Minería de Datos
IIC2433

Análisis de Texto

Vicente Domínguez

¿Se pueden aplicar los métodos vistos a documentos de texto?

Los conjuntos de datos que hemos estudiado se ven así:

	Atrib. 1	**Atrib.2**	**Atrib. 3**	**Atrib. 4**	**Atrib. 5**	**Atrib. 6**	**Atrib. 7**	**Atrib. 8**
Obj. 1	0	0	1	2	7	0	1	0
Obj. 2	1	1	5	8	0	0	1	0
Obj. 3	1	1	0	0	5	9	3	1
Obj. 4	3	7	3	6	3	8	2	2

Los conjuntos de datos que hemos estudiado se ven así:

	Atrib. 1	**Atrib.2**	**Atrib. 3**	**Atrib. 4**	**Atrib. 5**	**Atrib. 6**	**Atrib. 7**	**Atrib. 8**
Obj. 1	0	0	1	2	7	0	1	0
Obj. 2	1	1	5	8	0	0	1	0
Obj. 3	1	1	0	0	5	9	3	1
Obj. 4	3	7	3	6	3	8	2	2

Pero los documentos se ven así:

A pesar del fallido intento de la candidata presidencial de la DC, Carolina Goic, por cerrar la disputa con gesto al oficialismo por su respaldo unitario a favor del proyecto de elección de gobernadores regionales, esta tarde su coordinador político de campaña, Jorge Burgos, salió a defenderse tras los dichos sobre la izquierdización de la campaña de Alejandro Guillier, al nombrar como vocera a la comunista Karol Cariola. "No ocupé términos peyorativos o deshonrosos; solo establecí una posición sobre decisión de la candidatura de Guillier de otorgarle una vocería principal a la diputada, del significado que puede tener", explicó Burgos.

En condiciones de pasar a su segundo trámite legislativo al Senado quedó el proyecto que regula la elección de los nuevos gobernadores regionales, ello luego que la iniciativa fuera aprobada en general por la Cámara de Diputados.
La propuesta legal, que permite viabilizar la reforma constitucional de diciembre de 2016, fue objeto de un amplio debate, tanto en la sesión del miércoles pasado, cuando se inició la discusión, como en la presente sesión. En ambas oportunidades, los discursos manifestaron la voluntad descentralizadora de los legisladores, hecho que se ratificó a la hora de aprobar la idea de legislar de gran parte de las normas.

Chile colocó el martes deuda soberana en los mercados internacionales por unos 2.300 millones de dólares, mediante la reapertura de una emisión en euros, la oferta de un nuevo bono en dólares y la recompra de bonos.

En una primera operación, el Gobierno chileno realizó la reapertura de un bono por 700 millones de euros, con un rendimiento del 1,534 por ciento y una demanda que superó en dos veces la oferta.

Posteriormente, el gobierno ofreció deuda por 1.243 millones de dólares, con un retorno del 3,869 por ciento. La demanda representó 5,5 veces la cantidad ofertada.

¿Cómo representamos los documentos de forma numérica?

Corpus

Un **corpus** es un conjunto de documentos.

Ejemplo:

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

Vocabulario

Un **vocabulario** es una secuencia ordenada de **palabras** con un un identificador único.

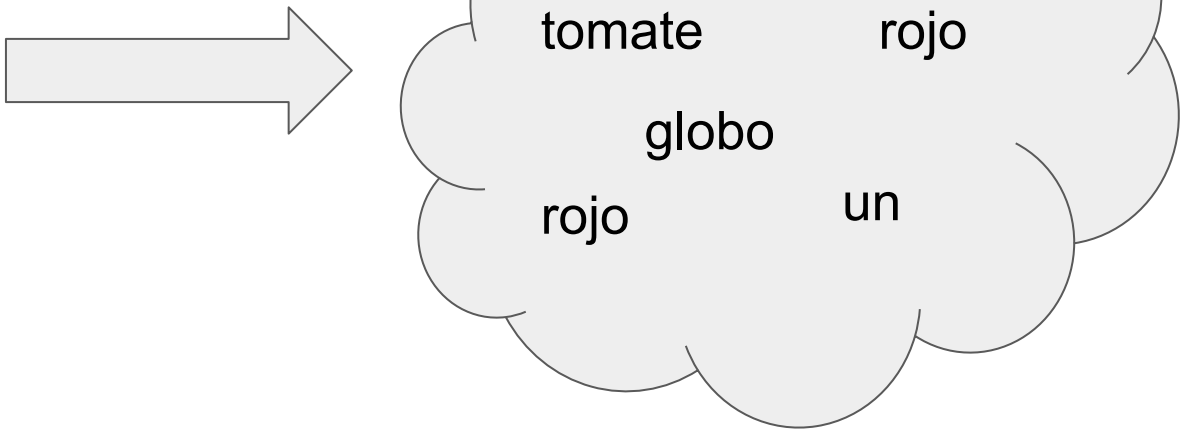
Ejemplo:

ID	palabra
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

Bag of Words *(bolsa de palabras)*

Representamos un documento como una **bolsa de palabras,** sin considerar el orden de éstas.

