Procesamiento de Lenguaje Natural

DIPLOMA/MAGISTER EN INTELIGENCIA ARTIFICIAL
UNIVERSIDAD ADOLFO IBÁÑEZ

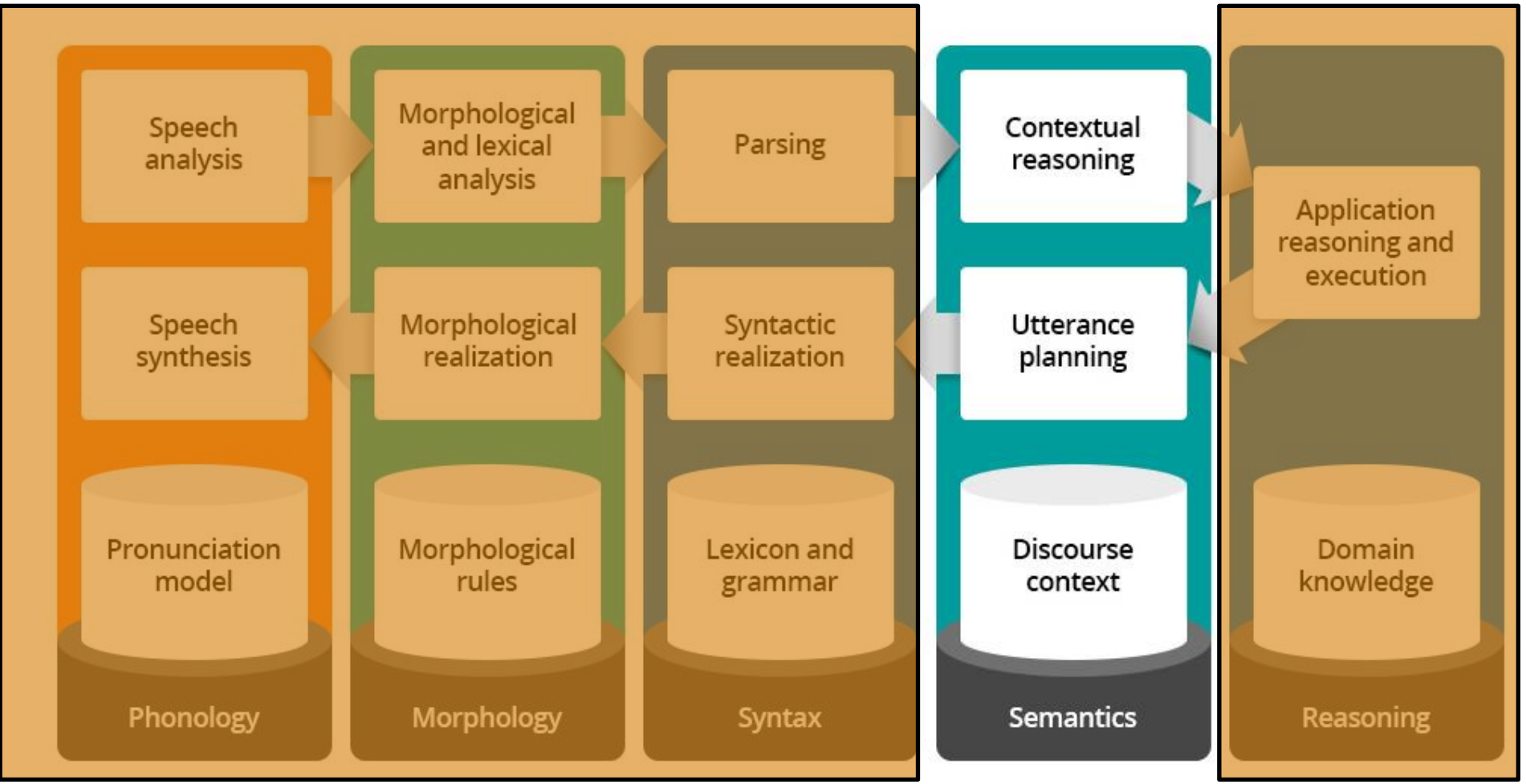
Profesor: Dr. John Atkinson

Análisis Semántico

OBJETIVO

Entender la manera en que se determina el significado (*semántica*) de las palabras y oraciones en lenguaje natural.

ETAPAS EN NLP



RAZONAMIENTO CONTEXTUAL

El *razonamiento contextual* involucra dos tipos de análisis:

- **Análisis Semántico** (el significado literal).
- **Análisis Discursivo** o **Pragmático** (lo que se intenta comunicar).

Análisis Semántico

- Tarea de determinar automáticamente el significado o *semántica* de las palabras y/o expresiones en lenguaje natural.

- O sea, la *semántica* es el estudio de lo que las palabras (u oraciones) *quieren decir* cuando hablamos o escribimos.

Semántica

La *semántica* en el lenguaje se puede estudiar desde tres perspectivas:

1. El significado de palabras individuales (*semántica léxica*).

2. La forma en que se combinan estos significados para formar significados para oraciones individuales.

3. La forma en que dichos significados se combinan para producir significados para un texto o discurso.

Semántica

La *semántica* en el lenguaje se puede estudiar desde tres perspectivas:

1. El significado de palabras individuales (*semántica léxica*).

2. La forma en que se combinan estos significados para formar significados para oraciones individuales.

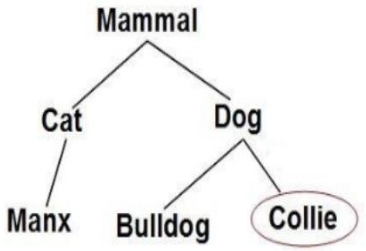
3. La forma en que dichos significados se combinan para producir significados para un texto o discurso.

Semántica Léxica

¿De dónde puedo *extraer* (o *inferir*) automáticamente el *significado* de las palabras?

