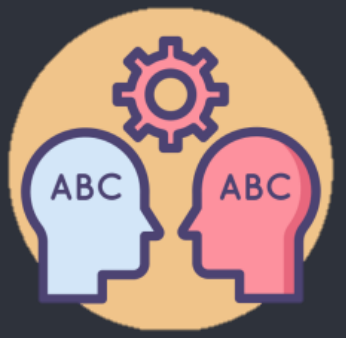
Procesamiento de Lenguaje Natural{



[NLP]

<Word Embeddings | GloVe

}

Ivan Rojas Gonzalez – Universidad Panamericana

La Agenda de hoy {

01 Que son los Word Embeddings

<Cómo funciona>

02 Que es GloVe

<Intuicion de GloVe>

03 Utilizando GloVe

<El poder de los embeddings pre-entrenados>

}

{Word Embeddings}

Ivan Rojas Gonzalez – Universidad Panamericana

Word Embeddings{

<Recordemos la tecnica de vectorizacion de texto que hemos usado mas frecuentemente>

TD-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

}

TF-IDF{

}

<Genera "sparce matrices". Si usamos bigramas, trigramas, n-gramas, la matriz crece>

<Los valores obtenidos no necesariamente representan una relación semántica entre las palabras>

<Varias arquitecturas de redes neuronales esperan que el numero de parametros a la entrada sea fijo. La matriz generada con TF-IDF es variable>

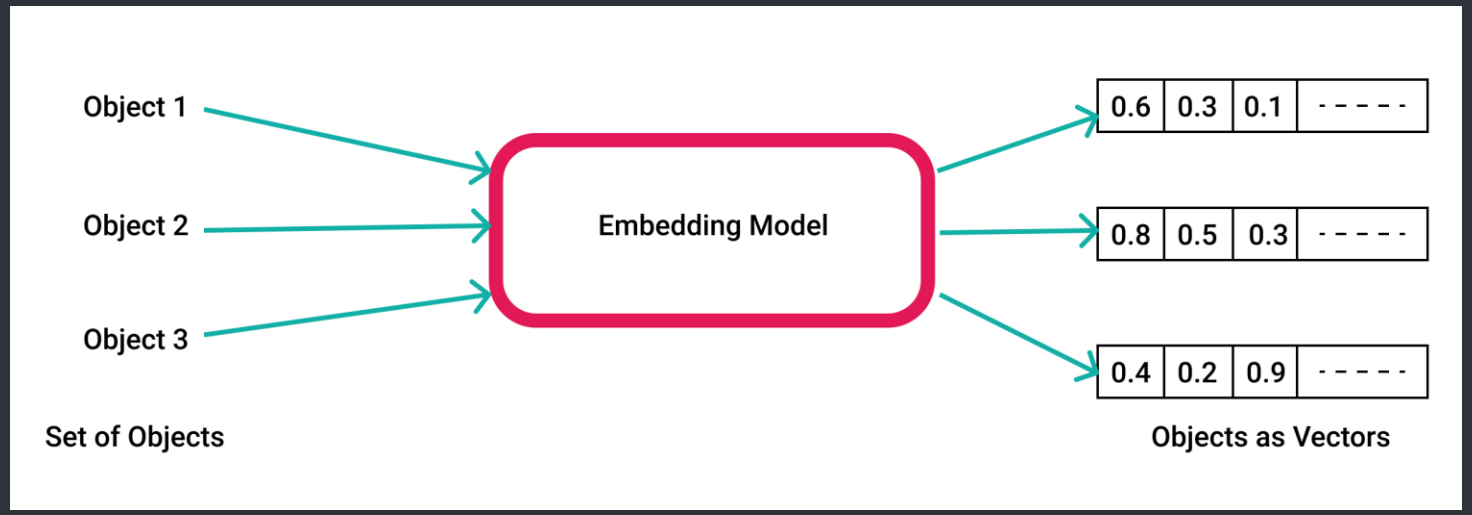
Word Embeddings{

<El termino "Embedding" refiere a una estructura contenida dentro de otra>

}

Word Embeddings{

<En "Word Embeddings" lo que se pretende es encapsular a la palabra y a su contexto semántico dentro de un vector>



}

Word Embeddings{

<Genera "vectores densos" de dimensiones fijas. I.e. ningun valor en el vector es 0 >

<Los valores obtenidos preservan la relacion semántica entre términos>

<Dado que el tamaño del vector obtenido es fijo, podemos usar esta informacion dentro de modelos de aprendizaje profundo>

