Kin Nevárez
19 de mayo 2022
*Facultad de Ciencias Físico Matemáticas*
*Maestría en Ciencia de Datos*

# Tarea 1 - Pre procesamiento de texto

***Resumen*—El reporte a continuación es una ejemplificación de los tipos de pre procesamiento que se pueden realizar para un correcto y oportuno análisis de texto, el cual, se puede observar en gráficas de frecuencia y estadísticas de palabras más usadas.**

***Palabras clave:*** *análisis de texto, tf-idf, wordcloud, gutenberg.*

## I. Introducción

La analítica de texto (minería de texto o text mining) engloba al conjunto de técnicas que permiten estructurar la información heterogénea presente en los textos con el objetivo de identificar patrones tales como el uso de palabras, con los que extraer nueva información (J. Amat, 2020).

Los textos utilizados son dos libros obtenidos de la página de https://www.gutenberg.org/ cuyas características son las siguientes:

Texto A
Title: The Amateur Cracksman
Author: E. W. Hornung
Release Date: November, 1996 [EBook #706]
Last updated: October 21, 2021

Texto B
Title: Murder in the Gunroom
Author: Henry Beam Piper
Release Date: February 26, 2006 [EBook #17866]
Last updated: January 27, 2009

Pese a que los libros pertenecen a autores diferentes, ambos son parte del género de Crimen - Ciencia Ficción, por lo que el objetivo del análisis es encontrar las palabras más frecuentes que puedan sugerir la trama o género de los libros, y comparar la frecuencia de palabras utilizadas entre autores. Todos los análisis en código fueron realizados en el lenguaje de programación Python con la herramienta de JupyterLab.

## II. Definiciones

### 1. *tf-idf*

TF-IDF es un método fiable para estimar la relevancia de un documento para un término. Cuanto mayor es la frecuencia de un término en un texto, se consigue un TF IDF alto, y menor es el número de documentos que mencionan ese término.

Este análisis se compone de dos partes TF (Term Frequency) y el IDF (Inverse Document Frequency)

**TF:** Sklearn calcula esta parte como el número de veces que una palabra aparece en un documento, es decir, su frecuencia absoluta. Cada documento tiene su propia frecuencia de términos. Se calcula de la siguiente manera:

$$TF_{i,j} = \sum_{k} n_{i,j} \tag{1}$$

**IDF:** Es una medida de qué tan común o raro es un término a lo largo de todos los documentos. Si la palabra es muy común, el idf (normalizado) se acercará a 0, por el contrario se acercará a 1 si es muy común. La forma en que Sklearn calcula esta métrica es:

$$IDF(t) = log\frac{1+n}{1+df(t)} + 1 \tag{2}$$

donde n es el número total de documentos y df(t) es la frecuencia del término t en el documento.

Y por último, el estadístico TF-IDF se calcula:

$$TFIDF(t,d) = tf(t,d) * idf(t) \tag{3}$$

## III. Metodología

### A. *Tokenización*

El primer tipo de pre procesado es la Tokenización, la cual consta de separar todo el texto en una lista de palabras como se ve en el ejemplo a continuación:

