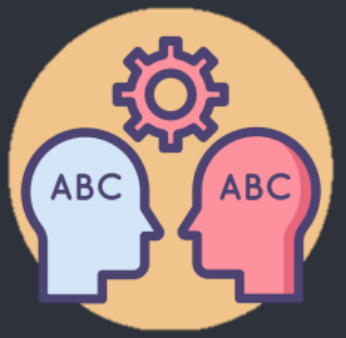
Procesamiento de Lenguaje Natural{



[NLP]

<Vectorizando Documentos| GloVe

}

Ivan Rojas Gonzalez – Universidad Panamericana

La Agenda de hoy {

01 Vectorizando documentos

<Document Embeddings>

02 Llevando esto a lo ya aprendido

<Conectando esto con el resto del curso>

03 Implementacion

<Codificando lo discutido>

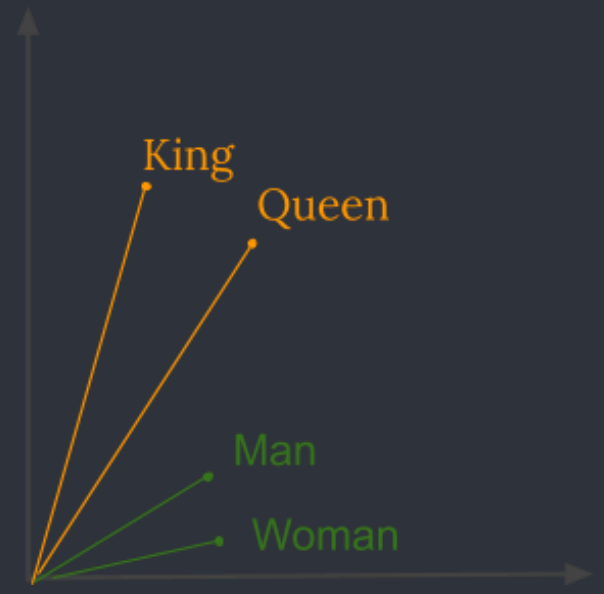
}

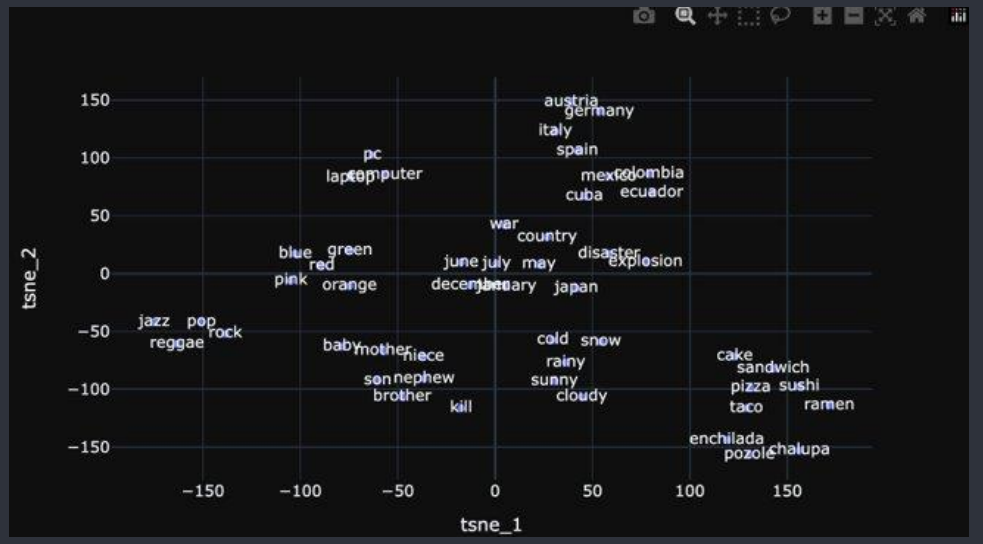
{Vectorizando Documentos}

Ivan Rojas Gonzalez – Universidad Panamericana

Vectorizando Documentos{

<Hasta ahora, nos hemos concentrado en obtener vectores de palabras aisladas a traves de sus Word Embeddings Pre-entrenados>





}

Vectorizando Documentos{

<Sin embargo, los archivos con los que hemos trabajado constan de oraciones, no de palabras aisladas>