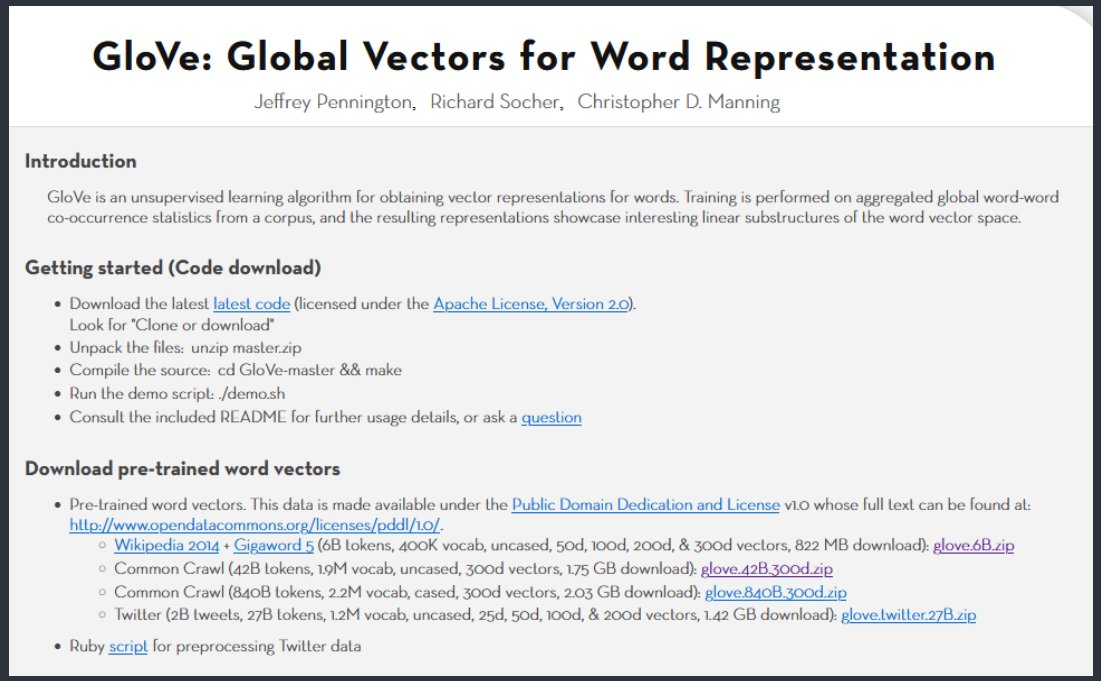
}

{GloVe}

Ivan Rojas Gonzalez – Universidad Panamericana

Global Vectors for Word Representation{

<GloVe es un algoritmo de aprendizaje no supervisado desarrollado en 2014 por miembros del departamento de NLP de la Universidad de Stanford>



GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning

Introduction

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Getting started (Code download)

- Download the latest latest code (licensed under the Apache License, Version 2.0). Look for "Clone or download"
- Unpack the files: unzip master.zip
- Compile the source: cd GloVe-master && make
- Run the demo script: ./demo.sh
- Consult the included README for further usage details, or ask a question

Download pre-trained word vectors

- Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: http://www.opendatacommons.org/licenses/pddl/1.0/.
  - Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
  - Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
  - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
  - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip
- Ruby script for preprocessing Twitter data