**Fig. 1. Comparativa texto normal vs texto tokenizado**

```
textA=open('textA.txt','r').read().splitlines()
textA=' '.join([str(item) for item in textA])
print(textA[638:1800])
textA=textA[638:]
```

```
THE AMATEUR CRACKSMAN  BY  E. W. HORNUNG      TO  A. C. D.   THIS FORM OF FLATTERY      THE AMATEUR CRACKSMAN
CONTENTS    THE IDES OF MARCH   A COSTUME PIECE   GENTLEMEN AND PLAYERS   LE PREMIER PAS   WILFUL MURDER   NINE P
OINTS OF THE LAW   THE RETURN MATCH   THE GIFT OF THE EMPEROR      THE IDES OF MARCH  I  It was half-past twelve w
hen I returned to the Albany as a last desperate resort.  The scene of my disaster was much as I had left it. The
baccarat-counters still strewed the table, with the empty glasses and the loaded ash-trays.  A window had been op
ened to let the smoke out, and was letting in the fog instead.  Raffles himself had merely discarded his dining j
acket for one of his innumerable blazers.  Yet he arched his eyebrows as though I had dragged him from his bed.
"Forgotten something?" said he, when he saw me on his mat.  "No," said I, pushing past him without ceremony.  And
I led the way into his room with an impudence amazing to myself.  "Not come back for your revenge, have you?  Bec
ause I'm afraid I can't give it to you single-handed.  I was sorry myself that the others--"  We were face to fac
e by his fireside, and I cut
```

```
tokensA = tokenizer.tokenize(textA)
print(tokensA[:200])
```

```
['THE', 'AMATEUR', 'CRACKSMAN', 'BY', 'E', 'W', 'HORNUNG', 'TO', 'A', 'C', 'D', 'THIS', 'FORM', 'OF', 'FLATTERY',
'THE', 'AMATEUR', 'CRACKSMAN', 'CONTENTS', 'THE', 'IDES', 'OF', 'MARCH', 'A', 'COSTUME', 'PIECE', 'GENTLEMEN', 'A
ND', 'PLAYERS', 'LE', 'PREMIER', 'PAS', 'WILFUL', 'MURDER', 'NINE', 'POINTS', 'OF', 'THE', 'LAW', 'THE', 'RETUR
N', 'MATCH', 'THE', 'GIFT', 'OF', 'THE', 'EMPEROR', 'THE', 'IDES', 'OF', 'MARCH', 'I', 'It', 'was', 'half', 'pas
t', 'twelve', 'when', 'I', 'returned', 'to', 'the', 'Albany', 'as', 'a', 'last', 'desperate', 'resort', 'The', 's
cene', 'of', 'my', 'disaster', 'was', 'much', 'as', 'I', 'had', 'left', 'it', 'The', 'baccarat', 'counters', 'sti
ll', 'strewed', 'the', 'table', 'with', 'the', 'empty', 'glasses', 'and', 'the', 'loaded', 'ash', 'trays', 'A',
'window', 'had', 'been', 'opened', 'to', 'let', 'the', 'smoke', 'out', 'and', 'was', 'letting', 'in', 'the', 'fo
g', 'instead', 'Raffles', 'himself', 'had', 'merely', 'discarded', 'his', 'dining', 'jacket', 'for', 'one', 'of',
'his', 'innumerable', 'blazers', 'Yet', 'he', 'arched', 'his', 'eyebrows', 'as', 'though', 'I', 'had', 'dragged',
'him', 'from', 'his', 'bed', 'Forgotten', 'something', 'said', 'he', 'when', 'he', 'saw', 'me', 'on', 'his', 'ma
t', 'No', 'said', 'I', 'pushing', 'past', 'him', 'without', 'ceremony', 'And', 'I', 'led', 'the', 'way', 'into',
'his', 'room', 'with', 'an', 'impudence', 'amazing', 'to', 'myself', 'Not', 'come', 'back', 'for', 'your', 'reven
ge', 'have', 'you', 'Because', 'I', 'm', 'afraid', 'I', 'can', 't', 'give', 'it', 'to', 'you', 'single', 'hande
d', 'I', 'was', 'sorry', 'myself', 'that']
```

Fuente: Elaboración propia

Esto permite tener un mejor control y análisis posteriores.

### *B. "Lower Case"*

Se transforman todas las palabras para que aparezcan únicamente en minúsculas, ya que el lenguaje Python toma las palabras con caracteres en mayúscula como si fueran palabras distintas a las palabras en minúscula.

### *C. Remover "stopwords"*

Se eliminan las palabras más frecuentes en el idioma inglés (stopwords). Éstas no proporcionan información relevante en los textos analizados, pues son utilizadas en todos los documentos, independientemente del género o texto.

**Fig. 2. Stopwords idioma inglés**

```
sw = nltk.corpus.stopwords.words('english')
print(sw[:100])
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'you
r', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself',
'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom',
'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'ha
s', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as',
'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'b
efore', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'agai
n', 'further', 'then', 'once']
```

Fuente: Módulo ntlk, Python

### *D. Lematización*

La lematización es un proceso lingüístico que consiste en, dada una forma flexionada, hallar el lema correspondiente. Esto es encontrar la palabra raíz de una palabra flexionada, como en los siguientes ejemplos:

**Fig. 3. Ejemplo de lematización**

```
lemmatizer = WordNetLemmatizer()

print("sayings :", lemmatizer.lemmatize("sayings"))
print("heights :", lemmatizer.lemmatize("heights"))
```

```
sayings : saying
heights : height
```

Fuente: Módulo ntlk, Python

### *E. Remover caracteres especiales*

Se eliminan todos los caracteres no alfa-numéricos que puedan aparecer, con la función "sub". Para hacer uso de esta función, es más sencillo unir todas las palabras pre procesadas de nuevo a un documento completo.

