



Maestría en Inteligencia Artificial Aplicada (MNA)

# Matrices DTM y Tf-idf

Procesamiento de Lenguaje Natural (NLP)

Luis Eduardo Falcón Morales

# Bolsa de palabras : Bag-of-Words (BOW)



En esta semana estudiaremos el tema conocido como “bolsa de palabras” o abreviado usualmente como “BOW” por sus siglas en inglés de “bag of words”.

Este concepto nos permitirá ir relacionando las principales palabras involucradas en un enunciado y que a su vez nos llevará a aplicarlo en problemas de análisis de sentimiento al combinarlo con modelos de aprendizaje automático (machine learning).

Dentro del área de NLP, la **Document-Term-Matrix** (DTM) de un documento es una matriz que muestra la frecuencia (conteo) de aparición de cada término (“token” o “palabra”) dentro de los documentos. Los términos y documentos se pueden acomodar en renglones o columnas de acuerdo al objetivo del análisis.

	tweet_1	tweet_2	tweet_3	...	tweet_n
Token_1	0	0	0	...	0
Token_2	1	2	0	...	0
Token_3	0	1	0	...	1
⋮	⋮	⋮	⋮	⋮	⋮
Token_m	0	0	3	...	0

**TDM :**  
**Term Document Matrix**

Token≈Term

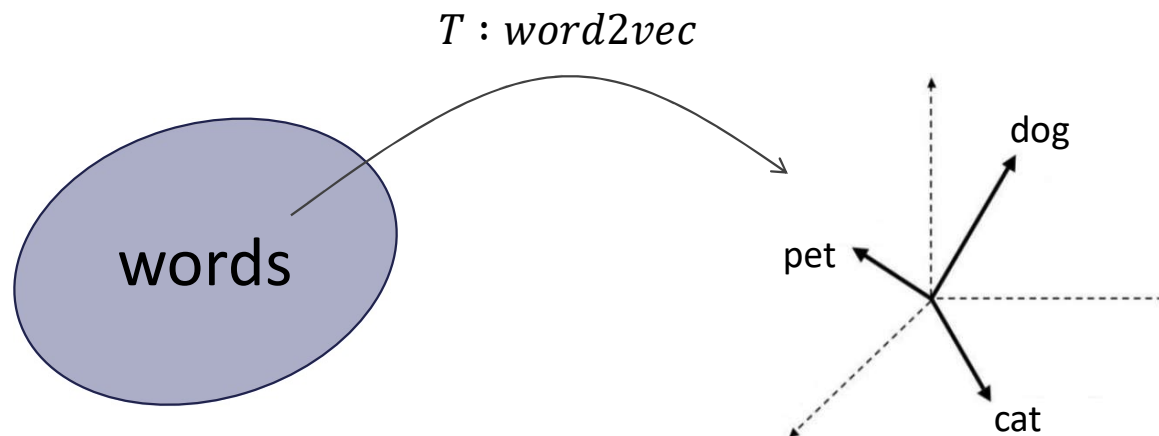
	Token_1	Token_2	Token_3	...	Token_n
tweet_1	0	1	0	...	0
tweet_2	0	2	1	...	0
tweet_3	0	0	0	...	3
⋮	⋮	⋮	⋮	⋮	⋮
tweet_m	0	0	1	...	0

**DTM :**  
**Document Term Matrix**

## Modelado/Vectorización del Lenguaje

Requerimos transformar los documentos de texto en información numérica para poder aplicar los conocimientos de matemáticas.

Al proceso de transformar texto en números se le conoce como **Vectorización** o **Técnicas de Embebido: word2vec**.






## Bolsa de Palabras / Bag of Words (BoW)

El método BOW es uno de los métodos más comunes y sencillos para la transformación numérica de un texto.

El método consiste en asignar a cada palabra su frecuencia de aparición en los documentos: se puede tratar de la frecuencia natural en cada documento, o de la frecuencia relativa, o simplemente un valor binario indicando si dicho término aparece o no en dicho documento.

Sin embargo, este método no funciona muy bien cuando se tienen pocos documentos y texto.






Doc\_1: Mario quiere jugar. Laura también quiere jugar.

Doc\_2: Laura también quiere estudiar.

De aquí podemos obtener un vocabulario para cada documento, indicando su frecuencia de aparición o conteo en cada uno. Ignorando los signos de puntuación:

Doc1 : { “Mario”: 1, “quiere”: 2, “jugar”: 2, “Laura”: 1, “también”: 1 }.

Doc\_2: { “Laura”: 1, “también”: 1, “quiere”: 1, “estudiar”: 1 }.



