

Introducción a Recuperación de Información

y Visualización de Texto

Denis Parra

(algunas slides de Lucas Valenzuela)

IIC1005 – PUC Chile

¿Cómo representa y opera los documentos un buscador como Google o DuckDuckGo?

The Google logo, consisting of the word "Google" in its characteristic multi-colored font.A white search bar with rounded ends, featuring a magnifying glass icon on the left and a microphone icon on the right.

Google Search

I'm Feeling Lucky



DuckDuckGo

A white search bar with rounded ends, featuring a magnifying glass icon on the right.

El buscador que no te rastrea. [¡Ayuda a difundir DuckDuckGo!](#)

Recuperación de Información

La recuperación de información (IR) es encontrar material (generalmente documentos) de una naturaleza no estructurada (generalmente texto) que satisface una necesidad de información desde grandes colecciones (generalmente almacenadas en computadoras).

¿Por qué necesitamos una clase especial para
modelar texto?
(Después de todo, son sólo datos)

Los conjuntos de datos que estamos acostumbrados (estructurados) 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

¿y por qué visualizar texto?

- Porque son datos no estructurados
- Porque es uno de los tipos de datos más comunes
- Porque visualizar texto nos puede dar muchos “insight”
- Porque es bueno saber qué tipo de “modificaciones” pudo sufrir el texto antes de ser visualizado

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 deshonorosos; 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
para poder procesarlos con un
computador?

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.

Documento

Un documento podría ser, por ejemplo

- **Cada línea de un archivo de texto**
- **Cada archivo de texto dentro de una carpeta**
- **Un campo particular dentro de una base de datos relacional**
- **Otros...**