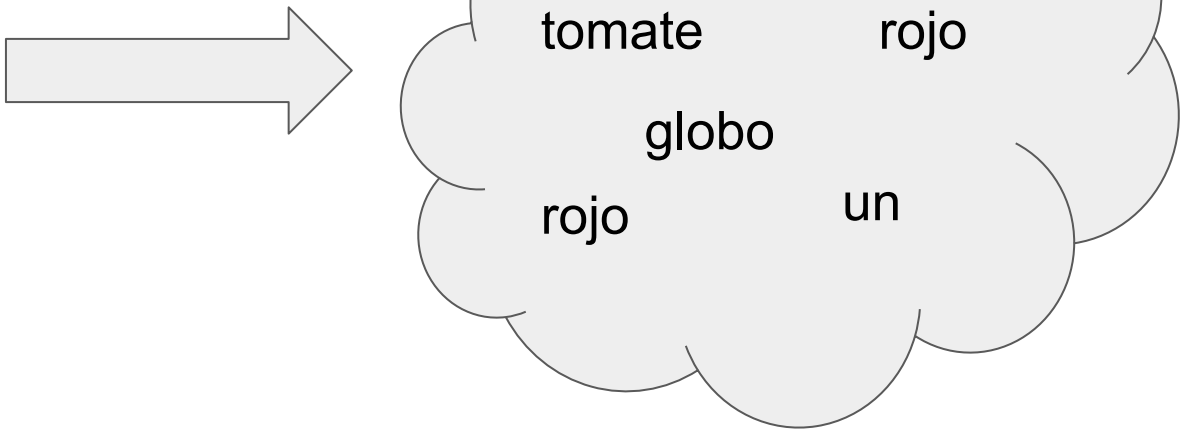
Un tomate rojo y un globo rojo.



Bag of Words *(bolsa de palabras)*

Representamos un documento como una **bolsa de palabras,** sin considerar el orden de éstas.

Un tomate rojo y un globo rojo.



Bag of Words *(bolsa de palabras)*

Podemos representar la bolsa de palabras de forma numérica, en una matriz

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Doc. 1									
Doc. 2									
Doc. 3									

ID	**palabra**
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

Bag of Words *(bolsa de palabras)*

Podemos representar la bolsa de palabras de forma numérica, en una matriz

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Doc. 1	0	1	0	0	1	0	1	0	0
Doc. 2	0	0	1	0	2	1	2	0	1
Doc. 3	1	0	0	1	0	1	2	1	1

ID	**palabra**
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

Tf-idf *(term frequency - inverse document frequency)*

La representación *Bag of Words* le asigna la misma importancia a cada palabra. ¿Está bien esto? ¿Todas las palabras nos entregan la misma cantidad de información?

- Palabras comunes:
  - Palabras como *el*, *y*, *la*, *de*, *con* que no me entregan mucha información sobre el documento.
- Palabras poco comunes:
  - Palabras como *mitocondria* me dan información acerca del contenido del texto.

Tf-idf *(term frequency - inverse document frequency)*

Podemos asignarle un peso a cada palabra de acuerdo a en cuántos documentos aparece.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Donde ***N*** es la cantidad de documentos en el **corpus** y $|\{d \in D : t \in d\}|$ es la cantidad de documentos en los que aparece la palabra ***t.***

Tf-idf *(term frequency - inverse document frequency)*

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

ID	palabra	idf
1	amarillo	
2	auto	
3	globo	
4	plátano	
5	rojo	
6	tomate	
7	un	
8	verde	
9	y	

Tf-idf *(term frequency - inverse document frequency)*

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

ID	palabra	idf
1	amarillo	0,48
2	auto	0,48
3	globo	0,48
4	plátano	0,48
5	rojo	0.17
6	tomate	0.17
7	un	0
8	verde	0,48
9	y	0.17

Tf-idf *(term frequency - inverse document frequency)*

Para representar los documentos multiplicamos la frecuencia de cada palabra **tf** por el peso calculado **idf**

