Análisis de Poemas con Glove y SOM

Redes Neuronales

Alumna:
Karla Adriana Esquivel Guzmán

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

15/Junio/2020

Introducción

En este proyecto se buscará la forma de clasificar poemas de diversos autores/poetas, a partir de la vectorización de cada uno de sus poemas, esto lo lograremos con ayuda de Glove, la representación vectorial de cada poema servirá para alimentar a SOM que al final nos dará como resultado una malla con los resultados de dicha clasificación.

Objetivo

¿Cómo vamos a vectorizar los poemas y obtener su clasificación?

Métodos

Principalmente se va a utilizar Glove,6b pero **¿Qué es Glove?** Glove es un modelo que utiliza un algoritmo de **aprendizaje no supervisado** que sirve para obtener la representación vectorial de palabras, esto se logra mapeando las palabras en un espacio dónde la distancia esta relacionada con respecto a su similitud semántica. El aprendizaje automatizado se basa en dar un conjunto de información sin etiquetas y a partir de ello una red neuronal extrae dicha información y a raíz de encontrar patrones etiqueta dicha información para posteriormente clasificarla.

¿Cómo se vectoriza un Poema?

El proceso de vectorización de un poema conlleva varios pasos que a continuación voy a enunciar:

1. Hay que tokenizar el poema, primero que nada debemos eliminar las palabras vacias o palabras sin significado (stop words), es decir palabras como artículos, peroposiciones, pronombres, etcétera, dichas palabras van a estar repetidas en muchos poemas, el dejarlas simplemente complicara más la diferenciación de los poemas. Como ejemplo tomaré el siguiente poema

The Appeal

It I have given you delight
By aught that I have done,
Let me lie quiet in that night
Which shall be yours anon:
And for the little, little, span
The dead are born in mind,
Seek not to question other than
The books I leave behind.

una vez eliminadas las palabras vacias obtendremos la siguiente lista:

['given', 'delight', 'aught', 'done', 'let', 'lie', 'quiet', 'night', 'shall', 'anon', 'little', 'little', 'span', 'dead', 'born', 'mind', 'seek', 'question', 'books', 'leave', 'behind']

2. Sin embargo como se puede notar aun tenemos palabras repetidas dentro de nuestra lista así que hay que eliminarlas, la lista final de palabras será la siguiente:

   ['anon', 'aught', 'behind', 'books', 'born', 'dead', 'delight','done', 'given', 'leave', 'let', 'lie', 'little','mind', 'night', 'question', 'quiet', 'seek', 'shall', 'span']

3. Una vez que obtenemos la lista de palabras vamos a obtener el vector asociado a cada una de ellas utilizando Glove, para ejemplificar utilizaré la palabra books, la siguiente es su representación vectorial:

```
(Pdb) glove['books']
array([-0.03345 ,  0.80877 , -0.20659 , -0.92842 ,  0.27433 ,  0.17549 ,
       -1.5501  , -1.9302  ,  0.42939 ,  0.15984 , -0.9263  ,  1.1073  ,
       -0.15518 , -0.3959  ,  1.431   , -0.74583 , -0.33683 , -0.21206 ,
        0.15386 , -0.022504,  1.127   ,  0.31919 ,  0.83898 ,  0.48734 ,
        0.2966  , -0.81942 , -1.5519  , -0.77488 , -0.28272 , -0.71643 ,
        2.6664  , -0.57047 ,  0.10581 , -0.10116 , -0.48662 ,  0.63422 ,
       -0.60514 ,  0.19788 , -0.25741 , -0.20731 ,  0.95538 , -0.10275 ,
        0.41266 ,  0.7293  , -0.29704 ,  0.60264 ,  0.29637 , -0.021069,
       -0.65104 , -0.58676 ])
```