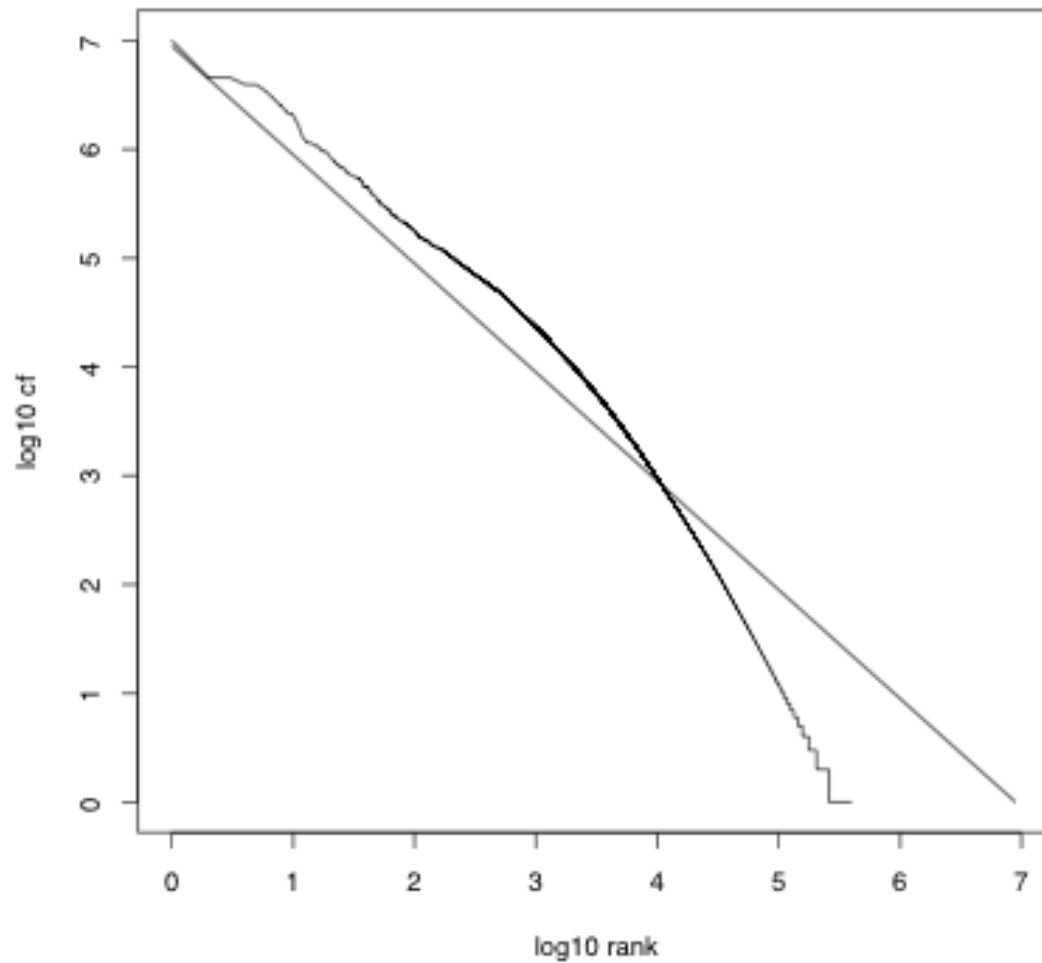
Vocabulario

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

Visualizar un Vocabulario: Ley de Zipf



- El término más frecuente (el) ocurre cf_1 veces
- El segundo término más frecuente (de) ocurre $cf_2 = cf_1/2$ veces
- El tercer término más frecuente (y) ocurre $cf_3 = cf_1/3$ veces
- ...

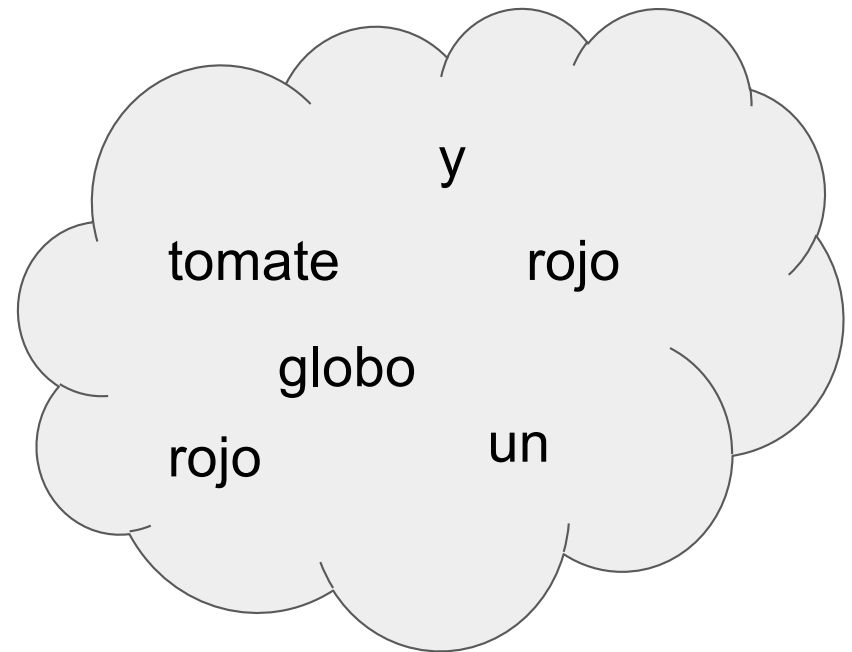
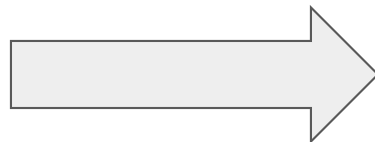
Otras Definiciones

- **Término (Term)** – A “normalized” word (case, morphology, spelling etc); an equivalence class of words.
- **Palabra (Word)** – A delimited string of characters as it appears in the text.
- **(Token)** – An instance of a word or term occurring in a document.

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	0	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

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

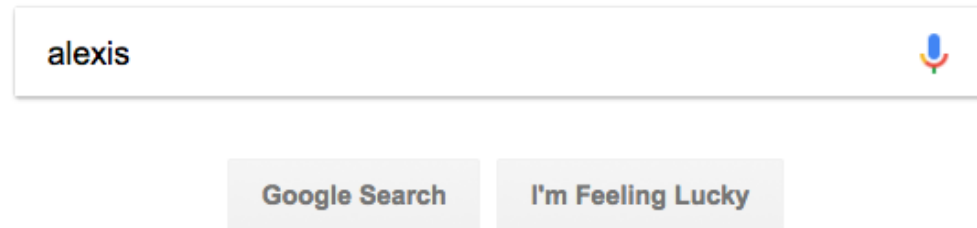
Pesos de las palabras (weighting)

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.