Doc1 : { “Mario”: 1, “quiere”: 2, “jugar”: 2, “Laura”: 1, “también”: 1 }.

Doc\_2: { “Laura”: 1, “también”: 1, “quiere”: 1, “estudiar”: 1 }.

La representación matricial en su forma Term-Document-Matrix (TDM) sería:

Vocabulario	Doc_1	Doc_2
Mario	1	0
Laura	1	1
quiere	2	1
jugar	2	0
también	1	1
estudiar	0	1

Observa que no se considera el orden de las palabras en BoW.

Vocabulario	Doc_1	Doc_2
Mario	1	0
Laura	1	1
quiere	2	1
jugar	2	0
también	1	1
estudiar	0	1

**Bag of Words vectors:** de dicha matriz ya podemos obtener representaciones vectoriales de cada término o documento:

Vectores 2-dim {  
 Vectorización de “Mario”: [1, 0]  
 Vectorización de “Laura”: [1, 1]  
 Vectorización de “también”: [1, 1] ... etc.

---

Vectores 6-dim {  
 Vectorización del Documento 1: [1, 1, 2, 2, 1, 0]  
 Vectorización del Documento 2: [0, 1, 1, 0, 1, 1]



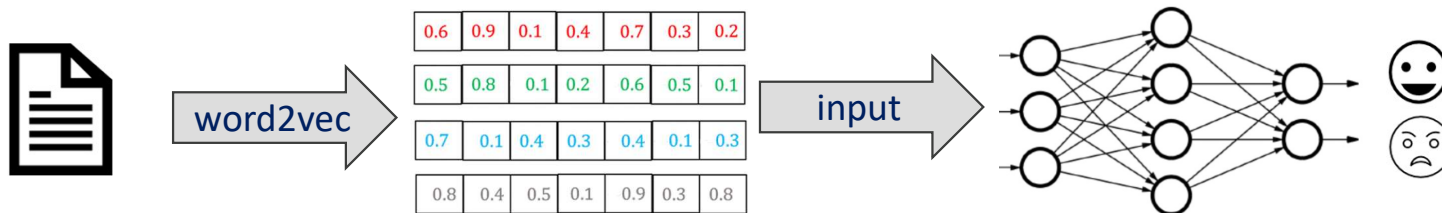
Vectores 2-dim

Vectorización de “Mario”: [1, 0]  
Vectorización de “Laura”: [1, 1]  
Vectorización de “también”: [1, 1] ... etc.

Vectores 6-dim

Vectorización del Documento 1: [1, 1, 2, 2, 1, 0]  
Vectorización del Documento 2: [0, 1, 1, 0, 1, 1]

Una vez que tenemos los documentos vectorizados (más adelante veremos otros casos), estos se podrán utilizar para por ejemplo usarlos como datos de entrada en una red neuronal.





# Matrices DTM, TDM, TF-IDF



En esta semana estudiaremos las diferentes formas de representar mediante una matriz, la información extraída de un document.

Existen diversas formas de llevar a cabo dicha representación, veremos las ventajas y desventajas de cada una de ellas.

Dentro del área de NLP, la **Document-Term-Matrix** (DTM) de un documento es una matriz que muestra la frecuencia (conteo) de aparición de cada término (token o “palabra”) dentro de los documentos. Los términos y documentos se pueden acomodar en renglones o columnas de acuerdo al objetivo del análisis.

	tweet_1	tweet_2	tweet_3	...	tweet_n
Token_1	0	0	0	...	0
Token_2	1	2	0	...	0
Token_3	0	1	0	...	1
⋮	⋮	⋮	⋮	⋮	⋮
Token_m	0	0	3	...	0

**TDM :**  
**Term Document Matrix**

	Token_1	Token_2	Token_3	...	Token_n
tweet_1	0	1	0	...	0
tweet_2	0	2	1	...	0
tweet_3	0	0	0	...	3
⋮	⋮	⋮	⋮	⋮	⋮
tweet_m	0	0	1	...	0

**DTM :**  
**Document Term Matrix**

Token  $\approx$  Term  $\approx$  Word  
Tweet  $\approx$  Document




Doc\_1: Mario quiere jugar. Laura también quiere jugar.

Doc\_2: Laura también quiere estudiar.

De aquí podemos obtener un vocabulario para cada documento, indicando su frecuencia de aparición o conteo en cada uno. Ignorando los signos de puntuación:

Doc1 : { “Mario”: 1, “quiere”: 2, “jugar”: 2, “Laura”: 1, “también”: 1 }.

Doc\_2: { “Laura”: 1, “también”: 1, “quiere”: 1, “estudiar”: 1 }.



