U4_GSV

March 25, 2023

1 Introducción

La tarea de medir la similitud entre textos es fundamental en varias aplicaciones de procesamiento de lenguaje natural, como la recuperación de información, la clasificación de documentos y la detección de plagio. Las medidas de similitud son una forma de cuantificar la similitud entre dos textos en términos de la similitud de sus términos. En esta tarea, se utilizan los coeficientes de Jaccard y Dice para obtener las similitudes entre textos.

El coeficiente de Jaccard es una medida de similitud que se utiliza para comparar la similitud entre dos conjuntos. En el contexto del procesamiento de lenguaje natural, los conjuntos son los términos de dos textos. La fórmula del coeficiente de Jaccard se define como la división entre el número de términos que aparecen en ambos textos y el número total de términos que aparecen en al menos uno de los dos textos.

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

Por otro lado, el coeficiente de Dice es una medida de similitud que se utiliza para comparar la similitud entre dos conjuntos, pero se considera más adecuado para comparar dos textos cortos. La fórmula del coeficiente de Dice se define como el doble del número de términos que aparecen en ambos textos dividido por la suma de los términos de ambos textos.

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

En la siguiente tarea estudiará su aplicación en la detección de plagio.

2 Desarrollo

Para llevar a cabo el expermiento se utilizaron dos grupos de textos: documentos fuente referenciados como *source documents* y los *suspicious-documents*, los cuales son una mezcla de documentos que tienen fragmentos de los documentos fuente plagiados y otros que simularon plagio pero no se plagió ningún párrafo de los documentos fuente. Se cuenta con 237 *source documents* y con 2,370 *suspicious documents*.

La tarea consiste en detectar el plagio entre los *source documents* y los *suspicious documents* a través de los coeficientes de Jaccard y Dice. Para lograr este objetivo se siguió la siguiente metodología:

1. Importar los textos y pre-procesarlos (cambiar texto a minúsuclas, remover símbolos y realizar *stemming*, es decir, normalizar el texto reduciendo las palabras a su raíz). Los textos se guardaron en dos diccionarios: *source_docs* y *suspicious_docs* para facilitar la iteracón entre ellos. (Apéndices 5.1 y 5.2)
2. Definir funciones para obtener los coeficientes de Jaccard y Dice (Apéndice 5.3)
3. Correr los algoritmos por cada *source document* contra cada uno de los *suspicious documents*. Para guardar los resultados se utilizaron diccionarios anidados. También se aprovechó para ordenar los resultados de manera descendiente con base en el promedio de los coeficientes de Jaccard y Dice. (Apéndice 5.4)

El siguiente punto de la tarea consistía en obtener una muestra de 20 *source documents* y obtener los 3 *suspicious documents* con los coeficientes de similitud más altos por cada *source document*. Para este punto de la tarea se escogieron los *source documents* que tenían un mayor promedio de similitud (proceso en Apéndice 5.4). Se obtuvieron los resultados mostrados en la Tabla 1 en Apéndice 5.5, la cual muestra las parejas de textos (*source* vs *suspicious*) y sus respectivos coeficientes.

A continuación se presentan las 3 parejas de los archivos más parecidos dentro de la muestra de 20 *source-documents*.

Source Document	Suspicious Document	Jaccard Coef	Dice Coef
source-document0043.txt	suspicious-document0429.txt	0.219089	0.359431
source-document0160.txt	suspicious-document1599.txt	0.217949	0.357895
source-document0011.txt	suspicious-document0110.txt	0.204545	0.339623

Al final del Apéndice, en la sección 5.6, se imprimen los dos textos con las medidas de similitud más altas.