}

Global Vectors for Word Representation{

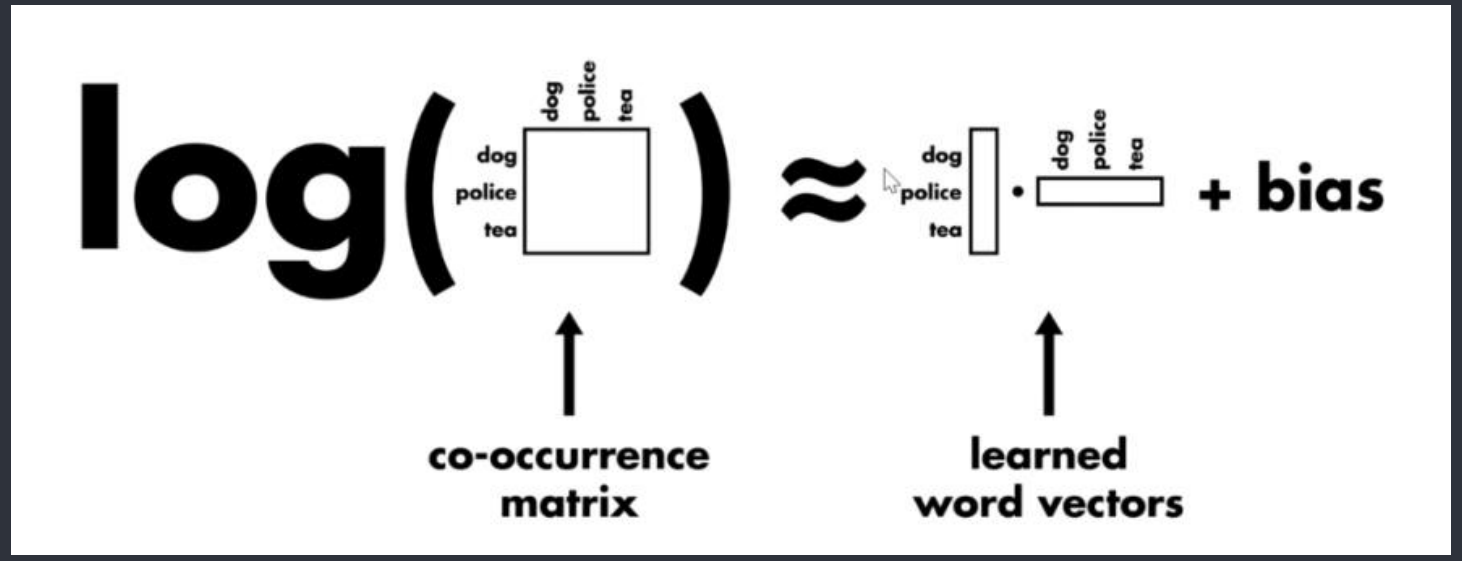
<GloVe parte del concepto de la matriz de co-ocurrencia. Para un corpus de vocabulario de dimension V, la mastriz sera una matriz de V X V donde la interseccion Xij denota el numero de veces que i ha co-ocurrido con j>

	the	cat	sat	on	mat
the	0	1	0	1	1
cat	1	0	1	0	0
sat	0	1	0	1	0
on	1	0	1	0	0
mat	1	0	0	0	0

}

Global Vectors for Word Representation{

<La idea es que el logaritmo del radio de probabilidades de cada termino en la matriz de co-ocurrencia es aproximadamente igual al producto punto de los vectores generados mas un "bias">



}

Global Vectors for Word Representation{

<Entrenar nuestros propios embeddings podria no ser muy practico dado que se requiere poder de procesamiento y un vocabulario extenso. Es por ello que procederemos a utilizar los vectores pre-entrenados de GloVe>

- Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
- Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip

}

{Utilizando GloVe}

Ivan Rojas Gonzalez – Universidad Panamericana

Utilizando GloVe{

<Debemos cargar los vectores a un diccionario de manera que podamos trabajar con ellos>

```python
#Numpy para que los vectores se cargen como numpy arrays
import numpy as np

#Diccionario vacio para guardar embeddings
embeddings ={}

#Abrimos el archivo de GloVe
with open('glove.6B.300d.txt') as f:
    for line in f: #Por cada par de palabra -> vector
        values = line.split() #Partimos por espacios
        word  = values[0] #Separamos la palabra del vector
        vectors = np.asarray(values[1:]) #Separamos el vector de la palabra
        embeddings[word] = vectors #Guardamos la palabra como un key en el diccionario cuyo value es el vector
```

}

Utilizando GloVe{

<Definimos una funcion para extraer el vector de alguna palabra provista>

```
def get_vector(word):
    return embeddings[word]
```

}

Utilizando GloVe{

<Para probar, podemos crear un dataset de campos semanticos>

```
vocab = ['computer','laptop','pc',
         'war','disaster','kill','explosion',
         'may','june','july','december','january',
         'rock','jazz','country','pop','reggae',
         'blue','red','green','orange','pink',
         'pizza','sandwich','taco','cake','sushi','ramen','chalupa','enchilada','pozole',
         'sunny','cloudy','snow','cold','rainy',
         'mexico','italy','germany','austria','japan','colombia','spain','cuba','ecuador',
         'brother','mother','son','baby','nephew','niece'
        ]

import pandas as pd
df = pd.DataFrame(vocab, columns=['word'])
df.head()
```

✓ 3.2s

	word
0	computer
1	laptop
2	pc
3	war
4	disaster

}

Utilizando GloVe{

<Obtenemos el vector de cada palabra>

```
df['vector'] = df['word'].apply(get_vector)
df.head()
```

	word	vector
0	computer	[-0.27628, 0.13999, 0.098519, -0.64019, 0.0319...
1	laptop	[-0.31974, 0.099409, -0.43893, -0.62199, -0.00...
2	pc	[-0.2184, 0.38967, -0.062816, -1.1064, -0.1782...
3	war	[-0.33042, 0.22503, 0.46759, -0.28312, 0.24944...
4	disaster	[0.031741, -0.022032, 0.27848, -0.24309, 0.286...