Can You Match These Britney Spears Songs To Their Albums

At Least 14 Dead in Montana Crash

Funding for new Museum of Liverpool approved

}

Vectorizando Documentos{

<Debemos obtener un vector que represente cada frase>

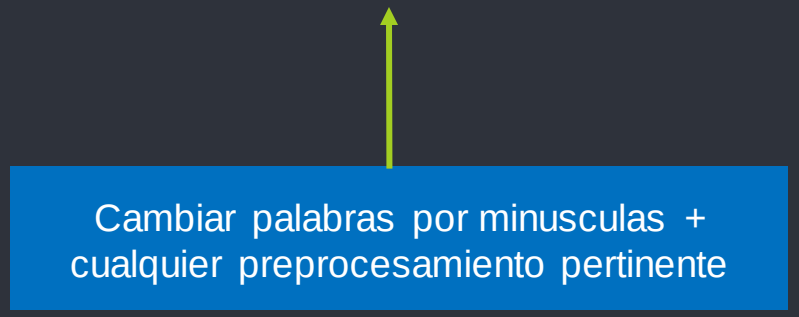
Funding for new Museum of Liverpool approved

}

Vectorizando Documentos{

<Debemos obtener un vector que represente cada frase>

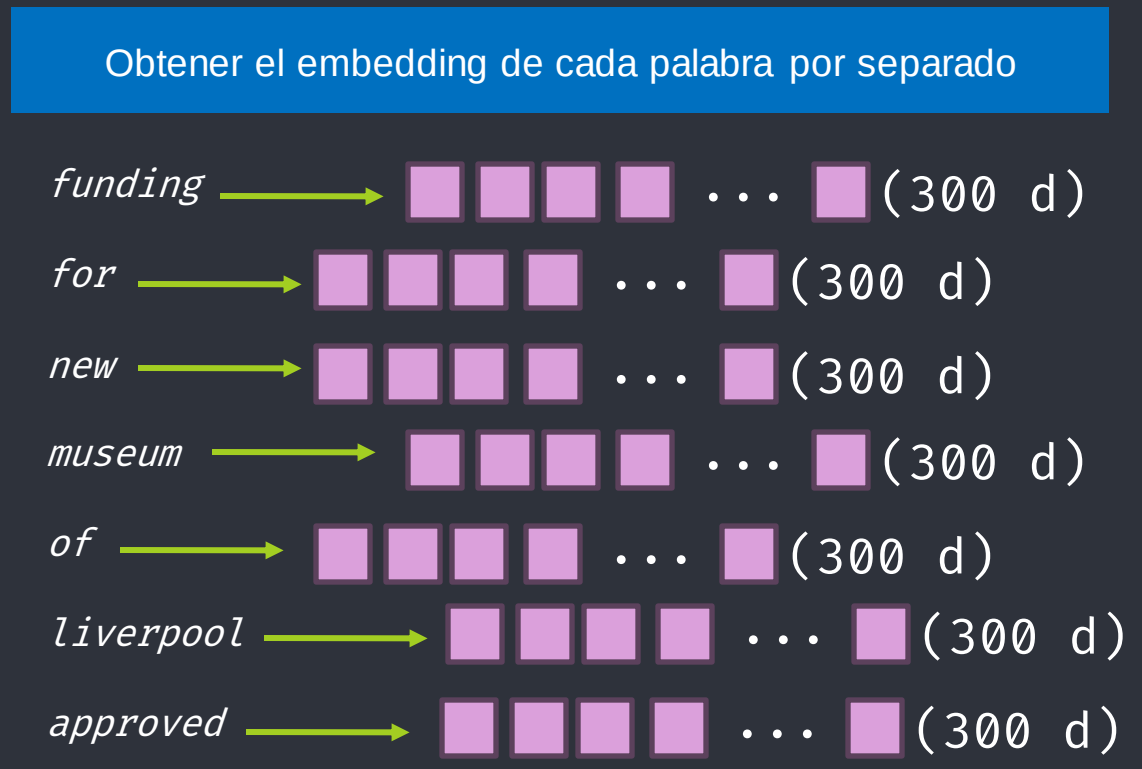
funding for new museum of liverpool approved