### Term-Document-Matrix (TDM) en su opción binaria:

En este caso solo se considera si cada palabra aparece en dicho documento sin importar su frecuencia.

---

En el caso del ejemplo anterior la matriz binaria queda como sigue:

Vocabulario	Doc_1	Doc_2
Mario	1	0
Laura	1	1
quiere	1	1
jugar	1	0
también	1	1
estudiar	0	1

Doc\_1: Mario quiere jugar. Laura también quiere jugar.

Doc\_2: Laura también quiere estudiar.



## Diccionario & Tokenización & Document-Term Matrix

Now that we have all the messages with the desired words, we must divide each message into its individual components.

This process is called **Tokenization**.

In our example, a single token is a word, but in other situations, it could be a letter, a sentence, a paragraph, etc.

This process will generate a dictionary of words or bag of words.





tf : term frequency

La **frecuencia de término** (term frequency), es simplemente la frecuencia absoluta, es decir, el número de veces que una palabra, o término,  $t$  aparece en un documento  $d$ , y lo denotamos  $tf_{(t,d)}$ .

En ocasiones abusando de la notación, la frecuencia de término la denotamos simplemente como  $tf$ .

Esta métrica, aunque muy sencilla, nos proporciona una buena primera aproximación para empezar a medir la importancia de las palabras en un documento.

---

#### Notación

En las siguientes diapositivas:

Cada documento lo denotaremos como  $d$ .

Al conjunto de todos los documentos lo denotaremos como  $D$ .



idf : inverse document frequency – frecuencia inversa de documento

La **inverse document frequency** de una palabra  $t$  es una medida sobre qué tan importante es dicha palabra en el conjunto de todos los documentos  $D$ .

La denotamos:  $idf_{(t,D)}$  y se define usualmente como:

$$idf_{(t,D)} = \log \left\{ \frac{|D|}{n_t} \right\}$$

donde:

$|D|$  : es el total de documentos.

$n_t$  : es el número de documentos donde aparece  $t$ .

Mientras más aparece una palabra en la mayoría de los documentos,  $idf \approx 0$ . Es decir, filtra las palabras comunes.

---

En ocasiones se usa  $\log\{|D|/(1 + n_t)\}$  para evitar división por cero, en caso de que la palabra no aparezca en ningún documento.

La base del logaritmo utilizada es indiferente. En particular, como predeterminado R utiliza base 2 y Python base  $e$ .

## tf-idf : term frequency – inverse document frequency

Finalmente definimos la métrica **term frequency – inverse document frequency** como el producto de ambas métricas:

$$\begin{aligned} tf-idf_{(t,d,D)} &= tf_{(t,d)} \times idf_{(t,D)} \\ &= tf_{(t,d)} \times \log \left\{ \frac{|D|}{n_t} \right\} \end{aligned}$$

De esta expresión heurística, observamos que los valores mayores de tf-idf se alcanzan cuando tenemos una frecuencia alta tf de una palabra  $t$  en un documento  $d$ , pero al mismo tiempo que dicha palabra no aparece en la mayoría de los documentos de  $D$ .

Es decir, se están filtrando las palabras comunes que no proporcionan información relevante a un documento.

## Normalización de la frecuencia de término tf

En ocasiones conviene usar una frecuencia relativa, en lugar de la frecuencia absoluta.

En particular se puede utilizar la frecuencia relativa con respecto al total de palabras/términos de cada documento:  $\sum_k tf_{(k,d)}$ .

Así, la frecuencia de término normalizada, denotada  $tf_{|t,d|}$ , sería:

$$tf_{|t,d|} = \frac{tf_{(t,d)}}{\sum_k tf_{(k,d)}}$$

Y por lo tanto la tf-idf quedaría como:

$$tf-idf_{|t,d,D|} = tf_{|t,d|} \times idf_{(t,D)}$$

### Ejemplo:

Veamos el siguiente texto muy sencillo en español, pero que nos ayude a ilustrar las métricas recién definidas:

```
> content  
[1] "Pitagoras es el padre de las Matematicas."  
[2] "Las Matematicas rigen la belleza de la naturaleza."  
[3] "Hay belleza en la naturaleza y belleza en las Matematicas."  
>
```

Consta de 3 enunciados o documentos.

Se están omitiendo los acentos.