Pesos de las palabras (weighting)

Dada una consulta $q = \{\text{alexis}\}$



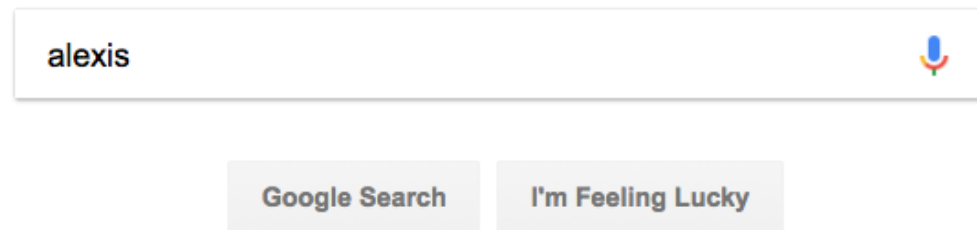
alexis

Google Search I'm Feeling Lucky

- Un documento con $tf = 10$ ocurrencias del término *alexis* es más relevante que un documento con $tf = 1$ ocurrencias del término
- Pero no es 10 veces más relevante
- Relevancia no crece proporcionalmente con la frecuencia del término

Pesos de las palabras (weighting)

Dada una consulta $q = \{\text{alexis}\}$



alexis

Google Search I'm Feeling Lucky

- Otra estrategia para “pesar” las palabras: logaritmo

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{if } \text{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- $\text{tf}_{t,d} \rightarrow w_{t,d}$: $0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4$, etc.

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

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

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

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*)

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

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*)

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

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

Variantes del pesado tf-idf

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^\alpha$, $\alpha < 1$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$				

¿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:
 - Ranking (dada una palabra o dado un documento)
 - Clasificación
 - Clustering

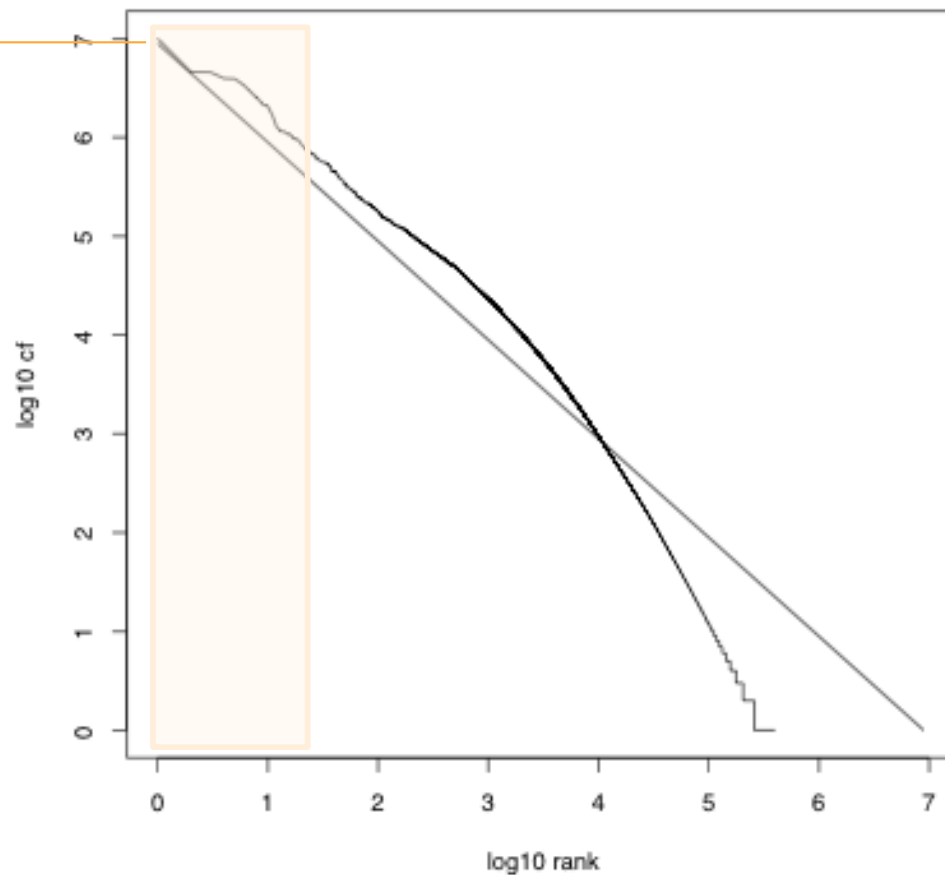
Pre-procesamiento

- Algunas cosas que hacer con los textos antes de procesarlos:
 - Normalizar (pasar todo a minúsculas, *case folding*)
 - Eliminación de stop-words (dependiendo de la tarea)
 - Stemming
 - Lematización
 - Tokenización (tokenization)
 - n-grams