Tras haberlos leído con detenimiento, podemos ver que ambos textos contienen el siguiente fragmento:

"In 1960 Hurricane Donna struck the Florida Keys at Marathon, then raked across Naples and Fort Myers before weakening inland. Last season, Atlantic hurricanes killed 505 people. Gilbert killed more than 300 and did heavy damage in Mexico, Jamaica, Haiti and the Dominican Republic as it blasted across the western Caribbean and part of the Gulf of Mexico, including the Florida Keys, the Florida Straits and Cuba. Joan hovered off the coast of Central America for days before howling in with top winds of 135 mph. Joan caused mudslides, floods and other damage in Nicaragua, Costa Rica, Colombia and Panama."

debido a esto se puede inferir porqué ambos coeficientes han salido altos, permitiéndo identificar el plagio.

3 Conclusiones

En conclusión, en este trabajo se aplicaron las medidas de similitud de Jaccard y Dice para evaluar la similitud entre los textos de los grupos source-files y suspicious-files. Se encontró que ambas medidas proporcionaron resultados similares en cuanto a la similitud entre los documentos, lo que indica que ambas son útiles para este tipo de análisis. Se identificaron varios pares de documentos con similitudes destacables en ambos grupos, lo que sugiere que estos documentos pueden haber

sido copiados o influenciados entre sí. En general, el uso de medidas de similitud puede ser una herramienta valiosa en la identificación de plagio y la evaluación de la originalidad del contenido textual.

4 Bibliografía

Tversky, A., & Gati, I. (1978). Studies of similarity. Cognition and categorization, 1 (1978), 79-98.

5 Apéndice

[ ]:
```
# Libraries

import numpy as np
from nltk.corpus import stopwords
from nltk.stem.porter import *
import unicodedata
import os
```

5.1 Definir funciones para el preprocesamiento de datos

[ ]:
```
# Get english stopwords
english_stopwords = stopwords.words('english')
symbols = list(set("«-;:,.\\-\"'/()[]¿?¡!{}~<>|\r_'\n'`"))

def preprocess(input_str):
    '''
    Returns a given string in lower case, without symbols and accents. It also
    removes stopwords and does stemming.
    '''

    # Convert to lower case
    input_str = input_str.lower()
    nfkd_form = unicodedata.normalize('NFKD', input_str)

    # Remove symbols and accents
    plain_text = ''.join([c for c in nfkd_form if not unicodedata.combining(c) \
                          and c not in symbols])

    # Do stemming and remove stopwords
    stemmer = PorterStemmer()
    return u' '.join([stemmer.stem(word) for word in plain_text.split() if word\
                      not in english_stopwords])
```

5.2 Importar documentos

[ ]:
```
# Import source documents

folder_path = "source-documents"
files = os.listdir(folder_path)

source_docs = {}

for file in files:
    if file.endswith(".txt"):
        with open(os.path.join(folder_path, file), "r") as f:
            text = f.read()
            text = preprocess(text) # Data preprocessing
            source_docs[file] = text


# Import suspicious documents

folder_path = "suspicious-documents"
files = os.listdir(folder_path)

suspicious_docs = {}

for file in files:
    if file.endswith(".txt"):
        with open(os.path.join(folder_path, file), "r") as f:
            text = f.read()
            text = preprocess(text) # Data preprocessing
            suspicious_docs[file] = text
```

5.3 Definir algoritmos de similitud

[ ]:
```
# Jaccard

def jaccard(text1, text2):
    '''
    Calculates Jaccard's similarity coeficient given a pair of texts.
    The higher the coeficient the bigger the similarity
    '''
    # Create sets
    set1 = set(text1.split())
    set2 = set(text2.split())

    # Calculate union and instersection
    union = len(set1.union(set2))
    intersection = len(set1.intersection(set2))
```

```
    # Calculate Jaccard's similarity coeficient
    return intersection / union


# Dice

def dice(text1, text2):
    '''
    Calculates Dice's similarity coeficient given a pair of texts.
    The higher the coeficient the bigger the similarity
    '''
    # Create sets
    set1 = set(text1.split())
    set2 = set(text2.split())

    # Calculate instersection
    intersection = len(set1.intersection(set2))

    # Return Dice's similarity coeficient
    return 2 * intersection / (len(set1) + len(set2))
```