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Doc. 1									
Doc. 2									
Doc. 3									

ID	**palabra**	**idf**
1	amarillo	0,48
2	auto	0,48
3	globo	0,48
4	plátano	0,48
5	rojo	0.17
6	tomate	0.17
7	un	0
8	verde	0,48
9	y	0.17

Tf-idf *(term frequency - inverse document frequency)*

Para representar los documentos multiplicamos la frecuencia de cada palabra **tf** por el peso calculado **idf**

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Doc. 1	0	0,48	0	0	0.17	0	0	0	0
Doc. 2	0	0	0	0	0.34	0	0	0	0
Doc. 3	0,48	0	0	0,48	0	0.17	0	0,48	0.17

ID	**palabra**	**idf**
1	amarillo	0,48
2	auto	0,48
3	globo	0,48
4	plátano	0,48
5	rojo	0.17
6	tomate	0.17
7	un	0
8	verde	0,48
9	y	0.17

¿Qué podemos hacer con este set de datos?

- Cualquiera de las cosas que se puede hacer con un conjunto de datos numérico como lo que ya hemos visto en el curso:
  - Clasificación
  - Clustering

Pre-procesamiento

- Algunas cosas que hacer con los textos antes de procesarlos:
  - Eliminación de stop-words
  - Stemming / Lematización

Eliminación de stop-words

- Consiste en eliminar del documento palabras muy comunes que no aportan información.

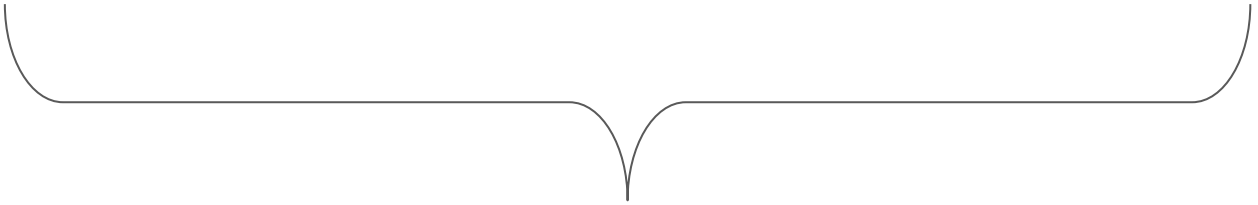
Stemming

Cambio, cambios, cambió, cambiando, cambiado

¿Deberían éstas ser consideradas la misma palabra o palabras distintas?

Stemming

Cambio, cambios, cambió, cambiando, cambiado



Cambi

Conservamos sólo la raíz de la palabra

Lematización

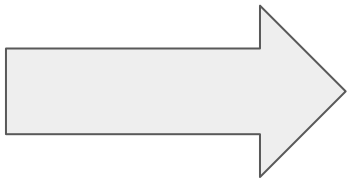
Es una alternativa a *stemming.* En vez de cortar las palabras, reemplaza cada palabra por su forma “base” no conjugada.

Fue

Será

Soy

Era



Ser

Los conjuntos de datos que hemos estudiado se ven así:

	Atrib. 1	**Atrib.2**	**Atrib. 3**	**Atrib. 4**	**Atrib. 5**	**Atrib. 6**	**Atrib. 7**	**Atrib. 8**
Obj. 1	0	0	1	2	7	0	1	0
Obj. 2	1	1	5	8	0	0	1	0
Obj. 3	1	1	0	0	5	9	3	1
Obj. 4	3	7	3	6	3	8	2	2

Dataset final

	Atrib. 1	**Atrib.2**	**Atrib. 3**	**Atrib. 4**	**Atrib. 5**	**Atrib. 6**	**Atrib. 7**	**Atrib. 8**
Obj. 1	0	0	1	2	7	0	1	0
Obj. 2	1	1	5	8	0	0	1	0
Obj. 3	1	1	0	0	5	9	3	1
Obj. 4	3	7	3	6	3	8	2	2

- Podemos aplicar todos los algoritmos vistos sobre un conjunto de este estilo
- Pero ¿habrán modelos especializados en texto?