Diccionario Electrónico

Taxonomía Conceptual



Contexto en que ocurren las palabras

HOME | VIDEOS | INDIA | MOVIES | TECH | SPORTS | FACT CHECK | NEWSMO | TRENDING | SAFAIGIRI AV

News / Trending News /

Street dog walks to pharmacy and shows injured paw. Adorable viral video has Internet in tears

A video of a street dog walking up to a pharmacist in Istanbul so as to ask for help for its injured paw has gone viral on social media.

Relaciones entre Palabras

Un aspecto en común es que el *significado* de las palabras se obtiene a partir de la *cercanía* de una palabra con otras cercanas: relaciones semánticas.

Ejemplo:

- Juan es un cliente del banco.
- Un perro es un animal de cuatro patas que usualmente las personas poseen como mascota para seguridad o para casería..

¿De dónde se obtienen?

Algunos tipos de relaciones de propósito general:

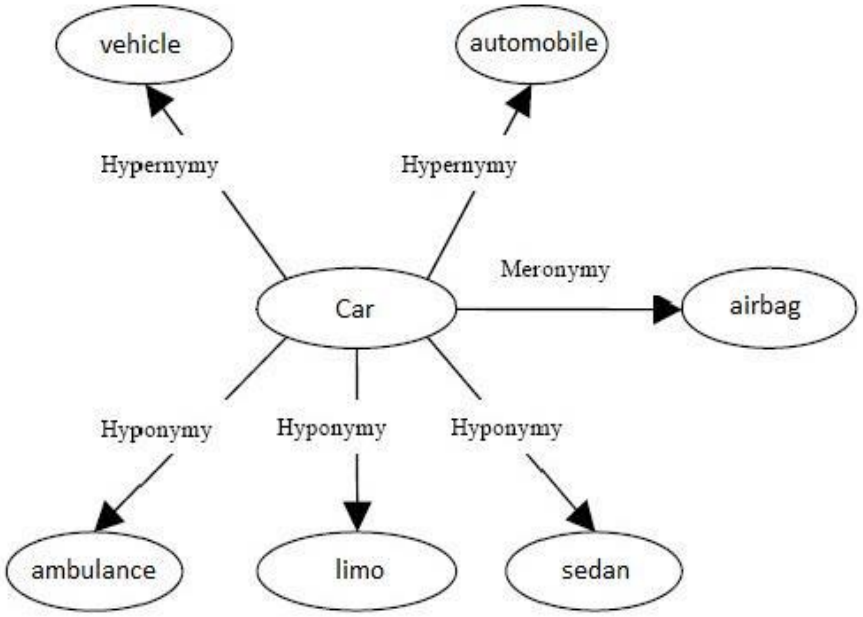
Homonimia: palabras que se escriben igual pero tienen significado diferente.
Sinonimia: diferentes palabras que tienen el mismo significado.
Hiponimias: palabras que son una subclase o subconjunto de otras.
etc

Ejemplo:

Perro es un hipónimo de mamífero

¿De dónde se obtienen?

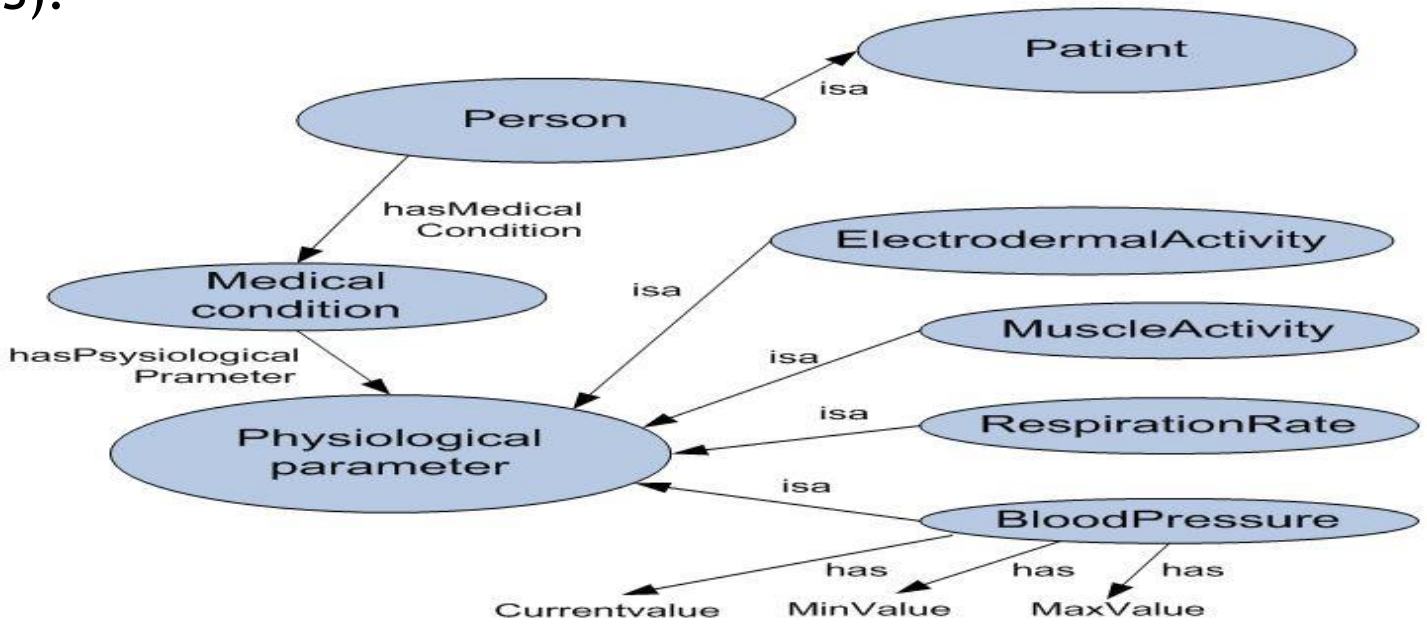
Si los conceptos son de **propósito general**, las relaciones se pueden obtener de una **ontología** o **taxonomía** electrónica (ej. *WordNet*, *ConceptNet*):



http://wordnetweb.princeton.edu/perl/webwn

¿De dónde se obtienen?

