Procesamiento de Información 2023

Unidad 4 - Tarea ¶

David Aarón Ramírez Olmeda

Introducción

El objetivo de esta tarea es detectar el plagio entre documentos usando la medida de similitud de Jaccard sin pesado y la medida de similitud de Dice sin pesado. Se proporcionó un conjunto de documentos y se pidió que se seleccionaran los 3 archivos más similares a partir de una muestra de 20 archivos de origen.

La medida de similitud de Jaccard, también conocida como coeficiente de Jaccard, hay dos versiones la primera que considera vectores binarios de características (está presente la propiedad o no lo está) y su versión de pesos asociados a cada una de las características.

$$Jaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Otra medida de similitud, es la similitud Dice, también conocida como coeficiente de Dice, al igual que Jaccard es popular para vectores binarios (presencia o ausencia de características), o en su forma de conjuntos. Esta medida de similitud toma valores entre 0 y 1. Un valor de 1 es la máxima similitud y 0 total disimilitud.

$$Dice(X, Y) = 2 * \frac{|X \cap Y|}{|X| + |Y|}$$

Desarrollo

Para resolver esta tarea, a grandes rasgos se siguieron los pasos:

- Leer archivos y obtener una muestra de los originales.
- Calcular las medidas de similitud.
- Ordena las similitudes de mayor a menor y selecciona los 3 archivos más similares.
- Muestra el texto original de los documentos para demostrar el plagio y comparar entre medidas

In [1]:
```
import os
import random
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
import numpy as np
from collections import defaultdict
from itertools import combinations
```

In [2]:
```
#nltk.download("stopwords")
stop_words = set(stopwords.words("english"))
stemmer = SnowballStemmer("english")
#nltk.download('punkt')
```

In [3]:
```
def load_files(directory):
    """
    Carga todos los archivos de un directorio y los guarda en una lista.
    """
    files = []
    for filename in os.listdir(directory):
        if filename.endswith(".txt"):
            filepath = os.path.join(directory, filename)
            with open(filepath, "r", encoding="utf-8") as f:
                text = f.read()
                files.append((filename, text))
    return files
```

In [4]:
```
source_dir = "source-documents"
suspicious_dir = "suspicious-documents"

muestra = 20
source_files = load_files("source-documents")
source_sample = random.sample(source_files, muestra)
suspicious_files = load_files("suspicious-documents")
```

In [5]:
```
def preprocess(text):
    text = text.lower()
    text = re.sub(r"[^a-z0-9]+", " ", text)
    return text.strip()
```

In [6]:
```
def preprocess(text):
    text = re.sub(r"[^a-zA-Z\s]", "", text)
    text = text.lower()
    tokens = nltk.word_tokenize(text)
    tokens = [token for token in tokens if token not in stop_words]
    tokens = [stemmer.stem(token) for token in tokens]
    return " ".join(tokens)
```

In [7]:
```
def jaccard_similarity(text1, text2):
    set1 = set(text1.split())
    set2 = set(text2.split())
    intersection = set1.intersection(set2)
    union = set1.union(set2)
    similarity = len(intersection) / len(union)
    return similarity

def dice_similarity(text1, text2):
    text1_tokens = set(text1.split())
    text2_tokens = set(text2.split())
    intersection = len(text1_tokens.intersection(text2_tokens))
    similarity = (2.0 * intersection) / (len(text1_tokens) + len(text2_to
    return similarity
```

In [8]:
```
similarities = []
for source_file in source_sample:
    for suspicious_file in suspicious_files:
        source_text = preprocess(source_file[1])
        suspicious_text = preprocess(suspicious_file[1])
        similarity = jaccard_similarity(source_text, suspicious_text)
        similarities.append((source_file[0], suspicious_file[0], similari
```

In [9]:
```
# Ordenar los resultados de similitud de mayor a menor
similarities.sort(key=lambda x: x[2], reverse=True)
```