Eliminación de stop-words

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

Estas palabras ocurren en casi todos los documentos, y puede ser preferible eliminarlas



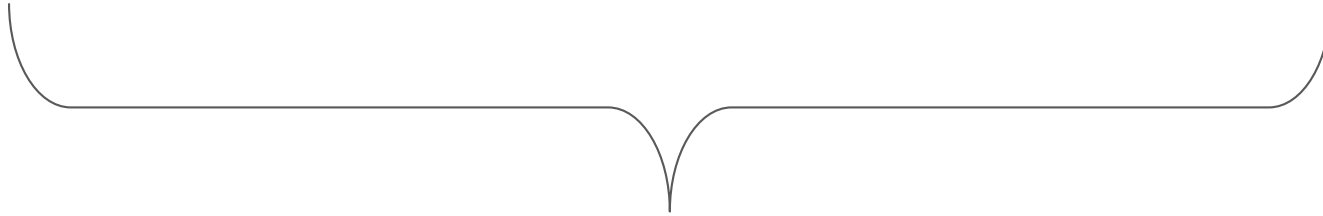
Stemming

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

Ejemplos de algoritmos de stemming

- Sample text:* Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Porter stemmer:* such an analysis can reveal features that are not easily visible from the variation in the individual gene and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Lovins stemmer:* such an analysis can reveal features that are not easily visible from the variation in the individual gene and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- Paice stemmer:* such an analysis can reveal features that are not easily visible from the variation in the individual gene and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lematización

Soy, es, eres, somos, son -> SER

Al usar lematización, queremos hacer una reducción apropiada a una palabra raíz (lema)