Si los conceptos son de **propósito específico**, necesitamos una ontología especializada para obtener las relaciones (ej. Tesauros en *UMLS*):



https://www.nlm.nih.gov/research/umls/index.html

Aplicaciones

- Encontrar el significado de una palabra para responder una pregunta de un sistema de pregunta-respuesta (ej. Alexa).
- Buscar documentos *más similares* a una consulta.
- Agrupar quejas de clientes.
- Caracterizar los términos/palabras más adecuados al significado de un documento (ej. para clasificar clientes).
- Determinar la polaridad de una opinion en redes sociales.
- Muchas más.

Problemas

- No siempre se disponen de *bases de conocimiento* específicas.
- Las palabras pueden tener más de un significado posible (o sea, varias relaciones posibles).
- Luego, tenemos problemas de **ambigüedad**: debemos encontrar el mejor sentido (*significado*) para una palabra dado varios posibles sentidos.

Word Sense Disambiguation (WSD)

Palabras en contexto:

Debo ir al **banco** en la mañana…

En este mes, el **banco** acumuló más cm3 cúbicos de sangre que todo el año…

…

Sentidos posibles:

Banco1: institución financiera
Banco2: Registro de muestras sanguíneas
…

WSD

El mejor sentido:

Tuve problemas con mi cuenta en el **banco**…

Banco1

¿Cómo se realiza WSD?

Necesitamos:

- ✓ *Un conjunto de etiquetas de sentidos.*
- ✓ *El corpus de entrenamiento etiquetado.*
- ✓ *Un conjunto de* ***features*** *extraído desde el corpus de entrenamiento.*
- ✓ *Un método que permita determinar la mejor similitud a partir de los* ***features*** *de entrenamiento.*

¿Cómo extraemos *Features*?

Bag-of-Words

Features sobre palabras que ocurren en cualquier lugar de una *ventana* (no depende de su posición).

Colocacional

Features sobre palabras en posiciones específicas cerca de la palabra.

¿Cómo determino la *Similitud*?

- Cualquier representación de **features** debería considerar la **cercanía** entre palabras como propiedad.
- Necesitamos medidas de **similitud** semántica o cercanía entre palabras o **features**.

Dos palabras son más similares si comparten más *features* del significado!!

Cálculo de *Similitud*

- ✓ Métodos basados en *Taxonomías*:
  - Similitud es una medida de la *distancia* entre términos en una *taxonomía*.
- ✓ Métodos basados en *Distribución*:
  - Cálculo basado en la distribución estadística de los términos en cierto contexto.
- ✓ Métodos basados en *Embeddings*:
  - LSA, Word2vec, BERT, GPT, etc

Representación de Features: *Intuición*

- Necesitamos una forma de representar el **contexto** de cada palabra.
- Considere una palabra **w**.
- Suponga que tenemos un ***feature binario*** f_i para cada una de las *N* **palabras** en un vocabulario (*lexicón*) **V**.
- O sea una "*palabra* V_i *ocurre en la vecindad de* ***w***": $W=(f_1,f_2,\ldots f_N)$

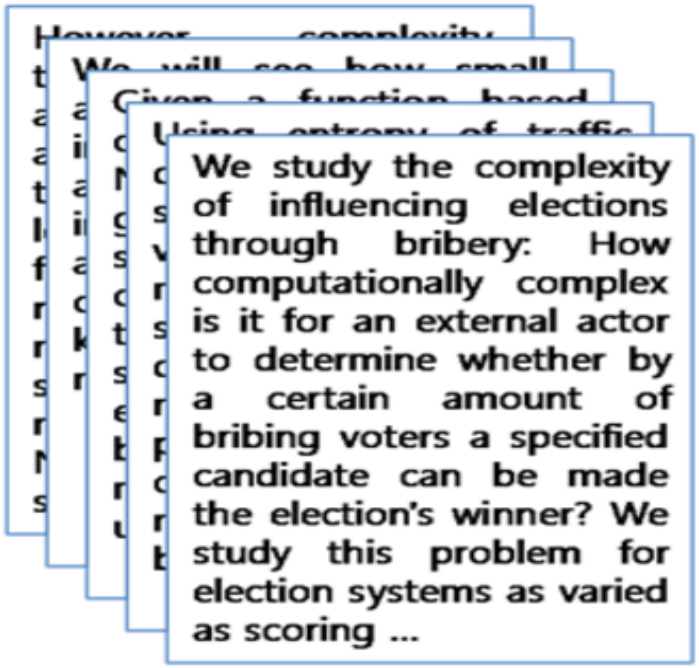
Representación de Features: *Intuición*

Representemos dos palabras según sus *features*:

		Features				
		se mueve	vivo	cuatro patas	tiene pelo	vuela
Palabras	PERRO	1	1	1	1	0
	PAJARO	1	1	0	1	1

Representación Vectorial

Documentos



Representación de Espacio Vectorial

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Ahora, estos son los features

Necesitamos Vectorizar

- Generemos una representación:
  - ✓ Podemos utilizar *Bag-of-Words* (BOW).
  - ✓ Opcionalmente podemos eliminar palabras irrelevantes (stopwords).
  - ✓ Lematizar (análisis morfológico).
- Ponderemos la *importancia* de los features: varios modelos de *pesos* (ponderaciones).

¿Cómo *Ponderamos*?

FORMA DIRECTA

Se pondera proporcional a la frecuencia de aparición de las palabras.

FORMA INDIRECTA

Método de aprendizaje genera la representacion y ponderación, como vectores en bajas dimensiones.

¿Cómo *Ponderamos*?

FORMA DIRECTA

Se pondera proporcional a la frecuencia de aparición de las palabras.

FORMA INDIRECTA

Método de aprendizaje genera la representacion y ponderación, como vectores en bajas dimensiones.

