Post-Clase I

Análisis de frecuencia de término de la Cuenta Pública de Chile (Frecuencia de Término)

¿Cómo podemos identificar las palabras o frases más relevantes en el mensaje anual del presidente a la nación en Chile? El objetivo es discernir las temáticas más prominentes en el discurso utilizando el algoritmo TF.

Entrada: La entrada para este problema sería el texto completo de la Cuenta Pública del presidente en formato string. Cada palabra o frase del discurso puede ser considerada como un término a analizar.

Restricciones: Los términos comunes o "palabras de parada" como "el", "la", "y", "en" deberían ser excluidos del análisis, ya que no aportan significado relevante.

Salida: Un diccionario de las palabras usadas y su frecuencia en el texto (Dict[str, float]) o una nube de palabras.

Ejemplo de Salida : (Cuenta publica 2023)


```
from typing import Dict

def tokenizar(text: str) -> List[str]:
    words = re.findall(r'\b\w+\b', text.lower())
    words = [word for word in words if len(word) > 2 and not word.isnumeric() and word not in stop_words_es]
    return words

def calcular_ft(texto: str) -> Dict[str, float]:
    palabras = tokenizar(texto)
    total_palabras = len(palabras)

    # Contar frecuencia de cada palabra
```

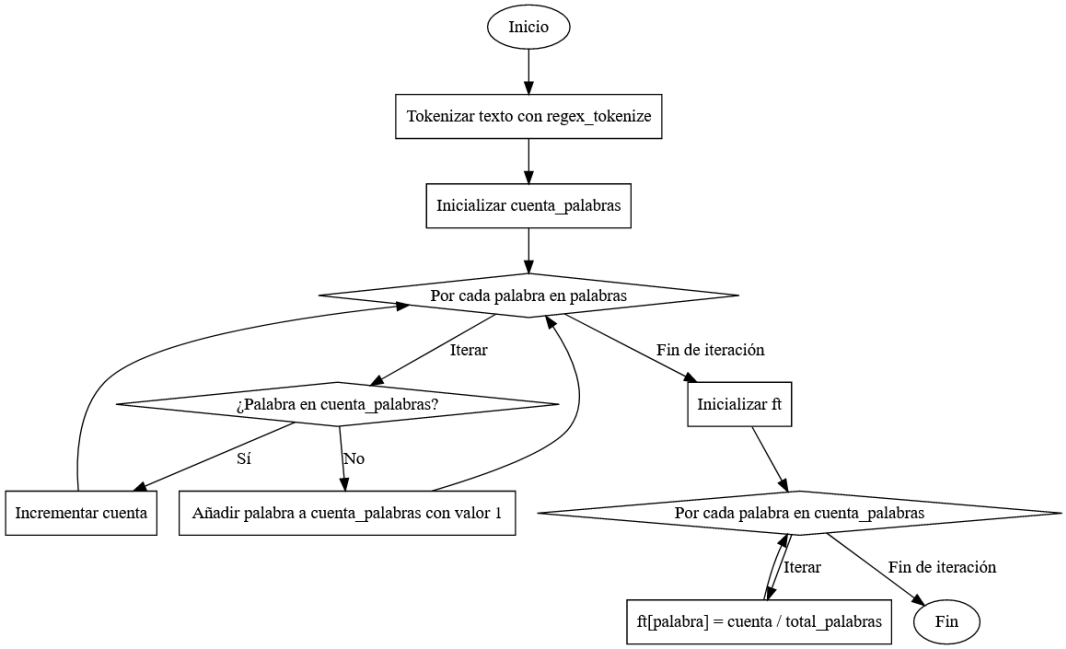
```
    cuenta_palabras = {}
    for palabra in palabras:
        if palabra in cuenta_palabras:
            cuenta_palabras[palabra] += 1
        else:
            cuenta_palabras[palabra] = 1

    # Calcular FT
    ft = {}
    cuenta = 0

    for palabra in cuenta_palabras:
        cuenta = cuenta_palabras[palabra]
        frecuencia_termino = cuenta / total_palabras
        ft[palabra] = frecuencia_termino

    return ft
```