N-gramas

● $N = 1$: Esta es una oracion unigramas

- Esta,
- es,
- una,
- oracion

● $N = 2$: Esta es una oracion bigramas

- Esta es,
- es una,
- una oracion

● $N = 3$: Esta es una oracion trigramas

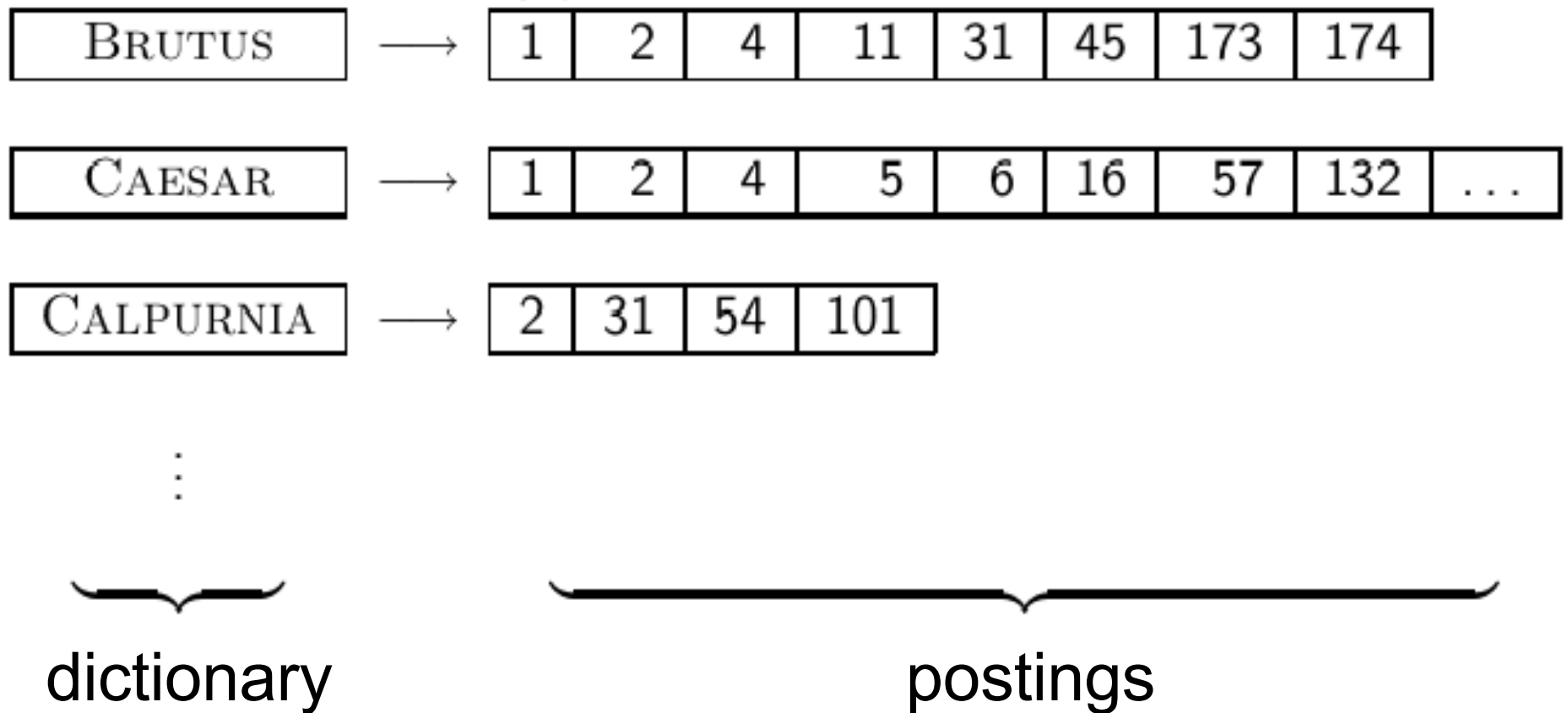
- Esta es una,
- Es una oracion

Ejercicio: Qué hace Google ?

- Stop words
- Normalization
- Tokenization
- Lowercasing
- Stemming
- Non-latin alphabets
- Umlauts
- Compounds
- Numbers

Cómo almacenar: el índice invertido

Por cada término t , almacenamos todos los documentos que contienen t



Construcción del índice invertido

- 1 Collect the documents to be indexed:

Friends, Romans, countrymen. So let it be with Caesar ...

- 2 Tokenize the text, turning each document into a list of tokens:

Friends Romans countrymen So ...

- 3 Do linguistic preprocessing, producing a list of normalized tokens, which are the indexing terms:

friend roman countryman so ...

- 4 Index the documents that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

Tokenizando y Pre-procesando

Doc 1. I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

Doc 2. So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:



Doc 1. i did enact julius caesar i was killed i' the capitol brutus killed me

Doc 2. so let it be with caesar the noble brutus hath told you caesar was ambitious

Generar posting

	term	docID
Doc 1. i did enact julius caesar i was killed i' the capitol brutus killed me Doc 2. so let it be with caesar the noble brutus hath told you caesar was ambitious	i	1
	did	1
	enact	1
	julius	1
	caesar	1
	i	1
	was	1
	killed	1
	i'	1
	the	1
	capitol	1
	brutus	1
	killed	1
	me	1
	so	2
	let	2
	it	2
	be	2
	with	2
	caesar	2
	the	2
	noble	2
	brutus	2
	hath	2
	told	2
	you	2
	caesar	2
	was	2
	ambitious	2