MODELO *TF*

*Por **Frecuencia de Términos** (tf):*

3 documents →

d1: "new york times"
d2: "new york post"
d3: "los angeles times"

	angeles	los	new	post	times	york
d1	0	0	1	0	1	1
d2	0	0	1	1	0	1
d3	1	1	0	0	1	0

W(i,j) = tf(i,j) (el peso es simplemente la frecuencia)

tf(i,j): frecuencia de **i**-ésimo término en un documento **j**

MODELO *TF x IDF*

*Por **Frecuencia Inversa de Documentos**(tf x idf):*

Recordemos de dónde veníamos:

$$W(i,j) = tf(i,j)$$

MODELO *TF x IDF*

*Por **Frecuencia de Documento Inversa** (idf):*

- ✓ Observe que si una palabra ocurre en:
  - *Demasiados documentos*: *menor poder discriminador.*
  - *Pocos documentos*: *mayor poder discriminador.*
- ✓ Debemos castigar la frecuencia de una palabra si es que esta ocurre en muchos documentos.
- ✓ Así, el peso podría reformularse como:

W(i,j) = tf(i,j) * (algún factor de castigo(i))

W(i,j) = tf(i,j) * IDF(i)

W(i,j) = tf(i,j) * log(N/df(i))

Donde **N** es el núm. de documentos y **df(i)** es el número de documentos donde ocurre la **i**-ésima palabra.

MODELO *TF x IDF*

3 documents

d1: "new york times"
d2: "new york post"
d3: "los angeles times"

TERM	DOC-FREQUENCY	IDF
angeles	1	$\log_2(3/1) = 1.584$
los	1	$\log_2(3/1) = 1.584$
new	2	$\log_2(3/2) = 0.584$
post	1	$\log_2(3/1) = 1.584$
times	2	$\log_2(3/2) = 0.584$
york	2	$\log_2(3/2) = 0.584$

Matriz TF

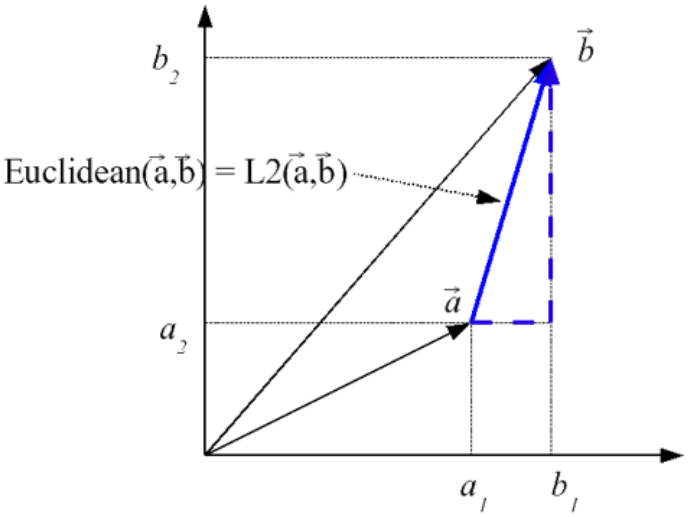
	angeles	los	new	post	times	york
d1	0	0	1	0	1	1
d2	0	0	1	1	0	1
d3	1	1	0	0	1	0

Matriz TF x IDF

	angeles	los	new	post	times	york
d1	0	0	0.584	0	0.584	0.584
d2	0	0	0.584	1.584	0	0.584
d3	1.584	1.584	0	0	0.584	0

Medidas de *Distancia o Similitud*

Medidas geométricas:



Cercanía coseno:

$$m_{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Aplicación: *Detección de Plagio*

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

¿Cómo *Ponderamos*?

FORMA DIRECTA

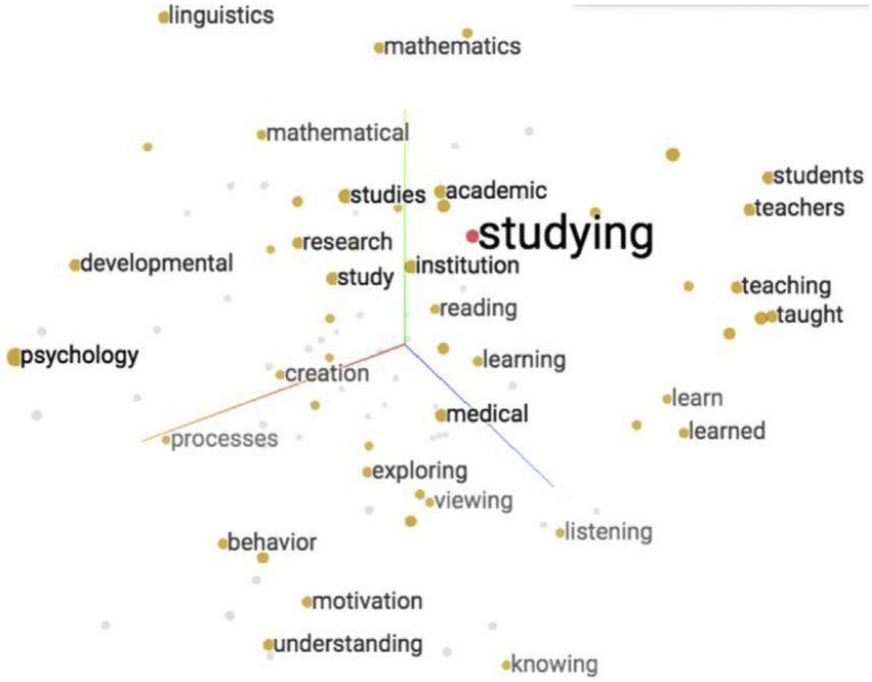
Se pondera proporcional a la frecuencia de aparición de las palabras.

FORMA INDIRECTA

Método de aprendizaje genera la representacion y ponderación, como vectores en bajas dimensiones.