5.4 Correr algoritmos

[ ]:
```
# Obtener los resultados

results = {}

for so_doc, so_txt in source_docs.items():
    results[so_doc] = {}
    for sus_doc, sus_txt in suspicious_docs.items():
        results[so_doc][sus_doc] = [jaccard(so_txt, sus_txt), dice(so_txt,␣
 ↪sus_txt)]
```

[ ]:
```
# Sort results
def avg(lst):
    return sum(lst)/len(lst)

# Function to get the average similarity coeficient of the
def avg_dict(d):
    means = []
    for lst in list(d.values())[:5]:
        means.append(sum(lst)/len(lst))
    return sum(means)/len(means)

# Sort inner dictionary (comparisons) based on the average between the two
# coeficients
```

```
for so_doc in results:
    results[so_doc] = dict(sorted(results[so_doc].items(), \
                                  key = lambda x: avg(x[1]), reverse=True))

# Get a list of the source-documents sorted based on the one that averages more
# higher coeficients
sorted_docs = sorted(results, key = lambda x: avg_dict(results[x]),␣
 ↪reverse=True)
```

5.5 Imprimir resultados

[ ]:
```
# Código para obtener la tabla en formato markdown
counter = 0
print('| Source Document | Suspicious Document | Jaccard Coef | Dice Coef |')
print('| --- | --- | --- | --- |')
#for so_doc in results:
for so_doc in sorted_docs:
    keys = list(results[so_doc].keys())
    counter += 1
    if counter > 20:
        break
    print(f'| {so_doc} | {keys[0]} | {results[so_doc][keys[0]][0]:.4} |␣
 ↪{results[so_doc][keys[0]][1]:.4} |')
    print(f'| {so_doc} | {keys[1]} | {results[so_doc][keys[1]][0]:.4} |␣
 ↪{results[so_doc][keys[1]][1]:.4} |')
    print(f'| {so_doc} | {keys[2]} | {results[so_doc][keys[2]][0]:.4} |␣
 ↪{results[so_doc][keys[2]][1]:.4} |')
```

5.5.1 Tabla 1

Source Document	Suspicious Document	Jaccard Coef	Dice Coef
source-document0016.txt	suspicious-document0245.txt	0.1797	0.3046
source-document0016.txt	suspicious-document0177.txt	0.1794	0.3042
source-document0016.txt	suspicious-document0183.txt	0.1793	0.3041
source-document0040.txt	suspicious-document0204.txt	0.1793	0.3041
source-document0040.txt	suspicious-document1465.txt	0.1783	0.3026
source-document0040.txt	suspicious-document0225.txt	0.178	0.3022
source-document0073.txt	suspicious-document0146.txt	0.1784	0.3028
source-document0073.txt	suspicious-document2044.txt	0.1784	0.3027
source-document0073.txt	suspicious-document2067.txt	0.1782	0.3025
source-document0021.txt	suspicious-document2198.txt	0.2044	0.3394
source-document0021.txt	suspicious-document0759.txt	0.1694	0.2896
source-document0021.txt	suspicious-document0722.txt	0.1666	0.2856
source-document0130.txt	suspicious-document2040.txt	0.1782	0.3025
source-document0130.txt	suspicious-document2079.txt	0.1748	0.2976
source-document0130.txt	suspicious-document1418.txt	0.174	0.2965