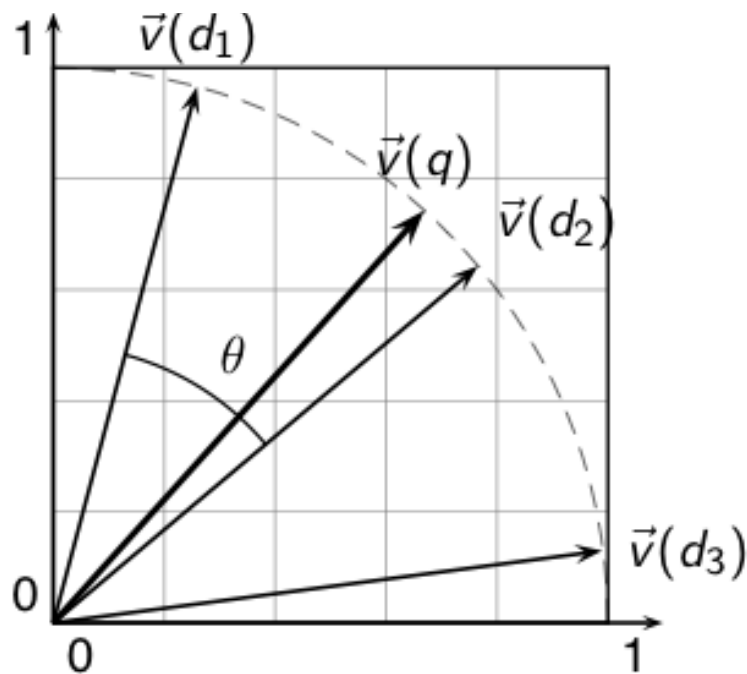
Ordinar postings

term	docID		term	docID
i	1		ambitious	2
did	1		be	2
enact	1		brutus	1
julius	1		brutus	2
caesar	1		capitol	1
i	1		caesar	1
was	1		caesar	2
killed	1		caesar	2
i'	1		did	1
the	1		enact	1
capitol	1		hath	1
brutus	1		i	1
killed	1		i	1
me	1	⇒	i'	1
so	2		it	2
let	2		julius	1
it	2		killed	1
be	2		killed	1
with	2		let	2
caesar	2		me	1
the	2		noble	2
noble	2		so	2
brutus	2		the	1
hath	2		the	2
told	2		told	2
you	2		you	2
caesar	2		was	1
was	2		was	2
ambitious	2		with	2

Determinar frecuencia de documentos

term	docID		term	doc. freq.	→	postings lists
ambitious	2		ambitious	1	→	2
be	2		be	1	→	2
brutus	1		brutus	2	→	1 → 2
brutus	2		capitol	1	→	1
capitol	1		caesar	2	→	1 → 2
caesar	1		did	1	→	1
caesar	2		enact	1	→	1
caesar	2		hath	1	→	2
did	1		i	1	→	1
enact	1		i'	1	→	1
hath	1		it	1	→	2
i	1		julius	1	→	1
i	1		killed	1	→	1
i'	1		let	1	→	2
it	2		me	1	→	1
julius	1		noble	1	→	2
killed	1		so	1	→	2
killed	1		the	2	→	1 → 2
let	2		told	1	→	2
me	1		you	1	→	2
noble	2		was	2	→	1 → 2
so	2		with	1	→	2
the	1					
the	2					
told	2					
you	2					
was	1					
was	2					
with	2					