}

Utilizando GloVe{

<Y ahora, usamos la magia de nuestro viejo amigo TSNE>

```
from sklearn.manifold import TSNE

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

model = TSNE(n_components=2, perplexity=12)
resultado =model.fit_transform(X)

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

df.head()
```

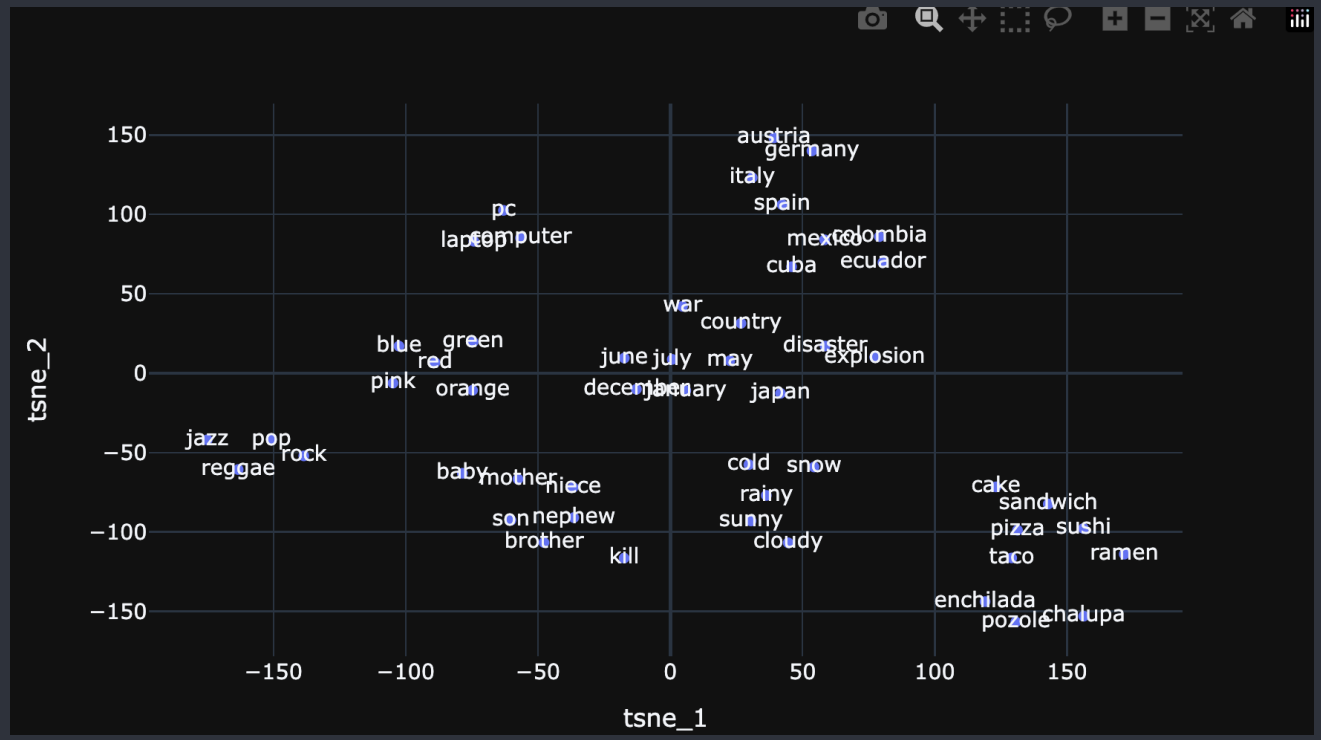
}

Utilizando GloVe{

<Al visualizar resultados, podemos constatar que los terminos semanticamente similares se encuentran agrupados>

```python
import plotly_express as px
fig = px.scatter(data_frame=df,
x=df['tsne_1'],
y=df['tsne_2'],
template='plotly_dark',
text=df['word']
)
fig.show()
```

}



Q&A