In [10]:
```
# Seleccionar los 3 archivos más parecidos y mostrar el texto original de
for i in range(3):
    source_filename = similarities[i][0]
    suspicious_filename = similarities[i][1]
    similarity = similarities[i][2]
    with open(os.path.join("source-documents", source_filename), "r", enc
        source_text = f.read()
    with open(os.path.join("suspicious-documents", suspicious_filename),
        suspicious_text = f.read()
    print(f"Los documentos '{source_filename}' y '{suspicious_filename}'
    print()
    print("Texto original del documento fuente:")
    print(source_text[:150] + "...")
    print("Texto original del documento sospechoso:")
    print(suspicious_text[:150] + "...")
    print()
    print()
```

```
Los documentos 'source-document0043.txt' y 'suspicious-document0429.tx
t' tienen una similitud de 0.221968:

Texto original del documento fuente:
Forecasters preparing for Thursday's opening of the Atlantic hurricane
season wish they could predict the arrival of new technological help th
ey say m...
Texto original del documento sospechoso:
TROY,  Mich.  ( AP )--  Delphi Automotive Systems Corp. , the auto-part
s manufacturer soon to be independent from  General Motors Corp. , has
no more ...


Los documentos 'source-document0236.txt' y 'suspicious-document2359.tx
t' tienen una similitud de 0.193906:

Texto original del documento fuente:
The accidental shooting death of a young stockbroker by an officer look
ing for a burglar is one more strike against a police force already str
uggling ...
Texto original del documento sospechoso:
Kansas City _ French investigators looking into the crash June of an  A
ir France Concorde  said May it was probable that a 16-inch piece of me
tal foun...


Los documentos 'source-document0194.txt' y 'suspicious-document1939.tx
t' tienen una similitud de 0.190349:

Texto original del documento fuente:
The 1990 Atlantic hurricane season begins today amid dire warnings that
killer storms on the East and Gulf coasts in the last two years may hav
e been ...
Texto original del documento sospechoso:
 SEATTLE  _ Once-abundant salmon runs of  Canada 's west coast have suf
fered drastic declines in recent years due to habitat destruction, over
fishing ...
```

In [11]:
```
similarities = []
for source_file in source_sample:
    for suspicious_file in suspicious_files:
        source_text = preprocess(source_file[1])
        suspicious_text = preprocess(suspicious_file[1])
        similarity = dice_similarity(source_text, suspicious_text)
        similarities.append((source_file[0], suspicious_file[0], similari
```

In [12]:
```
# Ordenar los resultados de similitud de mayor a menor
similarities.sort(key=lambda x: x[2], reverse=True)
```

In [13]:
```
# Seleccionar los 3 archivos más parecidos y mostrar el texto original de
for i in range(3):
    source_filename = similarities[i][0]
    suspicious_filename = similarities[i][1]
    similarity = similarities[i][2]
    with open(os.path.join("source-documents", source_filename), "r", enc
        source_text = f.read()
    with open(os.path.join("suspicious-documents", suspicious_filename),
        suspicious_text = f.read()
    print(f"Los documentos '{source_filename}' y '{suspicious_filename}'
    print()
    print("Texto original del documento fuente:")
    print(source_text[:150] + "...")
    print("Texto original del documento sospechoso:")
    print(suspicious_text[:150] + "...")
    print()
    print()
```

```
Los documentos 'source-document0043.txt' y 'suspicious-document0429.tx
t' tienen una similitud de 0.363296:

Texto original del documento fuente:
Forecasters preparing for Thursday's opening of the Atlantic hurricane
season wish they could predict the arrival of new technological help th
ey say m...
Texto original del documento sospechoso:
TROY,  Mich.  ( AP )--  Delphi Automotive Systems Corp. , the auto-part
s manufacturer soon to be independent from  General Motors Corp. , has
no more ...


Los documentos 'source-document0236.txt' y 'suspicious-document2359.tx
t' tienen una similitud de 0.324826:

Texto original del documento fuente:
The accidental shooting death of a young stockbroker by an officer look
ing for a burglar is one more strike against a police force already str
uggling ...
Texto original del documento sospechoso:
Kansas City _ French investigators looking into the crash June of an  A
ir France Concorde  said May it was probable that a 16-inch piece of me
tal foun...


Los documentos 'source-document0194.txt' y 'suspicious-document1939.tx
t' tienen una similitud de 0.319820:

Texto original del documento fuente:
The 1990 Atlantic hurricane season begins today amid dire warnings that
killer storms on the East and Gulf coasts in the last two years may hav
e been ...
Texto original del documento sospechoso:
 SEATTLE  _ Once-abundant salmon runs of  Canada 's west coast have suf
fered drastic declines in recent years due to habitat destruction, over
fishing ...
```

La medida de similitud de Jaccard sin pesado utiliza conjuntos para comparar dos textos, mientras que la medida de similitud de Dice sin pesado utiliza la cantidad de tokens compartidos. En general, la medida de similitud de Dice sin pesado es más estricta que la medida de similitud de Jaccard sin pesado, lo que significa que puede ser más difícil para dos documentos ser considerados similares según esta medida.

Conclusión

Hemos detectado el plagio entre documentos sando la medida de similitud de Jaccard sin pesado y la medida de similitud de Dice sin pesado.

Se ha demostrado que ambas medidas pueden ser efectivas para detectar el plagio, aunque la medida de similitud de Dice sin pesado es más estricta que la medida de similitud de Jaccard sin pesado, se pueden apreciar resultados similares en cuanto a las comparaciones de documentos, pero se difiere un poco en la medida en sí. La tarea se ha completado con éxito