```
import pandas as pd
import numpy as np
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
data = pd.read_excel('/content/drive/MyDrive/Datas/diccionario_otros.xlsx')
data = pd.DataFrame(data)
display(data)
```

	palabra	tema
0	acting on impulse	dysregulated emotions
1	anger outbursts	dysregulated emotions
2	difficulty regulating emotions	dysregulated emotions
3	drama	dysregulated emotions
4	dramatic	dysregulated emotions
...	...	...
264	wellbutrin	medication
265	methylphenidate	medication
266	antidepressants	medication
267	stimulants	medication
268	antipsychotics	medication

269 rows × 2 columns

```
import pandas as pd
import json

# Supongamos que tienes un DataFrame llamado df
# Convierte el DataFrame a una lista de diccionarios
data = data.to_dict(orient='records')

# Convierte la lista de diccionarios a formato JSON
json_data = json.dumps(data, indent=4)

# Luego, puedes guardar el JSON en un archivo o hacer lo que desees con él
```

```
# Especifica la ruta y el nombre del archivo donde deseas guardar el JSON
ruta_del_archivo = "/content/drive/MyDrive/Datas/criterios_nuevos.json"

# Abre el archivo en modo escritura y guarda el JSON
with open(ruta_del_archivo, "w") as archivo_json:
    archivo_json.write(json_data)
```

```
import pandas as pd

# Cargar el conjunto de datos
conjunto_de_datos = pd.read_csv('/content/drive/MyDrive/Datas/ADHD-comment.csv')  # Reemplaza con tu archivo de datos
```

```
conjunto_de_datos = conjunto_de_datos.drop_duplicates(['id'])
```

```
conjunto_de_dato2=conjunto_de_datos[conjunto_de_datos['body'] == '[deleted]'].index.to_list()
conjunto_de_datos = conjunto_de_datos.drop(conjunto_de_dato2)
```

```
conjunto_de_datos
```

	body	id	score	created_utc	created_datetime
0	[deleted]	c08otkh	1.0	1.239042e+09	2009-04-06 18:18:07
1	If I try to look this up right now I will get ...	c09y8qz	2.0	1.243790e+09	2009-05-31 17:08:19
2	potassium is used as the thing that stops your...	c09yia6	2.0	1.243815e+09	2009-06-01 00:07:50
3	I've love a link to anything about this. \n\n...	c0a81e6	3.0	1.244752e+09	2009-06-11 20:25:36
4	I don't know anything specific, but I would *d...	c0aixrg	2.0	1.245813e+09	2009-06-24 03:04:51

```
import re

# Define a function to clean the text
def clean(body):
    # Removes all special characters and numericals leaving the alphabets
    body = re.sub('[^A-Za-z]+', ' ', body)
    return body

# Cleaning the text in the review column
conjunto_de_datos['Cleaned Body'] = conjunto_de_datos['body'].apply(clean)
conjunto_de_datos.head()
```

	body	id	score	created_utc	created_datetime	Cleaned Body
0	[deleted]	c08otkh	1.0	1.239042e+09	2009-04-06 18:18:07	deleted
1	If I try to look this up right now I will get ...	c09y8qz	2.0	1.243790e+09	2009-05-31 17:08:19	If I try to look this up right now I will get ...
2	potassium is used as the thing that stops your...	c09yia6	2.0	1.243815e+09	2009-06-01 00:07:50	potassium is used as the thing that stops your...

```
import re

def limpiar_caracteres(texto):
    # Utilizamos una expresión regular para encontrar y eliminar caracteres individuales
    texto_limpio = re.sub(r'\b\w\b', '', texto)

    return texto_limpio


conjunto_de_datos['Cleaned Body'] = conjunto_de_datos['Cleaned Body'].apply(limpiar_caracteres)


comentarios = conjunto_de_datos['Cleaned Body'].dropna()
# Reemplaza "don't" por "dont"
texto_completo = comentarios
texto_completo = texto_completo.replace("don t", "dont")
conjunto_de_datos['Texto_Completo'] = texto_completo
conjunto_de_datos['Texto_Completo'] =conjunto_de_datos['Texto_Completo'].apply(limpiar_caracteres)
conjunto_de_datos
```

	body	id	score	created_utc	created_datetime	Cleaned Body	Text
0	[deleted]	c08otkh	1.0	1.239042e+09	2009-04-06 18:18:07	deleted	
1	If I try to look this up right now I will get ...	c09y8qz	2.0	1.243790e+09	2009-05-31 17:08:19	If try to look this up right now will get di...	If try rigl
2	potassium is used as the thing that stops your...	c09yia6	2.0	1.243815e+09	2009-06-01 00:07:50	potassium is used as the thing that stops your...	pota as
3	I've love a link to anything about this. \n\n...	c0a81e6	3.0	1.244752e+09	2009-06-11 20:25:36	ve love link to anything about this have to...	anyth

```
PP_SDE = conjunto_de_datos[conjunto_de_datos['body'] == '[deleted]'].index.to_list()
conjunto_de_datos = conjunto_de_datos.drop(index=(PP_SDE))

display(conjunto_de_datos)
```

	body	id	score	created_utc	created_datetime	Cleaned Body	Te
1	If I try to look this up right now I will get ...	c09y8qz	2.0	1.243790e+09	2009-05-31 17:08:19	If try to look this up right now will get di...	If t r
2	potassium is used as the thing that stops your...	c09yia6	2.0	1.243815e+09	2009-06-01 00:07:50	potassium is used as the thing that stops your...	po
3	I've love a link to anything about this. \n\n...	c0a81e6	3.0	1.244752e+09	2009-06-11 20:25:36	ve love link to anything about this have to...	an
4	I don't know anything	c0aixrg	2.0	1.245813e+09	2009-06-24 03:04:51	don know anything specific but	do

```
conjunto_de_datos = conjunto_de_datos.drop(['id','created_utc','score'], axis =1)
conjunto_de_datos.head()
```