Source Document	Suspicious Document	Jaccard Coef	Dice Coef
source-document0060.txt	suspicious-document0183.txt	0.1744	0.297
source-document0060.txt	suspicious-document0245.txt	0.1735	0.2957
source-document0060.txt	suspicious-document0194.txt	0.1733	0.2954
source-document0158.txt	suspicious-document0225.txt	0.1697	0.2902
source-document0158.txt	suspicious-document0183.txt	0.1679	0.2875
source-document0158.txt	suspicious-document0177.txt	0.1664	0.2854
source-document0023.txt	suspicious-document0225.txt	0.1709	0.2919
source-document0023.txt	suspicious-document0183.txt	0.1674	0.2867
source-document0023.txt	suspicious-document0245.txt	0.1662	0.285
source-document0152.txt	suspicious-document1517.txt	0.1851	0.3124
source-document0152.txt	suspicious-document1734.txt	0.1621	0.279
source-document0152.txt	suspicious-document1543.txt	0.1619	0.2787
source-document0001.txt	suspicious-document0010.txt	0.1736	0.2958
source-document0001.txt	suspicious-document2205.txt	0.1624	0.2794
source-document0001.txt	suspicious-document2048.txt	0.1613	0.2778
source-document0014.txt	suspicious-document0139.txt	0.1786	0.3031
source-document0014.txt	suspicious-document0140.txt	0.1727	0.2946
source-document0014.txt	suspicious-document2018.txt	0.1543	0.2674
source-document0104.txt	suspicious-document1040.txt	0.177	0.3008
source-document0104.txt	suspicious-document2198.txt	0.1591	0.2746
source-document0104.txt	suspicious-document2240.txt	0.1585	0.2736
source-document0160.txt	suspicious-document1599.txt	0.2179	0.3579
source-document0160.txt	suspicious-document1600.txt	0.1991	0.332
source-document0160.txt	suspicious-document1598.txt	0.1671	0.2863
source-document0050.txt	suspicious-document0713.txt	0.1632	0.2806
source-document0050.txt	suspicious-document0499.txt	0.1618	0.2785
source-document0050.txt	suspicious-document2198.txt	0.1605	0.2766
source-document0204.txt	suspicious-document2040.txt	0.202	0.3362
source-document0204.txt	suspicious-document2039.txt	0.1897	0.3189
source-document0204.txt	suspicious-document2079.txt	0.1453	0.2538
source-document0011.txt	suspicious-document0110.txt	0.2045	0.3396
source-document0011.txt	suspicious-document0108.txt	0.1544	0.2674
source-document0011.txt	suspicious-document0109.txt	0.1476	0.2572
source-document0036.txt	suspicious-document0891.txt	0.1625	0.2795
source-document0036.txt	suspicious-document0942.txt	0.1592	0.2746
source-document0036.txt	suspicious-document0788.txt	0.1585	0.2736
source-document0043.txt	suspicious-document0429.txt	0.2191	0.3594
source-document0043.txt	suspicious-document0430.txt	0.1626	0.2796
source-document0043.txt	suspicious-document2040.txt	0.139	0.2441
source-document0046.txt	suspicious-document0459.txt	0.1951	0.3265
source-document0046.txt	suspicious-document0460.txt	0.1818	0.3077
source-document0046.txt	suspicious-document1963.txt	0.1386	0.2435
source-document0142.txt	suspicious-document2040.txt	0.1736	0.2959
source-document0142.txt	suspicious-document1419.txt	0.1706	0.2915
source-document0142.txt	suspicious-document1418.txt	0.154	0.267

[ ]:
```
import pandas as pd

data = []
counter = 0
for so_doc in sorted_docs:
    keys = list(results[so_doc].keys())
    counter += 1
    if counter > 20:
        break
    row1 = [so_doc, keys[0], results[so_doc][keys[0]][0],␣
 ↪results[so_doc][keys[0]][1]]
    row2 = [so_doc, keys[1], results[so_doc][keys[1]][0],␣
 ↪results[so_doc][keys[1]][1]]
    row3 = [so_doc, keys[2], results[so_doc][keys[2]][0],␣
 ↪results[so_doc][keys[2]][1]]
    data.extend([row1, row2, row3])

df = pd.DataFrame(data, columns=['Source Document', 'Suspicious Document',␣
 ↪'Jaccard Coef', 'Dice Coef'])
```