PONDERACIÓN INDIRECTA

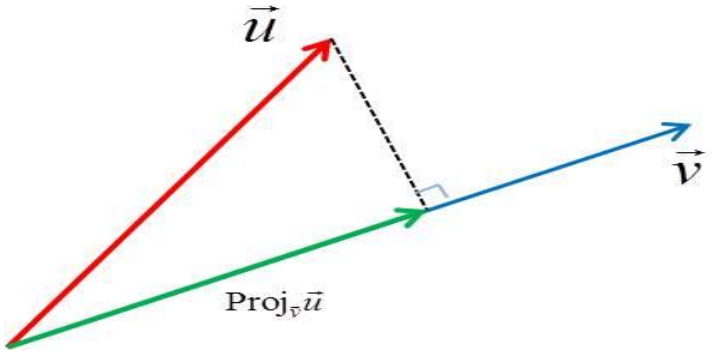
- Existen varias formas de generar representaciones vectoriales eficientes.
- Estas pueden aprenderse automáticamente mediante vectores en bajas dimensiones (*embeddings*).



Word Embeddings

Word Embedding: se refiere a cualquier técnica que *transforme* una palabra (o frase) desde su entrada altamente dimensional (muchos *features*!) a un espacio de *baja dimensión*.

Así, uno *incrusta* (*embed*) las palabras en un espacio diferente.



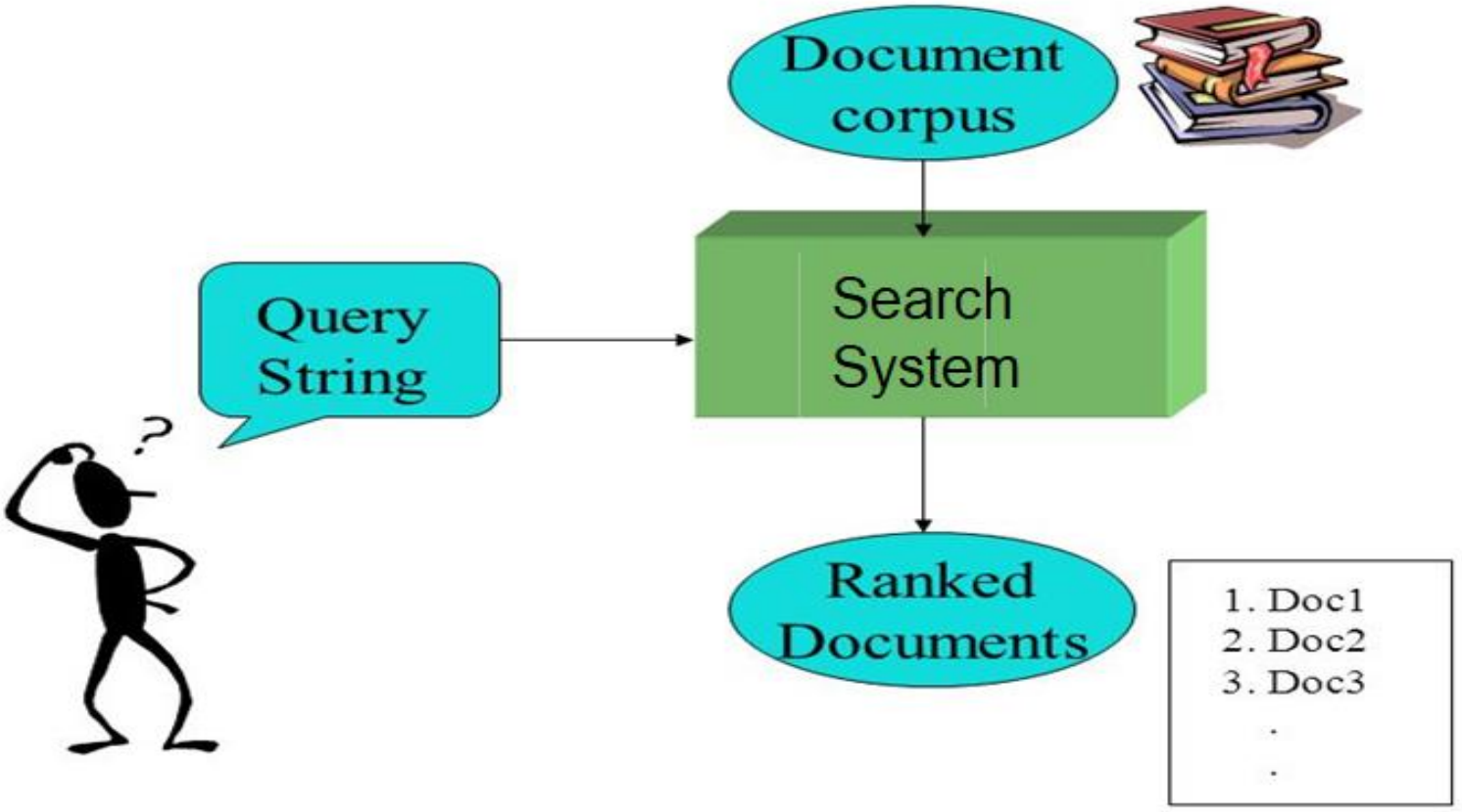
Métodos basados en *Corpus*

Recordar: La representación vectorial debe capturar los mejores ***features*** que expresen la *semántica* de las palabras!!.

- **Principio**: *si dos palabras son semánticamente similares (en la realidad), la distancia entre sus vectores de features deberían ser cercanos.*

- **Problema**: un texto es altamente *dimensional* por lo que los modelos convencionales (i.e., TF, TFxIDF) no son adecuados (i.e., ruido, dimensionalidad, etc).

Búsqueda de Documentos



Problemas

POLISEMIA

Análisis de palabras podría recuperar documentos *irrelevantes*.

SINONIMIA

Análisis de palabras falla al tratar de recuperar documentos *relevantes*.

Problemas

- Asumamos que un texto se puede representar como un Bag-of-Words (por ahora) y estas representan inicialmente las *dimensiones* de dicho texto.
- Generalmente, un texto posee MUCHAS dimensiones.