### *F. Análisis TF-IDF*

Como se mencionó anteriormente, el análisis TF-IDF sirve para conocer la relevancia de los términos en distintos documentos. En este caso, se desea obtener la relevancia en dos documentos de crimen-ciencia ficción para compararlas y ver si es posible identificar el género con el simple hecho de ver las palabras utilizadas o darse una idea de lo que se puede tratar.

Después de realizar los cálculos, las 10 palabras más relevantes son:

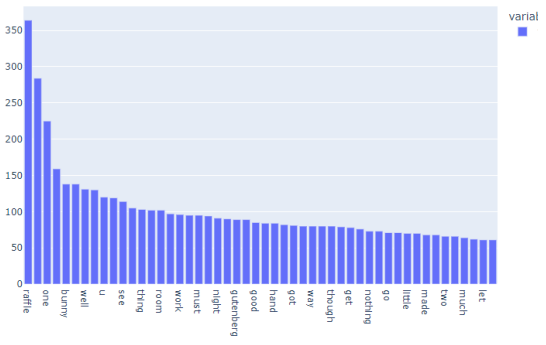
TABLA I
TF-IDF

| Término | tf-idf |
|---|---|
| raffle | 0.4813 |
| said | 0.2672 |
| one | 0.2116 |
| bunny | 0.1824 |
| would | 0.1495 |
| man | 0.1298 |
| well | 0.1232 |
| know | 0.1223 |
| could | 0.1119 |
| see | 0.1072 |

### *G. Gráficos*

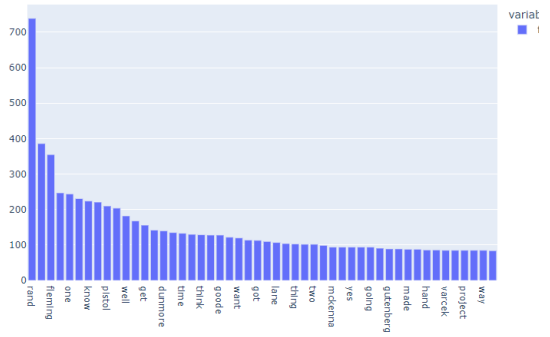
Después de obtener la métrica de TF-IDF en el apartado anterior, se realizan gráficos para visualizar mejor la información:

**Fig. 4. Gráfico frecuencia de palabras texto A**




Fuente: Elaboración propia

**Fig. 5. Gráfico frecuencia de palabras texto B**

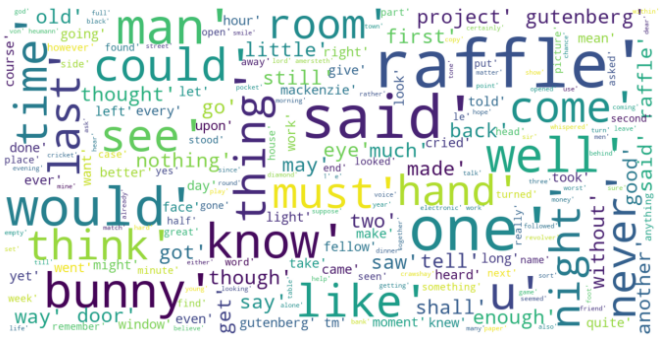



Fuente: Elaboración propia

De los gráficos anteriores, se observa que algunas de las palabras más frecuentes se pueden tomar como “stopwords” para el caso en cuestión. Un ejemplo de esto sería el término “gutenberg” que puede aparecer varias veces por ser la fuente de información de los textos. Otras palabras no relevantes podrían ser “would”, “could”, etc. Para futuros análisis se podrían eliminar.