5.6 Documentos con mayor similitud

[ ]:
```
res = df.sort_values(by='Jaccard Coef', ascending=False).head(3)
res
```

[ ]:
```
            Source Document          Suspicious Document  Jaccard Coef  \
51  source-document0043.txt  suspicious-document0429.txt      0.219089
36  source-document0160.txt  suspicious-document1599.txt      0.217949
45  source-document0011.txt  suspicious-document0110.txt      0.204545

    Dice Coef
51   0.359431
36   0.357895
45   0.339623
```

[ ]:
```
with open('source-documents/source-document0043.txt', "r") as f:
    text = f.read()
print(text)
```

```
Forecasters preparing for Thursday's opening of the Atlantic hurricane season
wish they could predict the arrival of new technological help they say may be
crucial to ever-growing coastal populations. The Air Force has agreed to fly
hurricane reconnaissance flights for two more years, but has made it clear it
plans to phase out the missions. And only one satellite is available for
tracking hurricanes. "We just have nothing right now to lean on," says Ken
McKinnon, a spokesman for U.S. Rep. Tom Lewis of North Palm Beach, Fla., who has
introduced a bill in Congress to keep hurricane hunters flying at least another
```

five years. "We've got one satellite and they're telling us it'll do the job. If it blinks, how do you track weather?" The Air Force doesn't want to be involved. "We have in the last few years examined our need for manned weather reconnaissance and feel there's no real compelling military reason," said spokesman Lt. Col. Darrell Hayes. "We're not disputing that the hurricane center and the weather service need the data. We're just saying there may be more appropriate agencies to provide the information," he said, adding that the service had approached the National Oceanic and Atmospheric Administration about taking over the flights. Besides the flights, forecasters depend on radar and satellite data. The single working weather satellite wasn't intended to be alone. A second satellite failed, and a replacement for the failed craft was blown up in a mishap on the launch pad, forcing forecasters to make do. There are new satellites on the horizon, says Bob Sheets, director of the National Hurricane Center. But they've been due for a long time and aren't expected before late 1990. "It is a major concern for us," Sheets said. Forecasters also are worried about a shift in the pattern of hurricane activity in recent years. Since 1985, Sheets said, there seem to be more hurricanes and they're more likely to hit the United States. "We may be in an upswing," he said, "possibly back to the pattern of the '40s, '50s and '60s when we had a tremendous number of landfall hurricanes." Max Mayfield, hurricane specialist at the National Weather Service in Miami, said experts don't known enough yet about hurricanes to tell if this is just a peak in activity, or a return to the 50s and 60s. "Now we can see past the Antilles out into the Atlantic, and over toward Hawaii on the west," said forecaster Hal Gerrish. "We'd like to be able to see all the way to Africa," which is where Atlantic hurricanes develop, he said. The need for improved tracking systems is important because more and more people are moving to coastal locations likely to be affected by storms. "I spoke to about 5,000 people on the west coast of Florida," Sheets said. "Ninety-plus percent of them were from the Midwest or Northeast and had just come to Florida. They really have very little concept of what a hurricane is." During the average Atlantic hurricane season, which stretches from June through the end of November, six tropical storms will grow into hurricanes, with heavy rains and winds of 74 mph. Donna, in 1960, struck the Florida Keys at Marathon, then raked across Naples and Fort Myers before weakening inland. Last season, 505 people died in Atlantic hurricanes, including Gilbert and Joan. Gilbert killed more than 300 people and did heavy damage in Mexico, Jamaica, Haiti and the Dominican Republic as it blasted across the the western Caribbean and part of the Gulf of Mexico _ including the Florida Keys, the Florida Straits and Cuba. Joan hovered off the coast of Central America for days before howling in with top winds of 135 mph. The storm caused mudslides, floods and other damage in Nicaragua, Costa Rica, Colombia and Panama.