Tareas: Ranking dada una consulta q



Doc_1 = {w_1, w_2, ..., w_3}



Doc_2 = {w_1, w_2, ..., w_3}

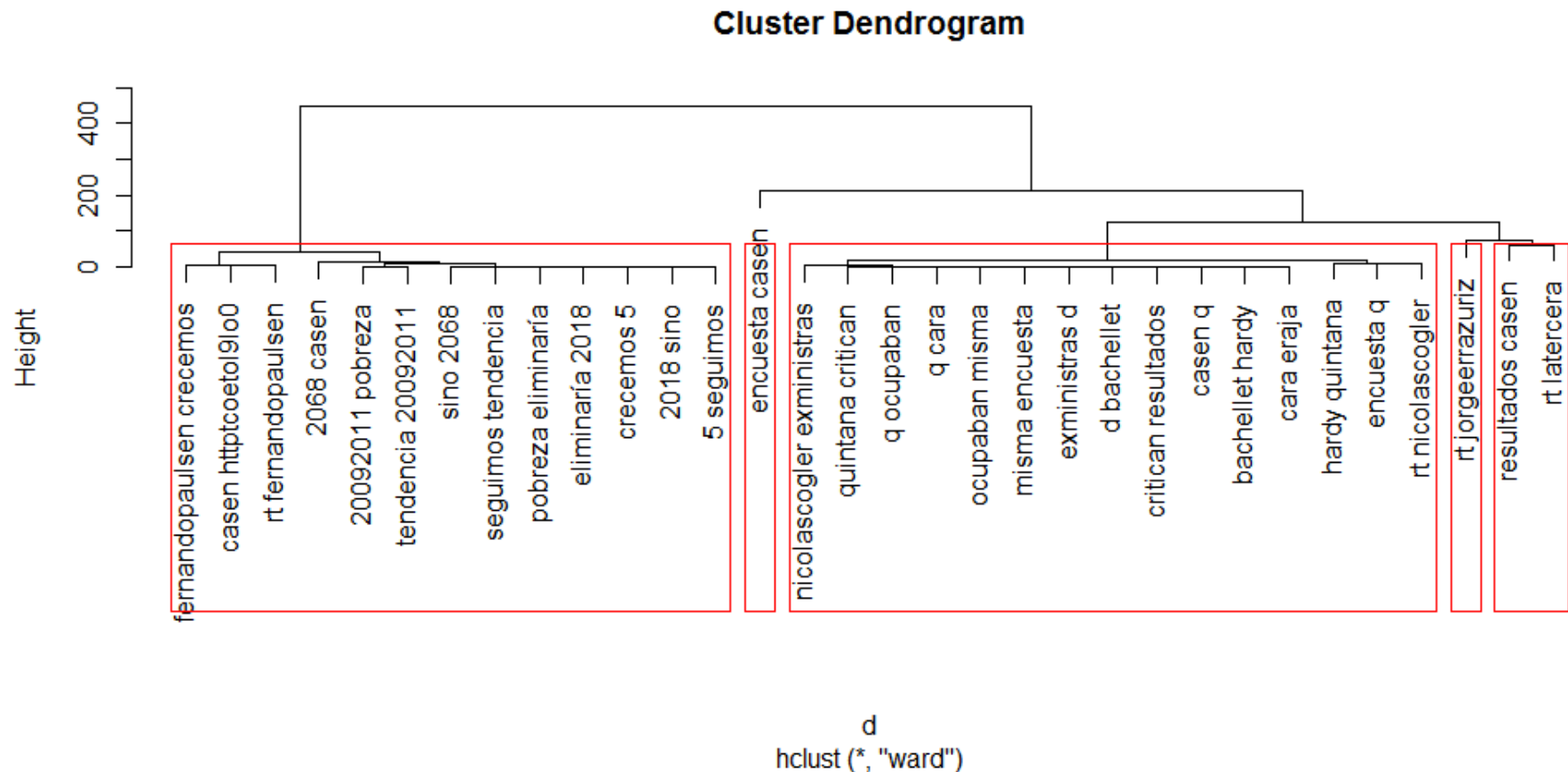


Doc_3 = {w_1, w_2, ..., w_3}

