# TF-IDF

### Nombre: Jorge García

## Librerías

In [24]:
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity,euclidean_distances
import matplotlib.pyplot as plt
```

## Datos

In [7]:
```
df = pd.read_csv('movie_metadata.csv')
df.head()
```

Out[7]:

| | Id | movie_title | genero | plot_keywords | color | dire |
|---|---|---|---|---|---|---|
| **0** | 1 | Avatar | Action\|Adventure\|Fantasy\|Sci-Fi | avatar\|future\|marine\|native\|paraplegic | Color | |
| **1** | 2 | Pirates of the Caribbean: At World's End | Action\|Adventure\|Fantasy | goddess\|marriage ceremony\|marriage proposal\|pi... | Color | Gor |
| **2** | 3 | Spectre | Action\|Adventure\|Thriller | bomb\|espionage\|sequel\|spy\|terrorist | Color | Sa |
| **3** | 4 | The Dark Knight Rises | Action\|Thriller | deception\|imprisonment\|lawlessness\|police offi... | Color | C |
| **4** | 5 | Star Wars: Episode VII - The Force Awakens ... | Documentary | NaN | NaN | D |

5 rows × 29 columns

In [8]:
```
df['genero'] = df['genero'].str.replace('|',' ')
df['plot_keywords'] = df['plot_keywords'].str.replace('|',' ')
df.head()
```

Out[8]:

| | Id | movie_title | genero | plot_keywords | color | director_name | num_critic_for_reviews | dur |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Avatar | Action Adventure Fantasy Sci-Fi | avatar future marine native paraplegic | Color | James Cameron | 723.0 | |
| **1** | 2 | Pirates of the Caribbean: At World's End | Action Adventure Fantasy | goddess marriage ceremony marriage proposal pi... | Color | Gore Verbinski | 302.0 | |
| **2** | 3 | Spectre | Action Adventure Thriller | bomb espionage sequel spy terrorist | Color | Sam Mendes | 602.0 | |
| **3** | 4 | The Dark Knight Rises | Action Thriller | deception imprisonment lawlessness police offi... | Color | Christopher Nolan | 813.0 | |
| **4** | 5 | Star Wars: Episode VII - The Force Awakens ... | Documentary | NaN | NaN | Doug Walker | NaN | |

5 rows × 29 columns

In [9]:
```
df['texto'] = df[['genero','plot_keywords']].apply(lambda row: ' '.join(row.va
df.head()
```

Out[9]:

| | Id | movie_title | genero | plot_keywords | color | director_name | num_critic_for_reviews | dur |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Avatar | Action Adventure Fantasy Sci-Fi | avatar future marine native paraplegic | Color | James Cameron | 723.0 | |
| **1** | 2 | Pirates of the Caribbean: At World's End | Action Adventure Fantasy | goddess marriage ceremony marriage proposal pi... | Color | Gore Verbinski | 302.0 | |