Gráfico del funcionamiento del cálculo de frecuencia de la palabra

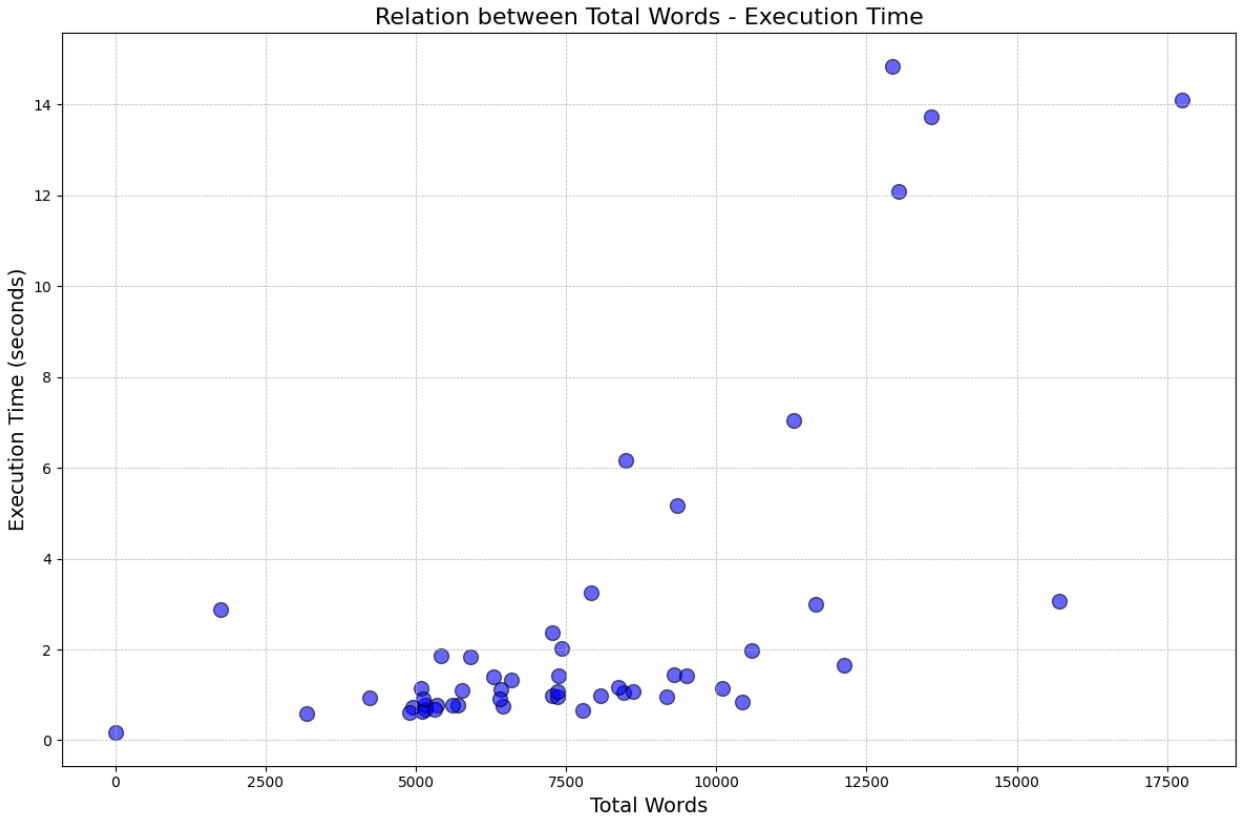


Complejidad temporal de la funcion calcular_ft

1. `tokenizar(texto)`: O(n), donde n es la longitud del texto.
2. `len(palabras)`: O(1).
3. Bucle que llena `cuenta_palabras`: O(m), donde m es el número de palabras.
4. Bucle que llena `ft`: O(k), donde k es el número de palabras únicas.

Dado que k≤m, la complejidad total es O(n)+O(m), lo que se simplifica a O(n+m)

Relación entre el total de palabras y el tiempo de ejecución



Google colab: https://colab.research.google.com/drive/1znrWxNoq0R-58VTOTXQHvemLx0Dsbytc?usp=sharing