Supongamos que se hace un limpiado de los documentos quitando signos de puntuación y stopwords para obtener:

```
[1] pitagoras padre matematicas  
[2] matematicas rigen belleza naturaleza  
[3] belleza naturaleza belleza matematicas
```

```
[1] pitagoras padre matematicas
[2] matematicas rigen belleza naturaleza
[3] belleza naturaleza belleza matematicas
```

DTM para el caso binario:

	Terms					
Docs	matematicas	padre	pitagoras	belleza	naturaleza	rigen
1	1	1	1	0	0	0
2	1	0	0	1	1	1
3	1	0	0	1	1	0

DTM para el caso de frecuencias absolutas (o conteo) tf

	Terms					
Docs	matematicas	padre	pitagoras	belleza	naturaleza	rigen
1	1	1	1	0	0	0
2	1	0	0	1	1	1
3	1	0	0	2	1	0

A partir de la matriz de las frecuencias de término tf o Document Term Matrix, obtenemos las frecuencias inversas de documento idf con la fórmula:

$$idf_{(t,D)} = \log \left\{ \frac{|D|}{n_t} \right\}$$

Y obtenemos, usando logaritmo base 2:

	matematicas	padre	pitagoras	belleza	naturaleza	rigen
idf =	0	1.58496	1.58496	0.58496	0.58496	1.58496

Observa que las idf son por cada palabra en relación a todos los documentos.

Y finalmente se obtiene la tabla con las tf-idf por cada palabra y cada documento para el caso **no normalizado**:

Doc:	matematicas	padre	pitagoras	belleza	naturaleza	rigen
[1]	0	1.58496	1.58496	0	0	0
[2]	0	0	0	0.58496	0.58496	1.58496
[3]	0	0	0	1.16993	0.58496	0

Así, en este caso muy simple: en el primer documento “padre” y “pitagoras” son las palabras más relevantes. En el documento dos es “rigen” y en el tercer documento “belleza”.

Observa que la palabra Matemáticas quedó filtrada por aparecer en todos los documentos.

Para el **caso normalizado**, simplemente tendríamos que calcular primero los totales de palabras por documento y calcular las frecuencias relativas con respecto a dichos subtotales. Es decir, a partir de:

Doc:	matematicas	padre	pitagoras	belleza	naturaleza	rigen	Totales
[1]	1	1	1	0	0	0	3
[2]	1	0	0	1	1	1	4
[3]	1	0	0	2	1	0	4

Obtenemos la tabla de términos de frecuencia normalizadas:

Doc:	matematicas	padre	pitagoras	belleza	naturaleza	rigen
[1]	0.3333	0.3333	0.3333	0	0	0
[2]	0.25	0	0	0.25	0.25	0.25
[3]	0.25	0	0	0.5	0.25	0

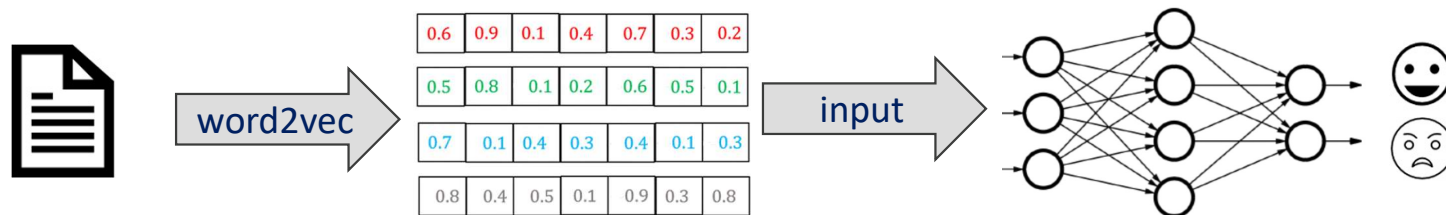
Y finalmente, con los mismos valores idf calculados previamente, obtenemos la tabla de los valores tf-idf normalizados:

Doc:	matematicas	padre	pitagoras	belleza	naturaleza	rigen
[1]	0	0.5283	0.5283	0	0	0
[2]	0	0	0	0.1462	0.1462	0.3962
[3]	0	0	0	0.2925	0.1462	0

Vectores 2-dim {  
Vectorización de “Mario”: [1, 0]  
Vectorización de “Laura”: [1, 1]  
Vectorización de “también”: [1, 1] ... etc.

Vectores 6-dim {  
Vectorización del Documento 1: [1, 1, 2, 2, 1, 0]  
Vectorización del Documento 2: [0, 1, 1, 0, 1, 1]

Una vez que tenemos los documentos vectorizados (más adelante veremos otros casos), estos se podrán utilizar para por ejemplo usarlos como datos de entrada en una red neuronal.







D.R.© Tecnológico de Monterrey, México, 2022.  
Prohibida la reproducción total o parcial  
de esta obra sin expresa autorización del  
Tecnológico de Monterrey.