}

Vectorizando Documentos{

<Debemos obtener un vector que represente cada frase>

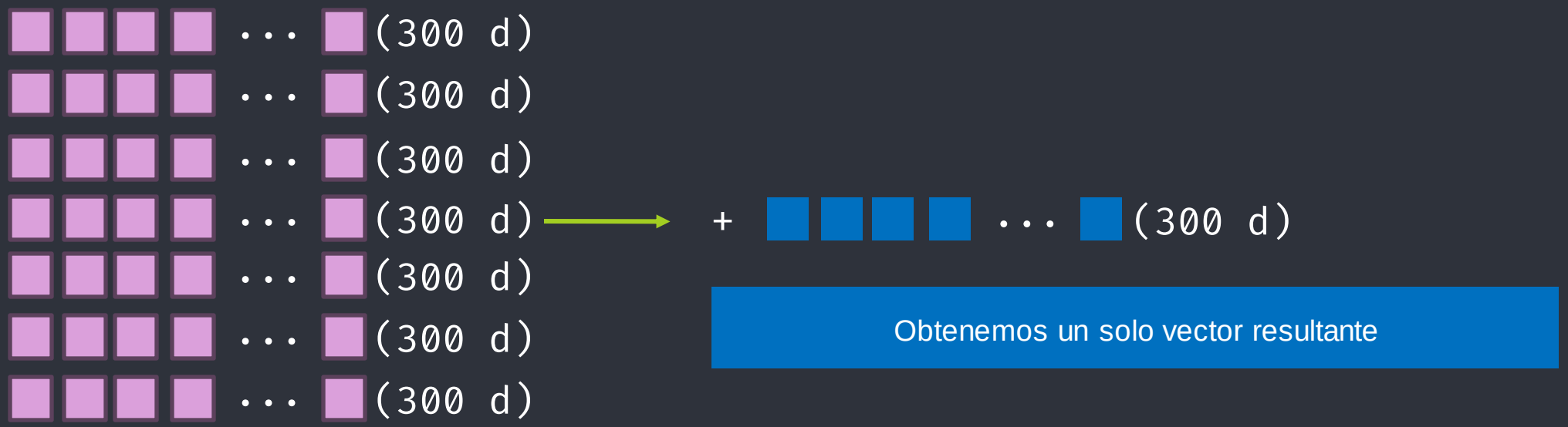


}

Vectorizando Documentos{

<Debemos obtener un vector que represente cada frase>

Sumar cada uno de los vectores obtenidos



Obtenemos un solo vector resultante

}

Vectorizando Documentos{

<Ese vector representara el embedding de la frase>

Podemos usar ese embedding para entrenar modelos de machine learning tradicional & deep learning

}

{Implementacion en Codigo}

Ivan Rojas Gonzalez – Universidad Panamericana

Implementando en Codigo{

```python
import re
import nltk
from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
stop_words_en = stopwords.words('English')

def vectorize(text):
    vector_size = 300
    texto = text.lower()
    texto = re.sub(r'([^0-9A-Za-z \t])','', texto)
    texto = word_tokenize(texto)
    texto = [palabra for palabra in texto if palabra not in stop_words_en]
    texto = [lemmatizer.lemmatize(palabra) for palabra in texto]

    vector =np.zeros(vector_size)
    for palabra in texto:
        if palabra in embeddings:
            vector = vector + embeddings[palabra].astype(float)
        else:
            print(f"No hay un embedding para la palabra {palabra}. Omitiendo...")
    vector = vector.reshape(1, -1)[0]
    return vector
```

}

El resto del codigo es un viejo conocido nuestro. Esta es la unica seccion nueva. En ella vectorizamos el documento dado como argumento

Implementando en Codigo{

```
df['vector'] = df['headline'].apply(vectorize)
df.head()
```

✓ 18.1s

```
No hay un embedding para la palabra theyve. Omitiendo...
No hay un embedding para la palabra latterday. Omitiendo...
No hay un embedding para la palabra hoverboards. Omitiendo...
No hay un embedding para la palabra antilandmine. Omitiendo...
No hay un embedding para la palabra delevingne. Omitiendo...
No hay un embedding para la palabra 200708. Omitiendo...
No hay un embedding para la palabra meetcute. Omitiendo...
No hay un embedding para la palabra itll. Omitiendo...
No hay un embedding para la palabra paddleboarded. Omitiendo...
No hay un embedding para la palabra prosharif. Omitiendo...
No hay un embedding para la palabra kengi. Omitiendo...
No hay un embedding para la palabra seventeenyearold. Omitiendo...
No hay un embedding para la palabra behindthescenes. Omitiendo...
```

Aplicamos funcion en columna de texto

}

Implementando en Codigo{

	headline	clickbait	vector
0	This Is What $1 USD Gets You In Food All Aroun...	clickbait	[-1.3219800000000002, 0.9059146, -0.8938480000...
1	Make These Easy Chicken Fajita Quesadillas At ...	clickbait	[1.4724720000000002, 0.785034, 0.4816820000000...
2	The Hardest "Walking Dead" Video Game Quiz You...	clickbait	[-1.0202828, 2.4568499999999998, -0.59897, -1....
3	34 Online Shops Based In The Southeast You Sho...	clickbait	[-1.403316, 0.7668303000000001, -0.26071499999...
4	US and France to work together for new Iran sa...	non-clickbait	[-0.728442, 0.36832800000000004, -0.8950400000...

