

Rayniel Ramos González | Sistemas de Información | September 29, 2019

Shakespeare

# term\_document\_matrix:

La idea es guardar cuantas veces aparece una palabra en un documento. Para ello se recorren las palabras que hay en cada línea que se nos proporcionan, y se le aumenta en uno la celda palabra x documento. Esto servirá después para usando algún criterio de similaridad ver que tan similares son los documentos a partir de que tanto ocurren las mismas palabras en ellos, también sirve para ver qué tan similares son las palabras, palabras más similares tienden a ocurrir en documentos similares.

# term\_context\_matrix:

La idea de este ejercicio es crear una matriz que nos diga que tanta relación guardan las palabras entre sí, se asume que las palabras que poseen palabras iguales alrededor de ellas están más relacionadas que las que no, para establecer que tan cercanas deben estar las palabras se usa una ventana, cierto número de palabras a la izquierda y a la derecha de la palabra que se está analizando, las palabras que se encuentran en esta ventana son llamadas contexto, entonces aquellas palabras con contextos similares deben estar más relacionadas que las que no. Para darle solución al ejercicio se puede iterar por cada palabra de cada línea y para ella iterar por las palabras pertenecientes a la ventana (palabras contexto), aumentando en uno la celda palabra x contexto.

# PPMI\_matrix:

Esta matriz es similar a la tf-idf. Se halla el vector suma por columnas y el vector suma por fila, se halla la suma total y se procede a setear los valores de la matriz acorde a la fórmula del documento. Esta matriz es otra forma usada para darle peso a los términos.

# tf\_idf\_matrix:

Como la frecuencia de aparición de las palabras contexto alrededor del termino no da siempre una medida justa de la asociación, ya que es poco discriminativo, existen palabras con medidas similares pero que no guardan relación alguna, como son aquellas que son muy utilizadas y pueden aparecer frecuentemente en el contexto de otras, o sea, las palabras que son ubicuas no son importantes, por eso se usa este otro método algo más exacto. Primero de calcula la term\_frecuency usando el logaritmo como se establece en el libro, se crea la matriz, luego se calcula el df\_array, por cada termino, la cantidad de documentos en la que aparece (document frecuency), luego se calcula el idf\_array, la inverse document frecuency, atendiendo a la formula, luego se procede a hallar la tf-idf matriz con la multiplicación como se establece en el libro.

# cosine\_similarity:

Usado para dados dos vectores ver que tan similares son, pueden ser vectores que representen documentos o palabras. Para la implementación se halla el dot product de los vectores y se divide por la multiplicación de sus normas, considerando las normas como la raíz de la sumatoria de los cuadrados de sus componentes.

# jaccard\_similarity:

Otra forma de calcular la similaridad de dos vectores. Se crea el conjunto de índices (pueden ser documentos o términos en dependencia del vector) en donde la componente sea mayor que cero, y se procede a aplicar la fórmula.

# dice\_similarity:

Similar a la anterior, la fórmula es parecida y se hace el mismo procedimiento.

# rank\_plays:

Se calcula la similaridad del vector parámetro con los restantes vectores, se crea una lista con el resultado, posteriormente se crea otra lista con los índices de los vectores y se ordena esta última usando la lista anterior de la similaridad.

# rank\_words:

Igual al anterior.