4. Una vez obtenida la representación vectorial de cada una de las palabras, se debe obtener el promedio de los vectores, lo cual nos dará como resultado la representación vectorial del Poema, en este caso la representación del poema The Appeal es la siguiente:

```
Pdb) poemsVector[0]
rray([ 0.20987075,  0.22164867, -0.19477876, -0.26669845,  0.2442112 ,
       0.03558175, -0.03955173,  0.05628565, -0.13514657,  0.21897788,
      -0.11065322,  0.00523873,  0.02434767,  0.08715873,  0.38328473,
       0.21781614,  0.04998812,  0.03060938, -0.10353592, -0.4575028 ,
      -0.06806463,  0.24727543,  0.26713475, -0.1051461 ,  0.30859751,
      -0.66750971, -0.31886267,  0.43215798,  0.50420189,  0.00996084,
       1.84621696,  0.15536525, -0.01349627, -0.24265808, -0.12012784,
       0.14951236, -0.02783506,  0.16087743,  0.06481896, -0.31197598,
      -0.12879216, -0.03553098, -0.03691021,  0.16485148,  0.18672237,
       0.09209333,  0.11499473, -0.23732449, -0.11618696, -0.02560735])
```

Si bien ya he mencionado los pasos a seguir para vectorizar un poema ahora queda hablar sobre la etapa de experimentación.

Experimentación

Utilizando 60 poemas de 3 autores distintos, es decir en total 180 poemas, lo primero que hice fue vectorizar cada uno de estos poemas, posteriormente esta representación vectorial se la di como entrada a SOM, pero ahora surge otra pregunta **¿Qué es SOM?** SOM es un tipo de **Red Neuronal Autoasociativa** que utiliza

aprendizaje no supervisado, representa vectores de dimensión alta a dimensión baja, en mi caso utilicé MiniSOM `https://pypi.org/project/MiniSom/` MiniSOM es una implementación de SOM. Ahora que ya mencioné esto voy a agregar el siguiente fragmento de código:

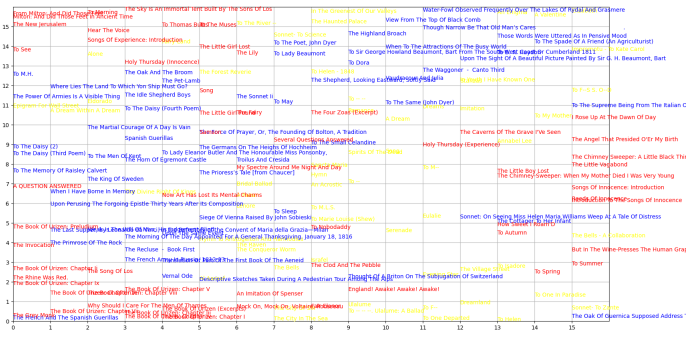
```
#Convertimos el arreglo en un arreglo de numpy
poemsVector = np.array(poemsVector)

#Definmos el tama;o del grid al que mapeare minisom
map_dim = 16
#Definimos el tama;o de entrada de los vectores
tamEntrada = 50
#Definimos la funcion de minisom
som = MiniSom(map_dim, map_dim, tamEntrada, sigma=1.0, random_seed=1)
#som.random_weights_init(poemsVector)
#Entrenamos la red neuronal con W que es el el arreglo de los revectores que
#representan los poemas
som.train_batch(poemsVector, num_iteration=len(poemsVector)*500, verbose=True)
```

En el código anterior simplemente estamos configurando el tamaño de la entrada que en este caso son vectores de Glove de tamaño 50, la malla de tamaño $16 * 16$ y la semilla con la que va a ejecutarse y para el entrenamiento se define el número de iteraciones.

Resultados

A continuación adjuntaré los resultados de diversos datos en los que me basé para la experimentación: De experimentar con el conjunto de autores de una época similar:
1809-1849 Edgar Allan Poe(Amarillo),
1770-1850 William Wordsworth(Azul),
1757-1827 William Blake(Rojo)