Y con ello hemos obtenido los vectores correpondientes a cada headline o documento

}

{Aplicaciones}

Ivan Rojas Gonzalez – Universidad Panamericana

Aplicaciones{

<La columna con los vectores puede ser usada como argumento de modelos de machine learning>

Visualizacion con TSNE

```python
from sklearn.manifold import TSNE

X = df['vector']
X = np.concatenate(X, axis=0).reshape(-1,300)

model = TSNE(n_components=2)
resultado = model.fit_transform(X)

df['tsne_1'] = resultado[:,0]
df['tsne_2'] = resultado[:,1]

df.head()
```

✓ 1m 23.6s

	headline	clickbait	vector	tsne_1	tsne_2
0	This Is What $1 USD Gets You In Food All Aroun...	clickbait	[-1.3219800000000002, 0.9059146, -0.8938480000...	15.579122	41.143387
1	Make These Easy Chicken Fajita Quesadillas At ...	clickbait	[1.4724720000000002, 0.785034, 0.4816820000000...	-72.174026	28.518377
2	The Hardest "Walking Dead" Video Game Quiz You...	clickbait	[-1.0202828, 2.4568499999999998, -0.59897, -1....	1.646112	82.645714
3	34 Online Shops Based In The Southeast You Sho...	clickbait	[-1.403316, 0.7668303000000001, -0.26071499999...	22.005449	5.986506
4	US and France to work together for new Iran sa...	non-clickbait	[-0.728442, 0.36832800000000004, -0.8950400000...	55.428680	-34.965839

}

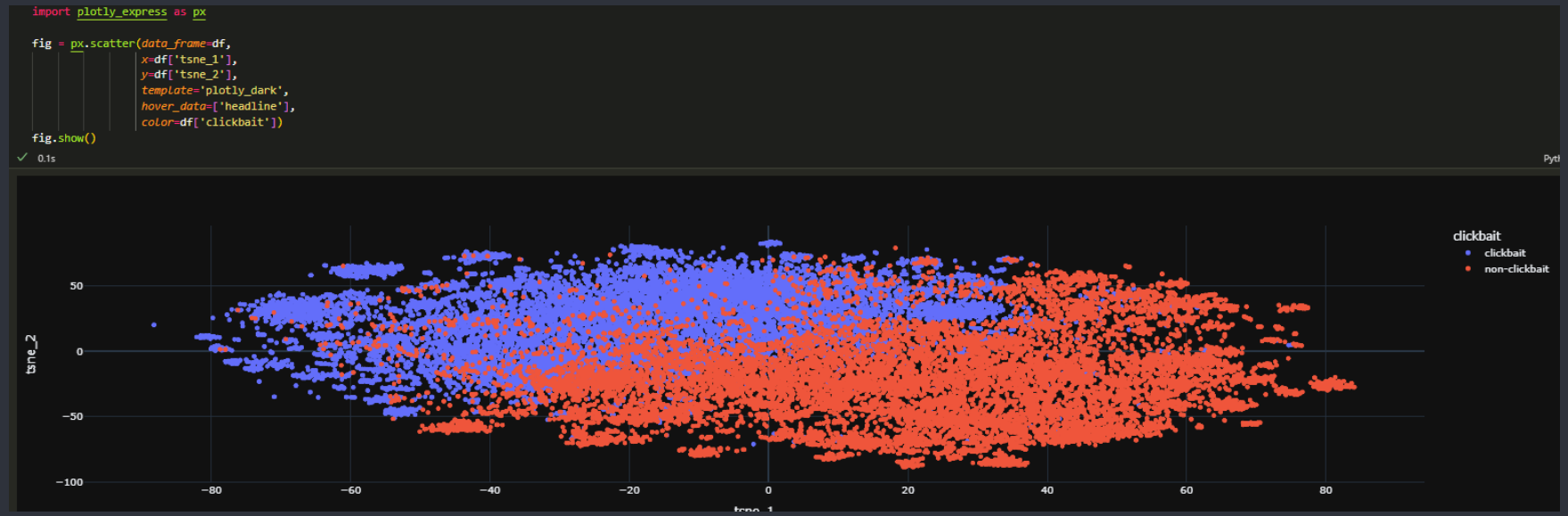
Aplicaciones{

<La columna con los vectores puede ser usada como argumento de modelos de machine learning>

