**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.

**Kullanılan Kütüphaneler ve Araçlar**

**NLTK** (Natural Language Toolkit): Metin normalizasyonunda yaygın olarak kullanılan bir NLP kütüphanesi.

**string**: Python dilinde bulunan string işlemleri için yerleşik kütüphane.

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

**Kullanılan Normalizasyon Teknikleri**

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

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

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

Kök Çıkarma (**Stemming**): Kelimelerin köklerinin çıkarılması.

**Lemmatization**: Kelimelerin sözlükteki köklerine dönüştürülmesi.

Belirteçleme (**Tokenization**): Metnin kelimelere veya belirteçlere bölünmesi.

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

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

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

**Kodlama**

# nltk kütüphanesinden gerekli modülleri import ediyoruz

# nltk ingilizce NLP için Python kütüphanesidir.

from nltk.stem import PorterStemmer, WordNetLemmatizer

import string

import re

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# 1) Case Normalization (Büyük/Küçük Harf Normalleştirme)

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)

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)

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)

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

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)

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)

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)

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)

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](mailto: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**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**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.