Modelos especializados

- Existen modelos especializados en tratar de capturar información relevante de los documentos de texto.
- Uno de los más conocidos es el modelo word2vec
- Sus detalles técnicos van más allá de lo que veremos en este curso, pero veremos sus nociones.
- Para saber más de estos temas, recomiendo tomar el curso de Aprendizaje Profundo

word2vec

- Modelo basado en redes neuronales
- Trata de transformar las palabras de un conjunto de documentos en un vector de características, de ahí el nombre.
- Fue el origen de los modelos "2vec" (doc2vec, prod2vec, vec2vec)

word2vec

- Tal como en los métodos anteriores, se genera un vocabulario de todas las palabras existentes.
- Se genera una ventana de palabras, siendo el número de palabras que considera la oración.
- En esta ventana, palabra es transformada en un ***one hot encoding***

I	1	0	0	0	0
Like	0	1	0	0	0
watching	0	0	1	0	0
movie	0	0	0	1	0
enjoy	0	0	0	0	1

https://medium.com/towards-artificial-intelligence/an-intuitive-introduction-of-word2vec-by-building-a-word2vec-from-scratch-a1647e1c266c

word2vec

- La intuición detrás de este modelo es el hecho de que palabras utilizadas en contextos similares, deberían tener significado similar.

- Por ejemplo si tuviera las oraciones:
  - *I like watching a movie.*
  - *I enjoy watching a movie.*

- Intuitivamente entendemos que *like* y *enjoy* deben tener un significado similar.

- Ahora nos surge la duda ¿Cómo aprende este contexto?

word2vec

- Dado los vecinos de una palabra, trata de aprender un vector latente que lo represente, también conocidos como ***Embeddings***.
- Para cada palabra, genera un vector de entrada y uno de salida.
- La palabra objetivo la deja en el vector de salida, y el contexto o las palabras que la acompañan las deja en el vector de entrada.

Entrada

like	watching	movie
I	watching	movie
I	like	movie
I	like	watching
enjoy	watching	movie
I	watching	movie
I	enjoy	movie
I	enjoy	watching

I
Like
watching
movie
I
enjoy
watching
movie

Salida

https://medium.com/towards-artificial-intelligence/an-intuitive-introduction-of-word2vec-by-building-a-word2vec-from-scratch-a1647e1c266c

word2vec

- Cada uno de estos vectores es transformado en formato *one hot encoding*

Vectorized input

I	like	watching	movie	enjoy
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0
0	0	1	1	1
1	0	1	1	0
1	0	0	1	1
1	0	1	0	1

