemmatize(word, pos='v') for word in
text_lemmatization.split(",")]
text_lemmatization = ",".join(lemmatized_words)
```

Bu kodlama sonucunda aldığımız Çıktı:

run,runner,run

```
# 6) Tokenization (Belirteçleme)
#Belirteçleme, metni bireysel kelimeler veya ifadeler olarak ayırmak anlamına
gelir
text_tokenization = "The quick BROWN Fox Jumps OVER the lazy dog."
text_tokenization = re.sub(r'[^\w\s]', '', text_tokenization)
tokens = word_tokenize(text_tokenization)
```

Bu kodlama sonucunda aldığımız Çıktı:

['The', 'quick', 'BROWN', 'Fox', 'Jumps', 'OVER', 'the', 'lazy', 'dog']

```
# 7) Replacing synonyms and Abbreviation to their full form to normalize the text
in NLP (Eşanlamlıları ve Kısaltmaları Tam Hâllerine Dönüştürme)
#Bu teknik, metin verisindeki eşanlamlıları veya kısaltmaları tam hâllerine
dönüştürmek gerektiğinde faydalıdır.

text_synonyms = "I'll be there at 2pm"
synonyms = {"I'll": "I will", "2pm": "2 pm"}
for key, value in synonyms.items():
    text_synonyms = text_synonyms.replace(key, value)
```

Bu kodlama sonucunda aldığımız Çıktı:

I will be there at 2 pm

```
# 8) Removing numbers and symbol to normalize the text in NLP (Sayıların ve
Sembollerin Kaldırılması)
#Bu teknik, metin verilerinde önemli olmayan sayıları ve sembolleri kaldırmak
gerektiğinde faydalıdır.
text_numbers = "I have 2 apples and 1 orange #fruits"
text_numbers = re.sub(r"[\d#]", "", text_numbers)
```

Bu kodlama sonucunda aldığımız Çıktı:

I have apples and orange fruits

```
# 9) Removing any remaining non-textual elements to normalize the text in NLP
(Kalan Metin Dışı Unsurların Kaldırılması)
#Bu teknik, metin verilerinde HTML etiketleri, URL'ler ve e-posta adresleri gibi
metin dışı unsurları
#kaldırmak gerektiğinde faydalıdır.
text_non_textual = "Please visit <a href='www.example.com'>example.com</a> for
more information or contact me at info@example.com"
text_non_textual = re.sub(r"(<[^>]+>)|(http[s]?://(?:[a-zA-Z]|[0-9]|[$-
_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+)", "", text_non_textual)
```

Bu kodlama sonucunda aldığımız Çıktı:

Please visit example.com for more information or contact me at info@example.com

```
# 10) Birleştirilmiş Normalizasyon Fonksiyonu
def normalize_text(text):
    text = text.lower()
    text = text.translate(str.maketrans("", "", string.punctuation))

    nltk.download('stopwords')
    stop_words = set(stopwords.words("english"))
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    text = " ".join(filtered_words)

    stemmer = PorterStemmer()
    stemmed_words = [stemmer.stem(word) for word in text.split()]
    text = " ".join(stemmed_words)

    lemmatizer = WordNetLemmatizer()
    lemmatized_words = [lemmatizer.lemmatize(word, pos='v') for word in
text.split()]
    text = " ".join(lemmatized_words)

    text = re.sub(r'[^\w\s]', '', text)
    tokens = word_tokenize(text)

    synonyms = {"I'll": "I will", "2pm": "2 pm"}
    for key, value in synonyms.items():
        text = text.replace(key, value)

    text = re.sub(r"[\d#]", "", text)
```

```
    text = re.sub(r"(<[^>]+>)|(http[s]?://(?:[a-zA-Z]|[0-9]|[$-
_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+)", "", text)

    return text

# Test metni
text = "The quick BROWN Fox Jumps OVER the lazy dog. I'll be there at 2pm. Please
visit <a href='www.example.com'>example.com</a> for more information or contact
me at info@example.com"
# Tüm normalizasyon tekniklerini içeren fonksiyonun uygulanması
normalized_text = normalize_text(text)
```

Bu kodlama sonucunda aldığımız Çıktı:

quick brown fox jump lazi dog ill pm pleas visit hrefwwwexamplecomexamplecoma inform contact infoexamplecom