	body	created_datetime	Cleaned Body	Texto_Completo
1	If I try to look this up right now I will get ...	2009-05-31 17:08:19	If try to look this up right now will get di...	If try to look this up right now will get di...
2	potassium is used as the thing that stops your...	2009-06-01 00:07:50	potassium is used as the thing that stops your...	potassium is used as the thing that stops your...
3	I've love a link to anything about this. \n\n...	2009-06-11 20:25:36	ve love link to anything about this have to...	ve love link to anything about this have to...

```
comentarios = conjunto_de_datos['Cleaned Body'].dropna()
# Reemplaza "don't" por "dont"
texto_completo = comentarios
texto_completo = texto_completo.replace("don t", "dont")
conjunto_de_datos['Texto_Completo'] = texto_completo
conjunto_de_datos['Texto_Completo'] =conjunto_de_datos['Texto_Completo'].apply(limpiar_caracteres)
conjunto_de_datos
```

	body	created_datetime	Cleaned Body	Texto_Completo
1	If I try to look this up right now I will get ...	2009-05-31 17:08:19	If try to look this up right now will get di...	If try to look this up right now will get di...
2	potassium is used as the thing that stops your...	2009-06-01 00:07:50	potassium is used as the thing that stops your...	potassium is used as the thing that stops your...
3	I've love a link to anything about this. \n\n...	2009-06-11 20:25:36	ve love link to anything about this have to...	ve love link to anything about this have to...
4	I don't know anything specific, but I would *d...	2009-06-24 03:04:51	don know anything specific but would defini...	don know anything specific but would defini...
5	Despite continued controversy, powerful new ev...	2009-11-09 16:12:44	Despite continued controversy powerful new evi...	Despite continued controversy powerful new evi...
...	...	...	...	...

```
!pip install pyspellchecker
```

```
Collecting pyspellchecker
  Downloading pyspellchecker-0.7.2-py3-none-any.whl (3.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 34.9 MB/s eta 0:00:00
Installing collected packages: pyspellchecker
Successfully installed pyspellchecker-0.7.2
```

```
!pip install rapidfuzz
```

```
Collecting rapidfuzz
  Downloading rapidfuzz-3.5.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 33.6 MB/s eta 0:00:00
Installing collected packages: rapidfuzz
Successfully installed rapidfuzz-3.5.2
```

```
import pandas as pd
from rapidfuzz import fuzz
from rapidfuzz import process
```

```
# Carga tu diccionario desde un archivo JSON (reemplaza con tu ruta)
with open("/content/drive/MyDrive/Datas/criterios_nuevos.json", "r") as json_file:
    diccionario = json.load(json_file)
```

```
opciones = [objeto['palabra'] for objeto in diccionario]
```

```
from joblib import Parallel, delayed
```

```
import time
from joblib import Parallel, delayed

# Crea un diccionario para mapear comentarios a índices
comentario_a_indice = {comentario: indice for indice, comentario in enumerate(conjunto_de_datos['Texto_Completo'])}

# Registra el tiempo de inicio
tiempo_inicio = time.time()

def palabras_elegidas(body):
    umbral_similitud = 80
    resultados = []

    # Divide el texto en palabras completas utilizando split()
    palabras = body.split()

    for palabra in palabras:
        mejores_coincidencias = []

        for item in diccionario:
            similitud = fuzz.ratio(palabra, item['palabra'])
            if similitud > umbral_similitud:
                mejores_coincidencias.append(item['palabra'])

        if mejores_coincidencias:
            resultados.extend(mejores_coincidencias)

    # Obtén el índice del comentario directamente desde el diccionario de mapeo
    posicion = comentario_a_indice[body]

    # No es necesario calcular el porcentaje completado en cada iteración

    return posicion, resultados

# Utiliza Parallel para aplicar palabras_elegidas en paralelo a tus datos
resultados_paralelos = Parallel(n_jobs=-1)(delayed(palabras_elegidas)(body) for body in conjunto_de_datos['Texto_Completo'])

# Registra el tiempo de finalización
tiempo_fin = time.time()

# Calcula el tiempo transcurrido en segundos
tiempo_transcurrido = tiempo_fin - tiempo_inicio

print("Tiempo transcurrido (segundos):", tiempo_transcurrido)

# Desempaqueta los resultados en índices y palabras
indices, palabras_encontradas = zip(*resultados_paralelos)

# Asigna los resultados de vuelta a tu DataFrame
conjunto_de_datos['Palabras_Elegidas'] = palabras_encontradas
```

```
/usr/local/lib/python3.10/dist-packages/joblib/externals/loky/process_executor.py:752: UserWarning: A worker stopped whi
  warnings.warn(
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-25-578ecdfefe34> in <cell line: 36>()
     34
     35 # Utiliza Parallel para aplicar palabras_elegidas en paralelo a tus datos
---> 36 resultados_paralelos = Parallel(n_jobs=-1)(delayed(palabras_elegidas)(body) for body in
conjunto_de_datos['Texto_Completo'])
     37
     38 # Registra el tiempo de finalización

                                 2 frames
/usr/local/lib/python3.10/dist-packages/joblib/parallel.py in _retrieve(self)
   1705                 (self._jobs[0].get_status(
   1706                     timeout=self.timeout) == TASK_PENDING)):
-> 1707                 time.sleep(0.01)
   1708                 continue
   1709

KeyboardInterrupt:
```

```
conjunto_de_datos
```