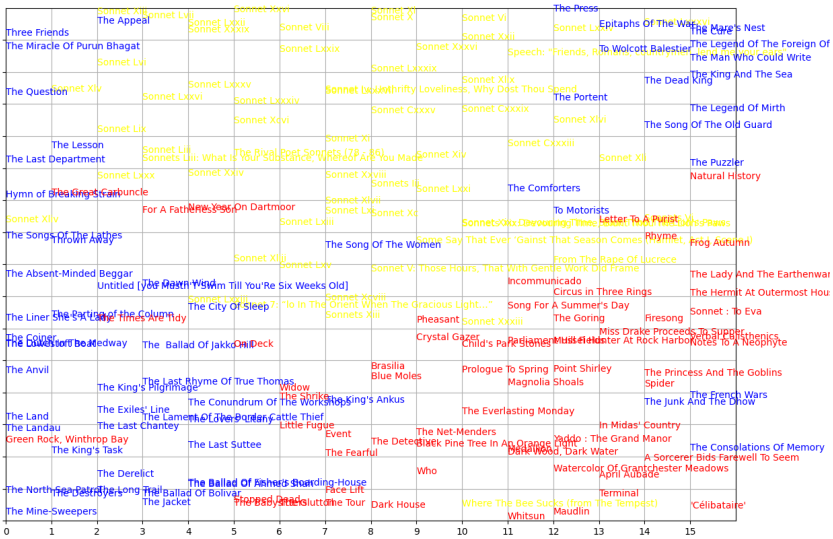


Conjunto de escritores de la misma nacionalidad en este caso Reino Unido
1608-1674 John Milton(Amarillo),

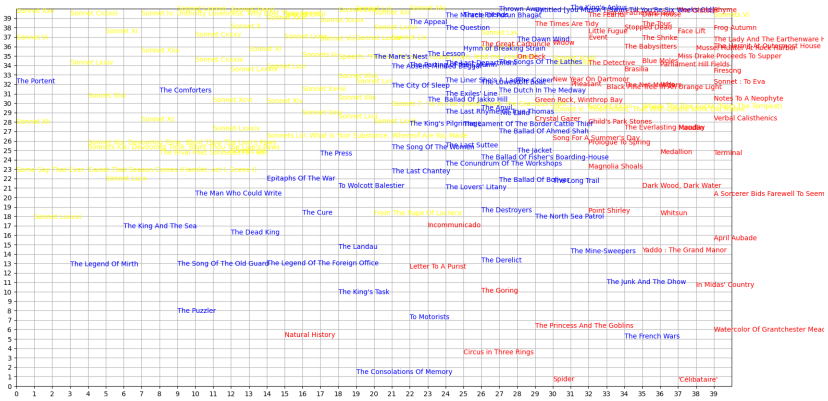
1770-1850 William Wordsworth(Azul),
1564-1616 William Shakespeare(Rojo),

Conjunto de escritores del mismo país USA
1819-1892 Walt Whitman(Amarillo),
1809-1849 Edgar Allan Poe(Rojo),
1885-1972 Ezra Pound(Azul)

Conjunto de escritores diferentes épocas y diferetes nacionalidades
1564-1616 William Shakespeare(Amarillo), Reino Unido,
1865-1936 Rudyard Kipling(Azul), India,
1932-1963 Sylvia Plath(Rojo), Estados Unidos
Este primer resultado se obtuvo utilizando un vector de tamaño 50 con una malla de $16 * 16$