Visualizacion con TSNE

```
import plotly_express as px

fig = px.scatter(data_frame=df,
                 x=df['tsne_1'],
                 y=df['tsne_2'],
                 template='plotly_dark',
                 hover_data=['headline'],
                 color=df['clickbait'])
fig.show()
```



}

Aplicaciones{

<La columna con los vectores puede ser usada como argumento de modelos de machine learning>

Clasificacion con Random Forests

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

X = df['vector']
y = df['clickbait']

X = np.concatenate(X, axis = 0).reshape(-1, 300)

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state = 101)
```

✓ 0.1s

```
model =  RandomForestClassifier()
model.fit(X_train,y_train)
```

✓ 46.9s

```
RandomForestClassifier
RandomForestClassifier()
```

```
y_pred = model.predict(X_test)
```

✓ 0.0s

}

Aplicaciones{

<La columna con los vectores puede ser usada como argumento de modelos de machine learning>

Clasificacion con Random Forests

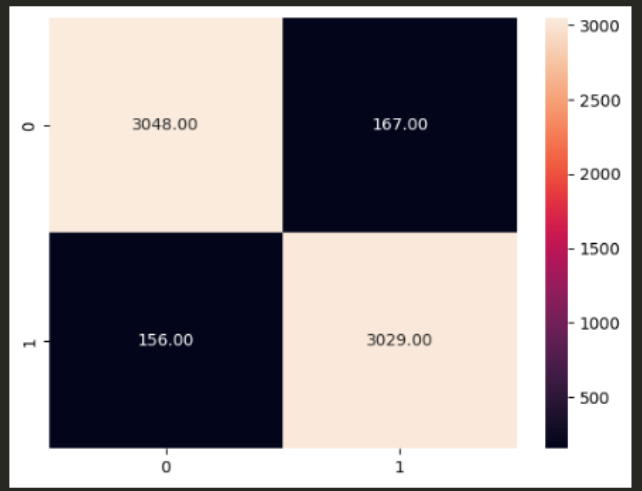
```
print(classification_report(y_test, y_pred)) #print and create the classification report
```

✓ 0.1s

	precision	recall	f1-score	support
clickbait	0.93	0.92	0.93	3215
non-clickbait	0.92	0.93	0.93	3185
accuracy			0.93	6400
macro avg	0.93	0.93	0.93	6400
weighted avg	0.93	0.93	0.93	6400

```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='.2f') #print the confusion matrix in a heat map
```

<AxesSubplot:>



}

Aplicaciones{

<La columna con los vectores puede ser usada como argumento de modelos de machine learning>

Clasificacion con Random Forests

```
def predecir(text):
    resultado = vectorize(text)
    prediccion = model.predict(resultado.reshape(-1, 300))
    return prediccion
```

✓ 0.0s

}

```
new_headline = "Your personality is based on your zodiac sign"
resultado = predecir(new_headline)
resultado
```

✓ 0.0s

```
array(['clickbait'], dtype=object)
```

```
new_headline = "Obama said that if you don't know your zodiac sign, you don't know about life"
resultado = predecir(new_headline)
resultado
```

✓ 0.0s

```
array(['clickbait'], dtype=object)
```

```
new_headline = "The president of Mexico worries about global warming"
resultado = predecir(new_headline)
resultado
```

✓ 0.0s

```
array(['non-clickbait'], dtype=object)
```

Aplicaciones{

<Incluso podemos utilizar similitud coseno y crear un sistema de recuperación de información>

```python
import numpy as np

def getTopXDocs(frase,x, export=False):
    data = {
        'headline':[],
        'sims':[]
    }
    buscar = vectorize(frase)
    for vector,headline in zip(df['vector'],df['headline']):
        A = buscar
        B = vector
        resultado = np.dot(A,B) / (np.linalg.norm(A)*np.linalg.norm(B))

        data['headline'].append(headline)
        data['sims'].append(resultado)
    final = pd.DataFrame(data).sort_values(by='sims',ascending=False).head(x)

    if export==True:
        final.to_csv('topDocs.csv', index=False)
    return final
```

}