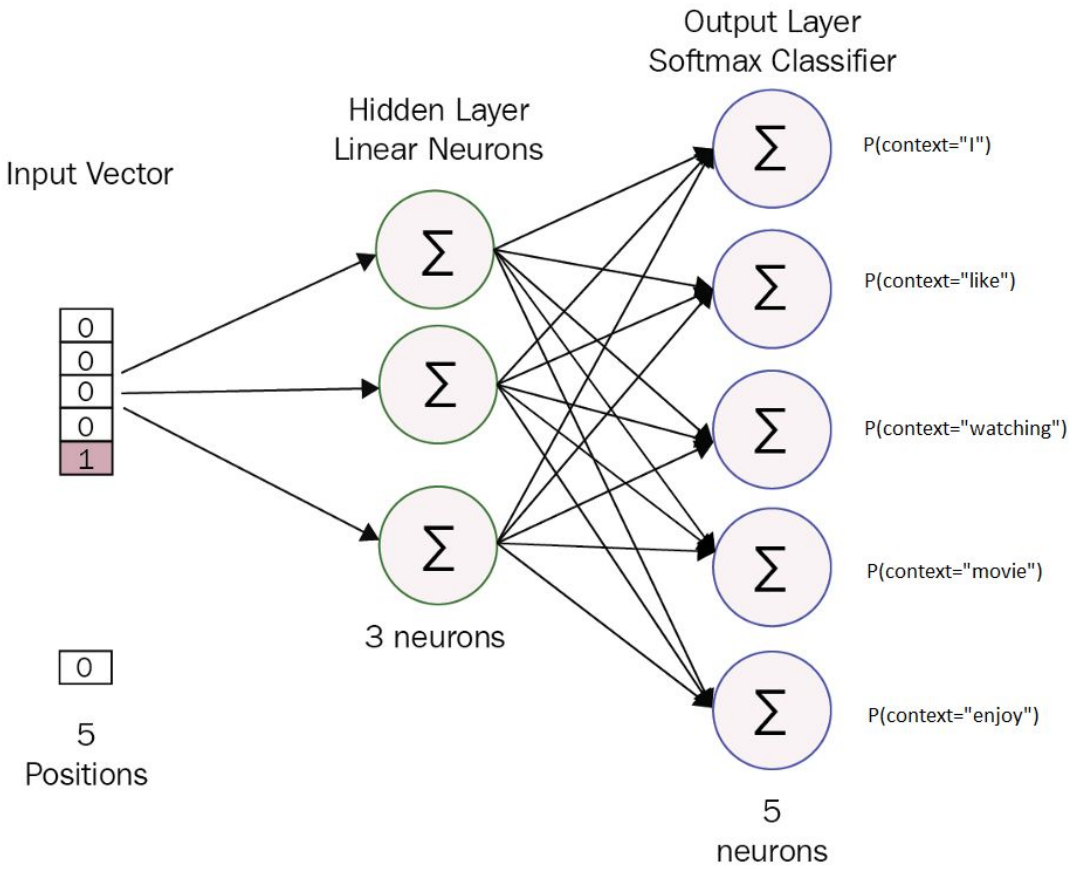
Output Vector

I	like	watching	movie	enjoy
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
1	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	0	0	1	0

https://medium.com/towards-artificial-intelligence/an-intuitive-introduction-of-word2vec-by-building-a-word2vec-from-scratch-a1647e1c266c

word2vec

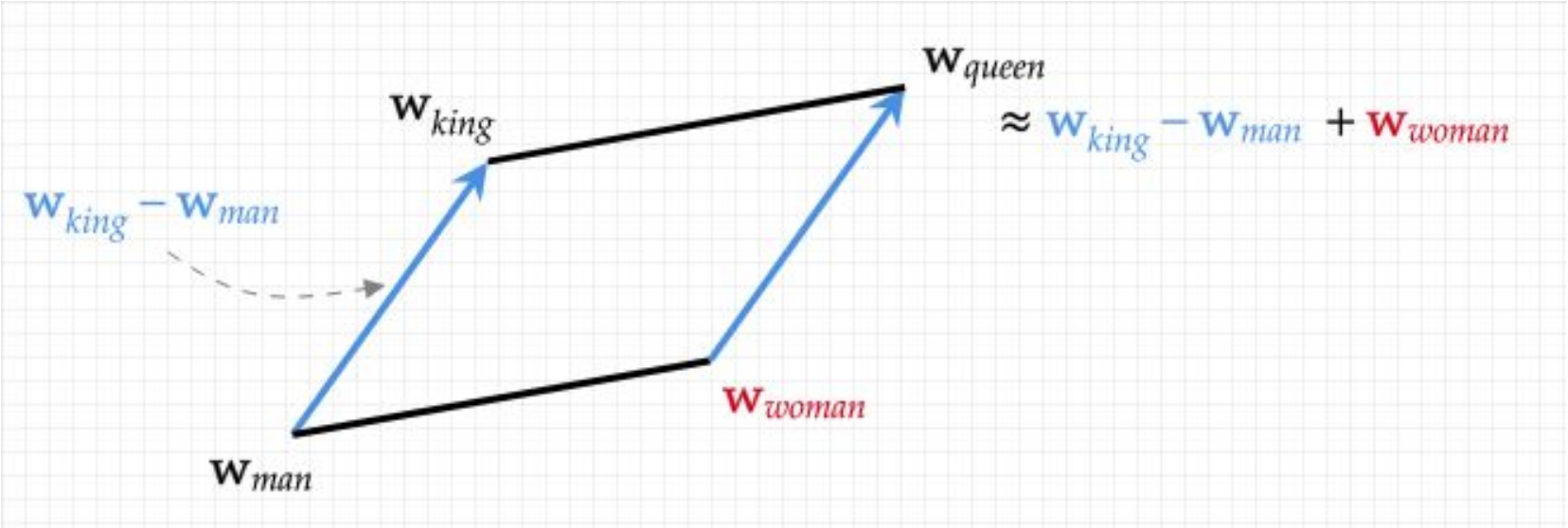
- Finalmente, la arquitectura del modelo es una red neuronal que trata de aprender un vector de *embedding* para cada palabra



https://medium.com/towards-artificial-intelligence/an-intuitive-introduction-of-word2vec-by-building-a-word2vec-from-scratch-a1647e1c266c

word2vec

- De los *embeddings* obtenidos, ocurren relaciones como la siguiente



https://www.ed.ac.uk/informatics/news-events/stories/2019/king-man-woman-queen-the-hidden-algebraic-struct

word2vec

Comentarios finales

- Modelo ampliamente utilizado en distintas áreas.
- Se puede utilizar con sus propios datos, o utilizar algún modelo pre entrenado.
- Existen modelos pre entrenados con conjuntos de datos masivos (todo wikipedia) incluso en español.
- https://github.com/dccuchile/spanish-word-embeddings
- Hay muchos otros modelos, sobre distintas áreas, pero van más allá de lo que podemos ver en este curso. Hay mucho por aprender aún 👀