¿Necesitamos todas?

- Mucho tiempo de procesamiento.
- No todas caracterizan al texto.
- Problemas de representación (ej. ruido, espacio, polisemia, etc).

Intuición

- Podríamos intentar **reducir** las *dimensiones* de la representación inicial.
- Al hacerlo, podrían aparecer relaciones desconocidas (implícitas) entre las palabras.
- Así, podemos representar las palabras (o documentos) como ***word embeddings*** o *espacios semánticos* en dimensiones reducidas.

Métodos para Generar *Embeddings*

- ✓ Métodos basados en descomposiciones de matrices:
  - **Latent Semantic Analysis** (LSA).
- ✓ Métodos basados en *redes neuronales artificiales (ANN)*:
  - Aprendizaje mediante ANN del tipo feed-forward: **Word2Vec** **GLoVe**, etc.
  - Aprendizaje profundo mediante *transformers bi-direccionales*:
    - **BERT** (*Google*), **GPT** (*OpenAI*), y varios otros.

Análisis Semántico Latente (*LSA*)

- LSA es una técnica de aprendizaje automático no-supervisado que permite generar un espacio semántico (embeddings) que representa las palabras de un corpus.

- *Hipótesis*: Existe una **estructura latente** en el uso de las palabras – *obstaculizado por la variabilidad en la selección de palabras*.

Análisis Semántico Latente (*LSA*)

Ideas principales:

Intuición: El contexto de palabras que rodea a una palabra dada, incide linealmente en la **representación semántica** de esta palabra.

- Comenzamos representando las palabras de un corpus con un modelo vectorial tradicional.
- Luego, realizamos reducción dimensional del modelo original para aprender la representación contextual de las palabras.
- La nueva representación (*embeddings*) aprendida captura relaciones semánticas ***latentes*** (*ocultas*) entre palabras.

¿Cómo funciona?

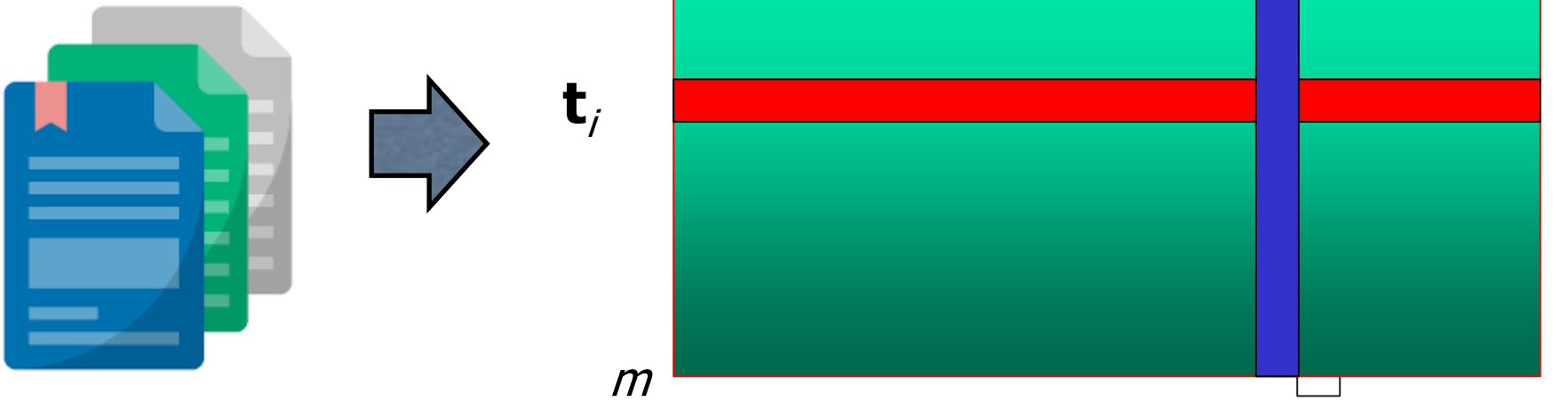
Dado un corpus de documentos **C**:

1. Crear un modelo de frecuencias ***X*** a partir de **C**.
2. Descomponer X en tres matrices que representan los impactos que tiene la vecindad en el significado de cada palabra. Utilizamos el método **SVD**.
3. Reconstruir la matriz original, *X'*, con un número menor de dimensiones.

X' corresponde al **espacio semántico** o **embeddings**.