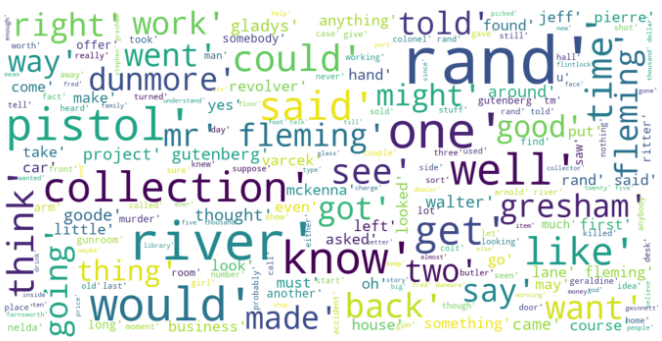
Observando los dos gráficos de frecuencias, se podría inferir que el autor del texto A tiene un léxico más variado porque incluso sus palabras más utilizadas tienen apenas la mitad de frecuencia del autor del texto B. Adicionalmente, es más fácil detectar en el texto B, aún sin necesidad de leerlo, que se trata de un texto de crimen por las palabras "pistol" y "revolver". Por otra parte, el autor del texto A utiliza más palabras como "night", "eye", "saw" que podrían dar un indicio de suspenso.

**Fig. 6. Wordcloud texto A**




Fuente: Elaboración propia

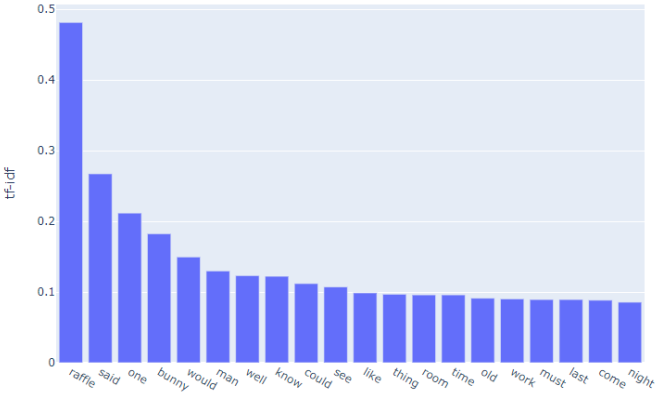
**Fig. 7. Wordcloud texto B**




Fuente: Elaboración propia

Las imágenes anteriores son otra forma de mostrar la frecuencia de palabras, utilizando el tamaño para hacer notar la frecuencia. Con esto también es sencillo darse una idea del género al que pertenece el segundo texto.

**Fig. 8. Gráfico tf-idf**




Fuente: Elaboración propia

El gráfico anterior representa en conjunto en los dos documentos (texto A y texto B) las palabras más frecuentes de ambos autores en sus libros. Se puede observar que no hay algún indicio del género dentro de estas palabras, con la excepción de "night" que apenas puede indicar que las escenas suceden durante la noche.

## IV. Conclusiones

El pre procesamiento de texto es crucial para el análisis o minería de datos, pues es necesario pulir muy bien cualquier inconsistencia que pueda tener, ya sea en la parte técnica, como caracteres especiales, números o información que no aporta valor al análisis y solo genera ruido; como también en la parte de contexto de negocio, es decir, eliminando frases o partes que se sabe por el contexto del análisis.

## Referencias

[1] E. W. Hornung. (2021). The Amateur Cracksman. Project Gutenberg. Obtenido el 16 de Mayo de 2022. Disponible en: https://www.gutenberg.org/files/706/706-h/706-h.htm.

[2] Beam Piper, H. (2009). Murder in the Gunroom. New York: Project Gutenberg. Obtenido el 16 de Mayo de 2022. Disponible en: https://www.gutenberg.org/cache/epub/17866/pg17866.txt.

[3] Alejandro Bassi, A. (2020). Lematización basada en análisis no supervisado de corpus. Santiago, Chile: Departamento de Ciencias de la Computación, Universidad de Chile. Obtenido el 16 de Mayo de 2022. Disponible en: https://users.dcc.uchile.cl/~abassi/ecos/lema.html.

[4] Análisis de texto (text mining) con Python by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0) at https://www.cienciadedatos.net/documentos/py25-text-mining-python.html.

[5] https://github.com/KinMichelle/FCFM.git