| **2** | 3 | Spectre | Action Adventure Thriller | bomb espionage sequel spy terrorist | Color | Sam Mendes | 602.0 | |
| **3** | 4 | The Dark Knight Rises | Action Thriller | deception imprisonment lawlessness police offi... | Color | Christopher Nolan | 813.0 | |
| **4** | 5 | Star Wars: Episode VII - The Force Awakens ... | Documentary | NaN | NaN | Doug Walker | NaN | |

5 rows × 30 columns

# TF_IDF

In [12]:
```
tfidf = TfidfVectorizer(max_features=2000)
X = tfidf.fit_transform(df['texto'])
X
```

Out[12]:
```
<5043x2000 sparse matrix of type '<class 'numpy.float64'>'
        with 44000 stored elements in Compressed Sparse Row format>
```

In [14]:
```
peliculas = pd.Series(df.index, index=df['movie_title'])
peliculas.index = peliculas.index.str.strip()
peliculas
```

Out[14]:
```
movie_title
Avatar                                         0
Pirates of the Caribbean: At World's End       1
Spectre                                        2
The Dark Knight Rises                          3
Star Wars: Episode VII - The Force Awakens     4
                                             ...
Signed Sealed Delivered                     5038
The Following                               5039
A Plague So Pleasant                        5040
Shanghai Calling                            5041
My Date with Drew                           5042
Length: 5043, dtype: int64
```

In [16]:
```
indice =peliculas['The Following']
consulta =X[indice]
print(consulta.toarray())
```

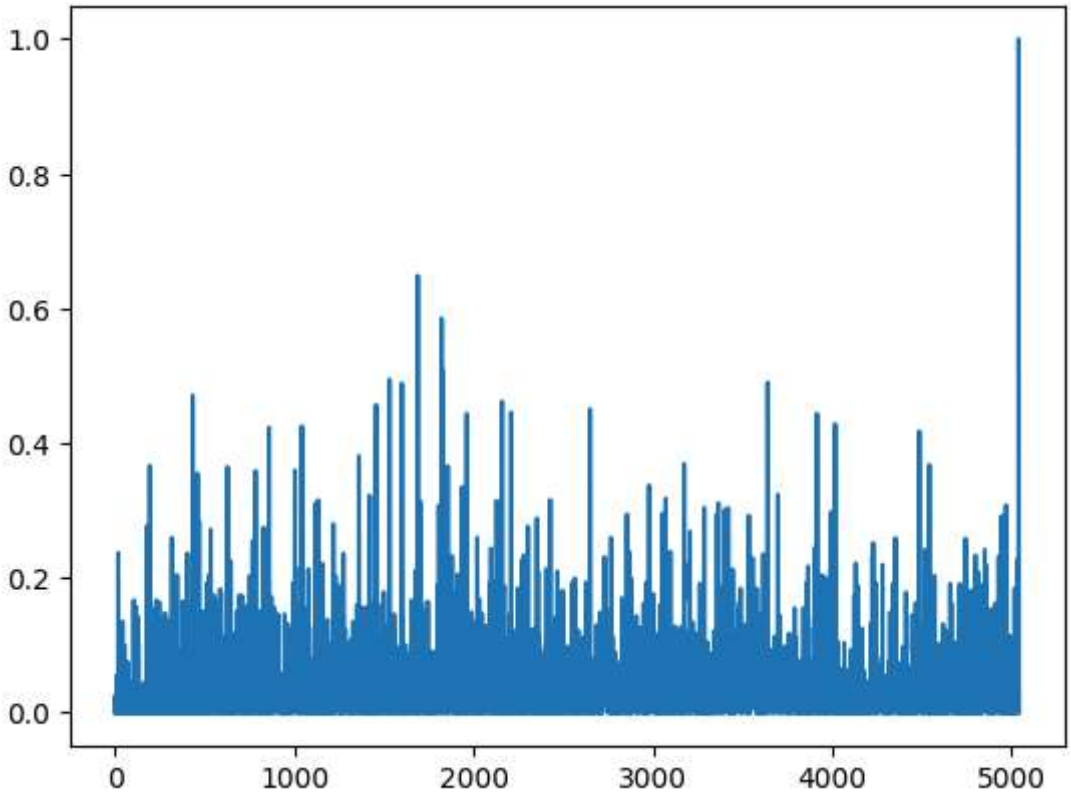
```
[[0. 0. 0. ... 0. 0. 0.]]
```

In [27]:
```
similitud =cosine_similarity(consulta,X)
similitud = similitud.flatten()
similitud
```

Out[27]:
```
array([0.        , 0.        , 0.0257117 , ..., 0.09187877, 0.0344376 ,
       0.        ])
```

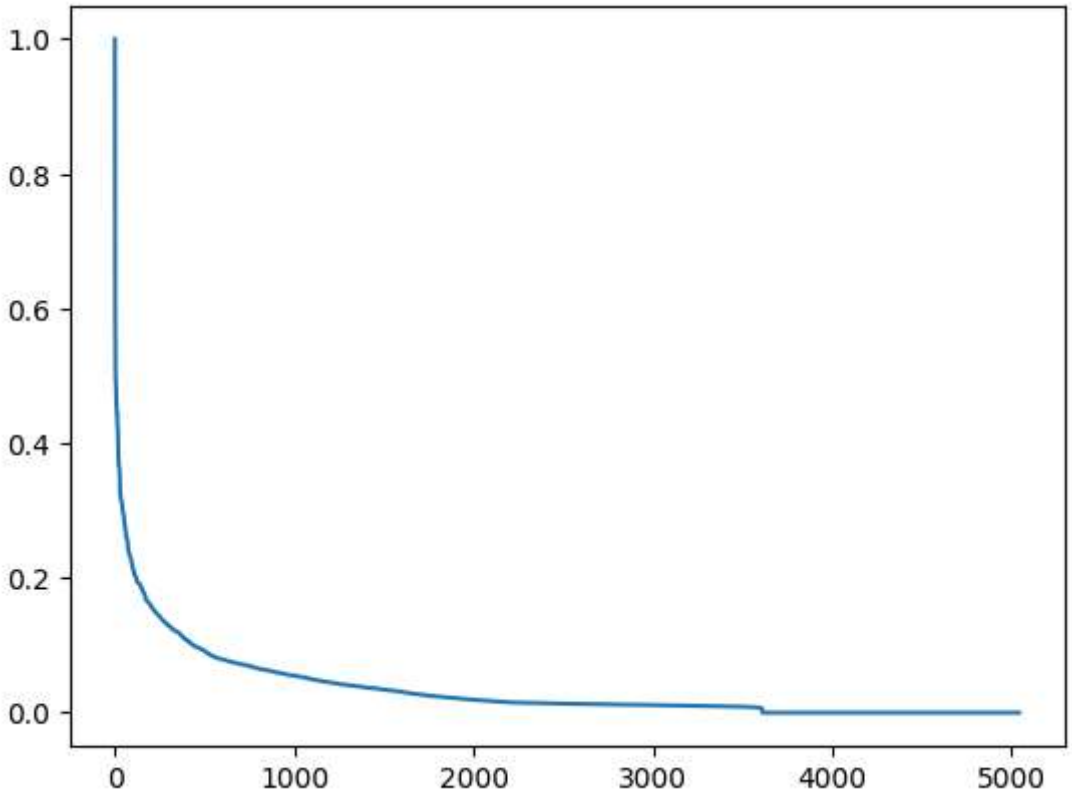
In [28]:
```
plt.plot(similitud)
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x1fa726ca410>]
```



In [30]:
```
similitud_ordenado = (-similitud).argsort()
plt.plot(similitud[similitud_ordenado])
```

Out[30]:
```
[<matplotlib.lines.Line2D at 0x1fa75bb2310>]
```




## Recomendación

In [31]:
```
recomendacion = similitud_ordenado[1:11]
df['movie_title'].iloc[recomendacion]
```

Out[31]:
```
1689                  88 Minutes
1822                Suspect Zero
1828                 Mindhunters
1531                 The Watcher
3640                 Lucky Break
1600                       Se7en
434                       Zodiac
2158    The Silence of the Lambs
1457                 Untraceable
2649     Halloween: Resurrection
Name: movie_title, dtype: object
```

In [ ]:
```

```