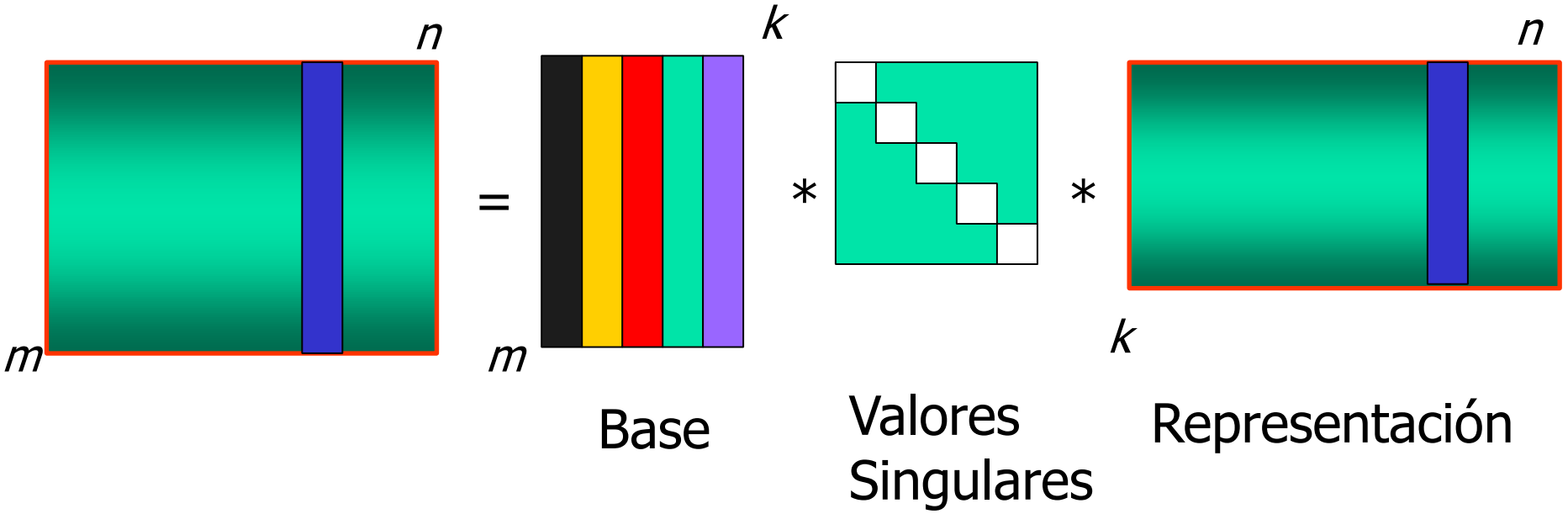
Paso (1): Representación inicial

Modelo inicial ***tf***



Paso (2): Descomposición por SVD

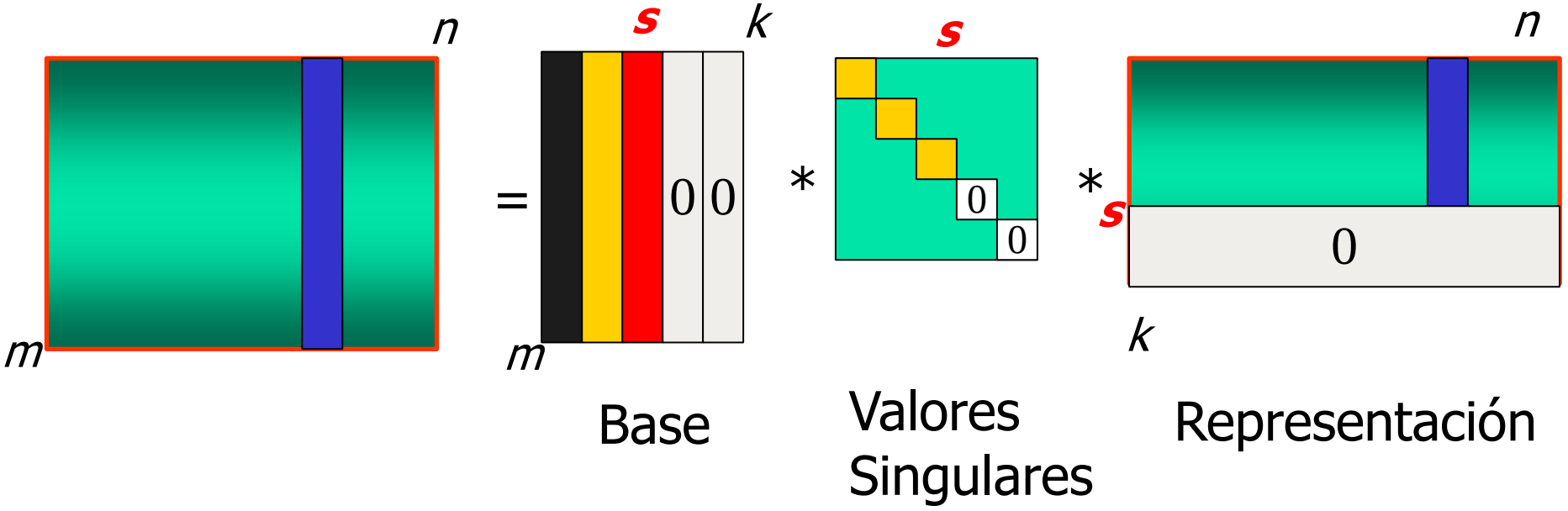
$$X = W * S * V^T$$



Restricciones sobre representación: **W, V** ortonormal; S **diagonal**

Paso (3): Reconstrucción en s dimensiones

$X_s = W * S_s * V^T$



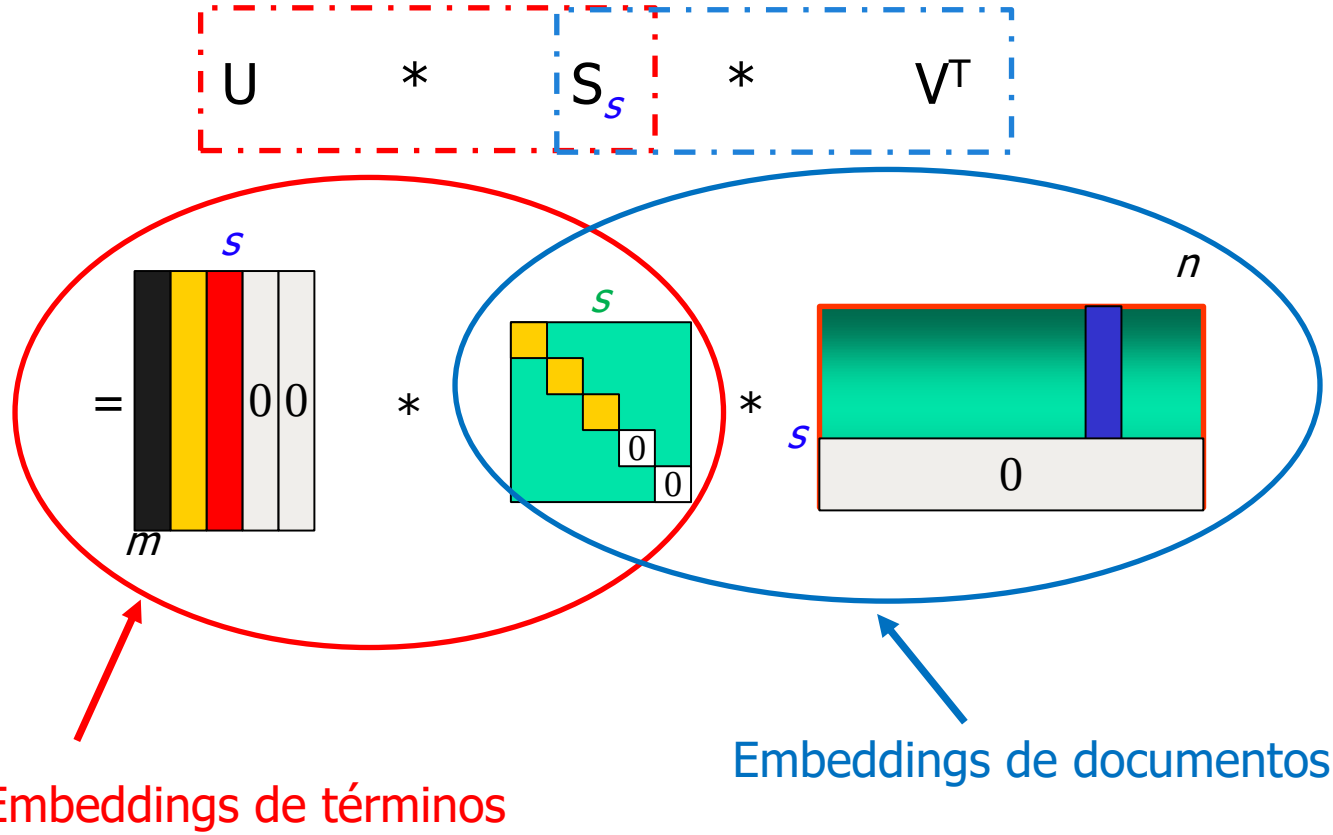
Base

Valores Singulares

Representación

Reconstrucciones de espacio inicial en ***s*** << ***m*** dimensiones.

¿Cómo obtengo los Embeddings?



S representa los features de los términos

Embeddings de términos

Embeddings de documentos

Ejemplo

Asumamos 9 “documentos” cortos (sólo títulos), de los cuales: 5 “hablan” sobre temas de “*computación*” (c_i) y 4 “hablan” sobre temas de “*matemáticas*” (m_i):

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Espacio Inicial

$$\{X\} =$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

¿Cuál es la cercanía entre el vector de *human* y *user*?
¿Cuál es la cercanía entre el vector de *human* y *minors*?

Descomposición SVD

$$\{X\} = \{W\}\{S\}\{P\}^{'}$$

$\{W\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$\{S\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$\{P\} =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Reconstrucción con 2 dimensiones

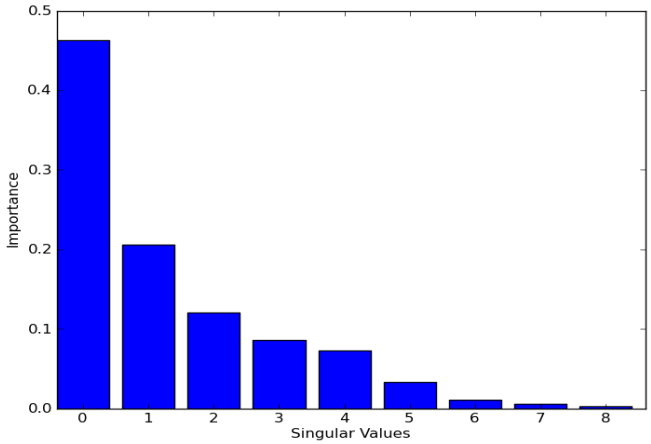
$$\{\hat{X}\} =$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

¿Cuál es la cercanía entre el vector de *human* y *user*?
¿Cuál es la cercanía entre el vector de *human* y *minors*?

¿Cómo elijo el número de dimensiones? (1)

- ✓ ***Corpus grandes***: entre 100 y 500 dimensiones (aprox).
- ✓ ***Corpus pequeños***: construir histograma de *importancia* (i.e., *cuadrado de los valores singulares*) vs valores singulares (dimensiones). Luego, elegir la de mayor *importancia* (*descartando la 1era dimensión*):