```
getTopXDocs('war in the middle east escalates',20, export=True)
```

✓ 0.1s

C:\Users\ivani\AppData\Local\Temp\ipykernel_2684\2893026570.py:12: RuntimeWarning:

invalid value encountered in double_scalars

	headline	sims
13713	The Middle Kingdom Meets the Middle East	0.759033
12563	Iraq on verge of civil war, head of Arab leagu...	0.710702
16206	Pope Runs Into Politics of Middle East	0.710456
2896	Hezbollah-Israel conflict continues	0.705599
5638	Crisis or Not, Russia Will Build a Bridge in t...	0.681948
7111	Sri Lankan War Nears End, but Peace Remains Di...	0.675415
23287	North Korea warns of 'self-defensive blows,' n...	0.670252
8997	Kofi Annan: Iraq situation much worse than civ...	0.667778
25332	Middle Eastern troops enter Bahrain after prot...	0.667252
30861	Obama supports Middle East protesters in speech	0.667110

Aplicaciones{

<Asi funcionan las bases de datos de vectores utilizadas hoy en dia para modelos de lenguaje masivos! (ChatGPT et. Al.)>

```
getTopXDocs('war in the middle east escalates',20, export=True)
```

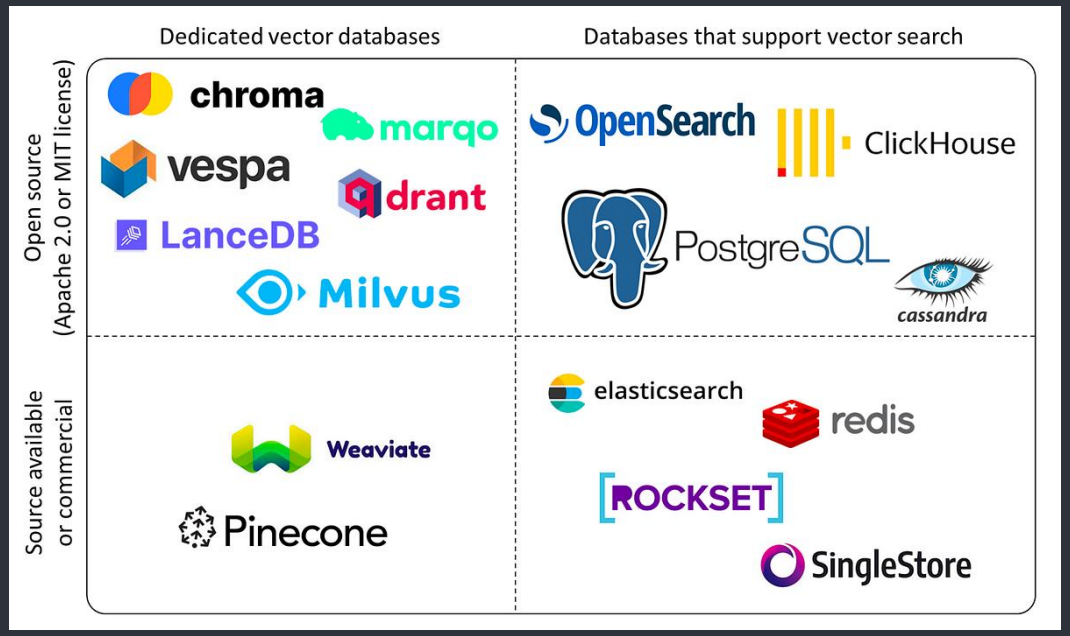
✓ 0.1s

C:\Users\ivani\AppData\Local\Temp\ipykernel_2684\2893026570.py:12: RuntimeWarning:

invalid value encountered in double_scalars

	headline	sims
13713	The Middle Kingdom Meets the Middle East	0.759033
12563	Iraq on verge of civil war, head of Arab leagu...	0.710702
16206	Pope Runs Into Politics of Middle East	0.710456
2896	Hezbollah-Israel conflict continues	0.705599
5638	Crisis or Not, Russia Will Build a Bridge in t...	0.681948
7111	Sri Lankan War Nears End, but Peace Remains Di...	0.675415
23287	North Korea warns of 'self-defensive blows,' n...	0.670252
8997	Kofi Annan: Iraq situation much worse than civ...	0.667778
25332	Middle Eastern troops enter Bahrain after prot...	0.667252
30861	Obama supports Middle East protesters in speech	0.667110

}



Q&A