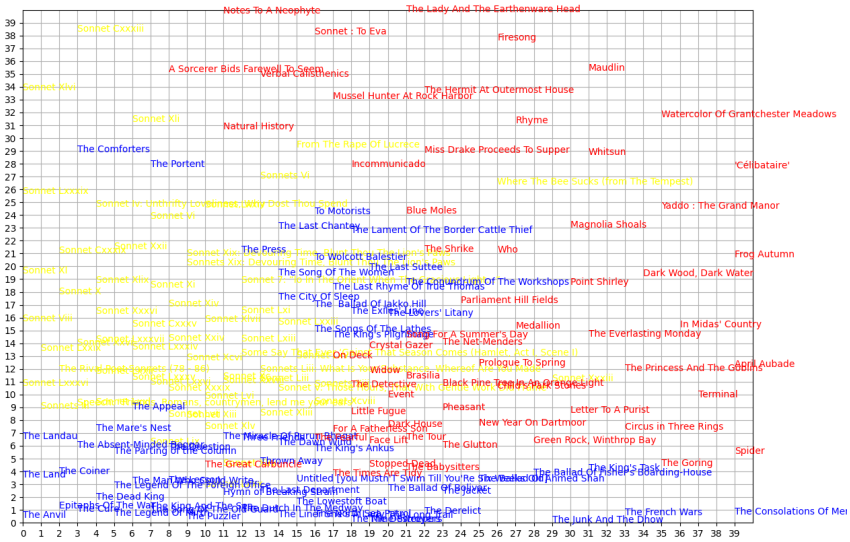


Tamaño del vector: 50 malla: $40 * 40$

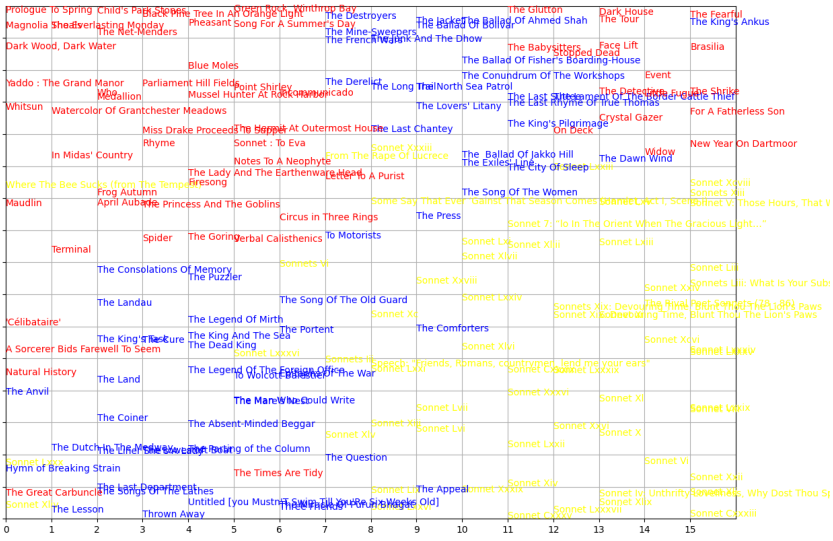


Tamaño del vector: 100 malla: $16 * 16$

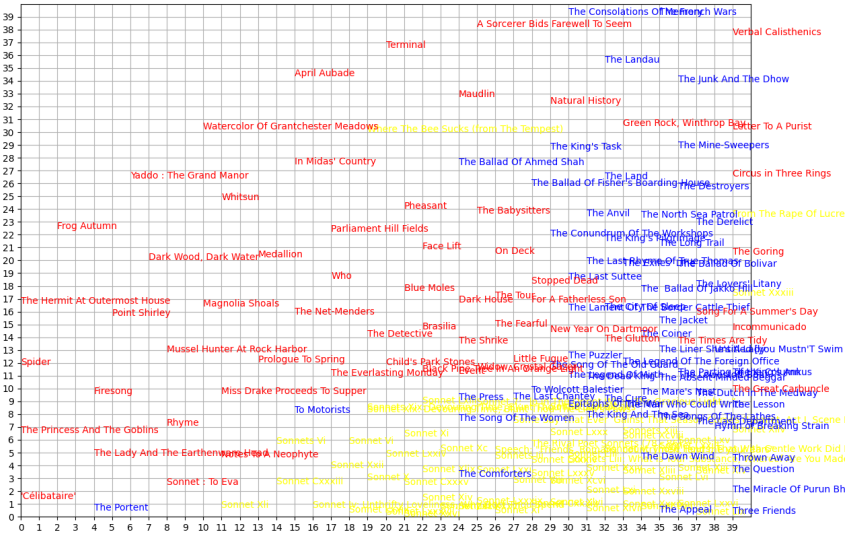
Tamaño del vector: 100 malla: $40 * 40$



Tamaño del vector: 300 malla: $16 * 16$



Tamaño del vector: 300 malla: $40 * 40$



Conclusiones

De los resultados de la experimentación pude encontrar datos muy interesantes incluído que tal parece que es más importante la nacionalidad a la cual hayan sido pertenecientes los Poetas que a la época en que vivieron, aunque claramente también la época es un factor importante, durante esa etapa del experimento no se obtuvieron tantos empates entre poetas como cuando todos son dela misma nacionalidad(Reino Unido), sin embargo en el experimento dónde cada uno de los poeta era de nacionalidad distinta podemos ver que la segmentación dentro de la malla es más clara.

Referencias

[1] `https://github.com/JustGlowing/minisom/blob/master/examples/PoemsAnalysis.ipynb`

[2] `https://www.poemhunter.com/`

[3] `https://pypi.org/project/MiniSom/#description`

[4] `https://en.wikipedia.org/wiki/GloVe_(machine_learning)`

[5] `https://en.wikipedia.org/wiki/Self-organizing_map`