¿Porqué?

- *Para documentos*, la 1era dimensión correlaciona con el largo del documento.
- *Para palabras*, la 1era dimensión correlaciona con la frecuencia de las palabras en todos los documentos.

¿Cómo elijo el número de dimensiones? (2)

Otra forma de elegir es considerando la tarea posterior que se realiza con los *embeddings*:

- ✓ *Clasificación*: podría elegir el número de dimensiones que ayudan a incrementar el *accuracy* de la clasificación.
- ✓ *Similitud entre documentos*: podría elegir número de dimensiones que permitan obtener la similitud más realista entre documentos (o palabras).
- ✓ Otras tareas.


Tiempo de Ejercicios

Ejercicio (1)

- ✓ Cargue en *Google Colab* el programa **`lsa.py`**.
- ✓ Ajuste la ruta del corpus de ejemplo de los programas.
- ✓ Ejecute el programa utilizando el corpus previamente proporcionado.


CASE STUDY

Problema

- ✓ Una empresa recibe muchas opiniones de clientes a través de las redes sociales.
- ✓ Se desea conocer la percepción de dichos clientes en términos de sus sentimientos implícitos: positivos (1) o negativos (0).
- ✓ Dada la cantidad de datos, esta actividad no puede realizarse manualmente.

Posible Solución

1. Recolectar opiniones previamente “etiquetadas” con sentimientos (1/0) para generar un modelo de clasificación de sentimientos.
2. Obtener los embeddings LSA para cada opinión de (1).
3. Separar los embeddings en un set de training y otro de test.
4. Entrenar un modelo simple de clasificación bayesiana con el set de training.
5. Probar el modelo con el set de test.

Implementación

- ✓ Cargue en *Google Colab* el programa **`clasificador_lsa.py`**.
- ✓ El programa implementa el clasificador de sentimientos utilizando LSA.
- ✓ Siga las instrucciones del profesor.