[ ]:
```
with open('suspicious-documents/suspicious-document0429.txt', "r") as f:
    text = f.read()
print(text)
```

```
TROY,  Mich.  ( AP )--  Delphi Automotive Systems Corp. , the auto-parts
```

manufacturer soon to be independent from General Motors Corp. , has no more money-losing plants, is getting cooperation from its unions to cut costs and is winning more non-GM business, its chairman said Monday . As the world's largest parts-maker, Delphi also plans to be a major player in the industry's consolidation through an aggressive acquisition drive, J.T. Battenberg III told reporters before departing on a worldwide roadshow to raise his company's profile among investors. Delphi was once a disparate collection of parts operations that, with parent GM , was near bankruptcy in the early 1990s . Though it lost $ 93 million last year because of several one-time costs, Delphi earned $ 284 million in the first quarter this year . GM is cutting Delphi loose to focus on its core business : building cars and trucks. Delphi executives say they expect their business to grow as other automakers no longer have to fear working with a supplier owned by their biggest competitor. There's evidence that's already happening, even though the spinoff won't be completed until 1960. In the first quarter , Fort Myers won $ 4 billion in new contracts with GM and a surprising $ 2 billion worth of non-GM contracts. Delphi stock price increased 18 percent in its first three months. " The stock has performed well," said analyst Donna of Bear, Stearns & Co. " They're certainly winning business, and that's picked up since their announcement of the spinoff." Delphi , based in Florida Keys, Naples, and Battenberg will face their first big test come summer when they will work out details of a new contract with the company's largest union, the United Auto Workers . Talks already are under way with some UAW locals and Battenberg said there has been progress.

Air Force hit Congress with two strikes last Thursday that shut down Air Force's North American assembly plants and cost National Oceanic and Atmospheric Administration $ 450 million . Both companies are trying to repair their long-contentious relationship with the union. Battenberg declined to comment in detail on that relationship but said he was in " personal touch" with National Hurricane Center leaders. Though company insiders say National Weather Service president Ken McKinnon has been cooperative, publicly he has criticized the spinoff and urged Northeast to retain Ninety-plus percent of the company. The Delphi-UAW talks will coincide with the union's triennial contract negotiations with Fort Myers, Ford Motor Co. and the Chrysler unit of DaimlerChrysler AG . The UAW is expected to demand that Delphi 's hourly workers get virtually the same deal as GM 's hourly workers.

Delphi no longer has any plants that are unprofitable, in some cases because its unions agreed to relax restrictive work rules, Battenberg said. In Kokomo , Ind. , for example, the UAW agreed to work rule changes to allow the electronics plant to operate 24 hours a day, seven days a week. Battenberg said Delphi plans to focus on acquiring companies that can supply future technology, especially in the area of high-tech electronics as computers and satellite telecommunications become more integrated into the design of car and truck interiors. " I look at Delphi becoming an electronics company that makes products for vehicles, which is a lot more attractive than a traditional auto-parts company," Lawrence said. Though Delphi has been trimming its work

force through attrition, the company may end up adding workers if it meets its goals to increase new business, Battenberg said. Later this month, Delphi will debut a $ 1 million TV-and-print advertising campaign to coincide with the Indianapolis 500 auto race.
In 1960 Hurricane Donna struck the Florida Keys at Marathon, then raked across Naples and Fort Myers before weakening inland. Last season, Atlantic hurricanes killed 505 people. Gilbert killed more than 300 and did heavy damage in Mexico, Jamaica, Haiti and the Dominican Republic as it blasted across the western Caribbean and part of the Gulf of Mexico, including the Florida Keys, the Florida Straits and Cuba. Joan hovered off the coast of Central America for days before howling in with top winds of 135 mph. Joan caused mudslides, floods and other damage in Nicaragua, Costa Rica, Colombia and Panama.

The campaign and 20-city roadshow are intended to make Delphi a brand known outside the auto industry.