```
# Normalizasyon yöntemlerinin çıktıları
print("Case Normalization (Büyük/Küçük Harf Normalleştirme):")
print(text_case)
print("\nPunctuation Removal (Noktalama İşaretlerinin Kaldırılması):")
print(text_punctuation)
print("\nStop Word Removal (Durak Kelime Kaldırma):")
print(text_stopwords)
print("\nStemming (Kök Çıkarma):")
print(text_stemming)
print("\nLemmatization:")
print(text_lemmatization)
print("\nTokenization (Belirteçleme):")
print(tokens)
print("\nReplacing synonyms and Abbreviation to their full form to normalize the
text in NLP (Eşanlamlıları ve Kısaltmaları Tam Hâllerine Dönüştürme):")
print(text_synonyms)
print("\nRemoving numbers and symbol to normalize the text in NLP (Sayıların ve
Sembollerin Kaldırılması):")
print(text_numbers)
print("\nRemoving any remaining non-textual elements to normalize the text in NLP
(Kalan Metin Dışı Unsurların Kaldırılması):")
print(text_non_textual)
print("\nBirleştirilmiş Normalizasyon Fonksiyonu:")
print(normalized_text)
```

Tam Sayfa Terminal Çıktısı

```
PS C:\Users\FIRAT\Desktop>  & 'c:\Users\FIRAT\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\FIRAT\.vscode\extensions\ms-python.debugpy-2024.2
6255' '--' 'C:\Users\FIRAT\Desktop\text_normalization.py'
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\FIRAT\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\FIRAT\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Case Normalization (Büyük/Küçük Harf Normalleştirme):
the quick brown fox jumps over the lazy dog.

Punctuation Removal (Noktalama İşaretlerinin Kaldırılması):
The quick BROWN Fox Jumps OVER the lazy dog

Stop Word Removal (Durak Kelime Kaldırma):
quick BROWN Fox Jumps lazy dog.

Stemming (Kök Çıkarma):
run,runner,ran

Lemmatization:
run,runner,run

Tokenization (Belirteçleme):
['The', 'quick', 'BROWN', 'Fox', 'Jumps', 'OVER', 'the', 'lazy', 'dog']

Replacing synonyms and Abbreviation to their full form to normalize the text in NLP (Eşanlamlıları ve Kısaltmaları Tam Hâllerine Dönüştürme):
I will be there at 2 pm

Removing numbers and symbol to normalize the text in NLP (Sayıların ve Sembollerin Kaldırılması):
I have  apples and  orange fruits

Removing any remaining non-textual elements to normalize the text in NLP (Kalan Metin Dışı Unsurların Kaldırılması):
Please visit example.com for more information or contact me at infc@example.com

Birleştirilmiş Normalizasyon Fonksiyonu:
quick brown fox jump lazi dog ill  pm pleas visit hrefwwwexamplecomexamplecoma inform contact infoexamplecom
PS C:\Users\FIRAT\Desktop>
```

Sonuç

Bu Kodlama Raporunda Metinlerin Normalizasyon yapılabilmesi için 9 yöntem incelendi. Bu 9 yöntemin her biri için bir örnekle birlikte Python'da kodlarla birlikte test edildi. Her yöntem için Çıktı ile kontrol sağlandı. En sonda Ortak bir metinde 9 normalizasyon yöntemi birleştirilerek tek bir metne 9 normalizasyon işlemi yapılsa ne olur? Sorusuna cevap alabilmek amacıyla bir fonksiyon oluşturdu.

Bu çalışma ve kodlamalar sonucunda Text Normalization hakkında kısaca şunlar söylenebilir: NLP uygulamalarında temel bir ön işleme adımıdır ve metin verilerinin daha tutarlı ve işlenebilir bir formatta olmasını sağlar. Buradaki normalizasyon teknikleri, geniş bir metin işleme yelpazesinde kullanılabilir ve NLP projelerinde veri hazırlama aşamasında önemli bir rol oynar.