$q = \{w_1, w_2, \dots, w_3\}$ usando TF-IDF

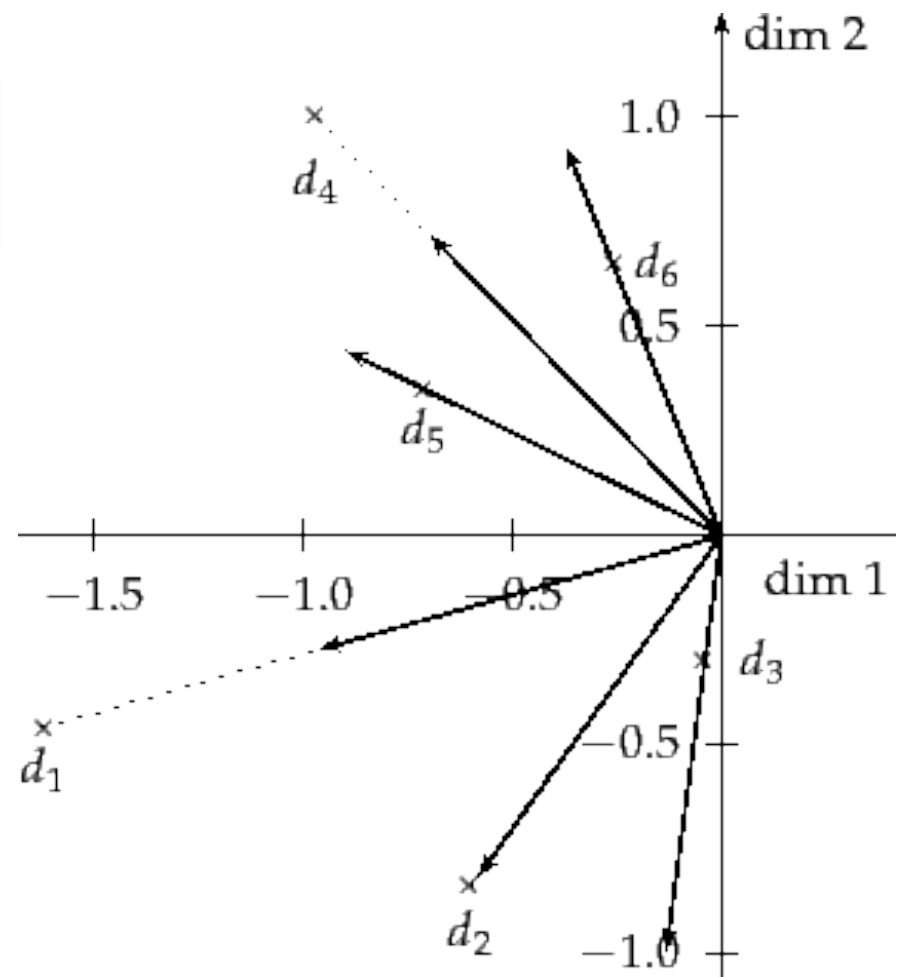
Clustering de texto

- Tweets de la encuesta CASEN 2011



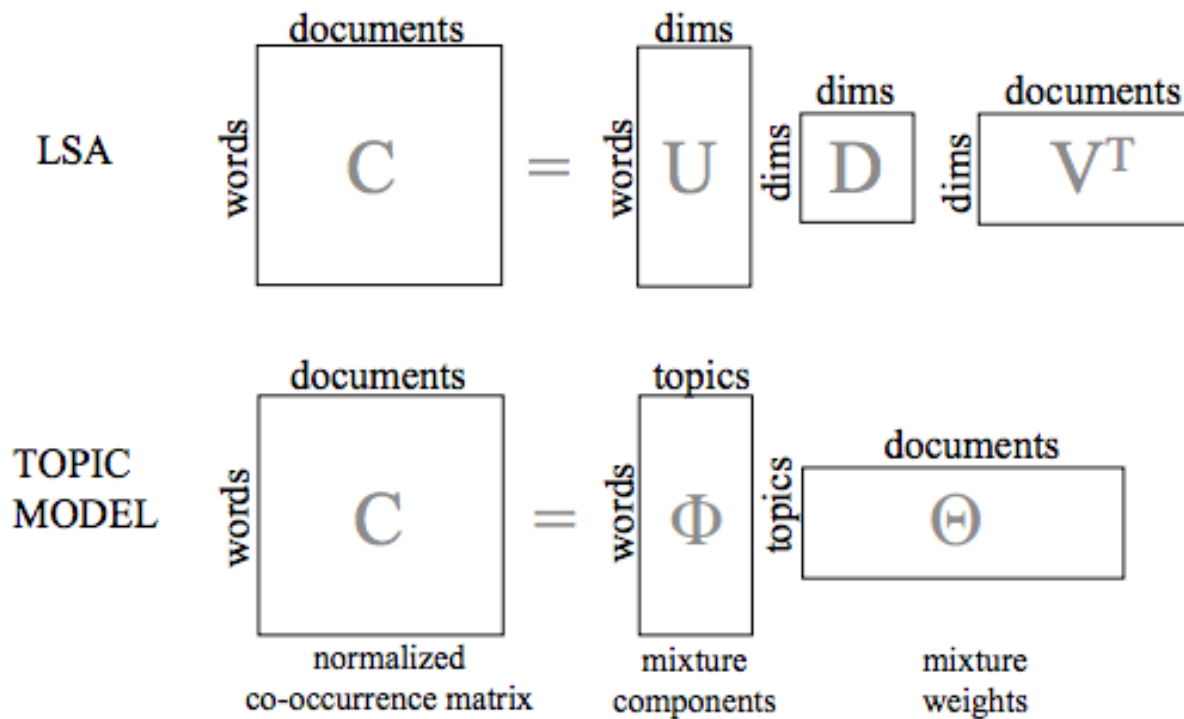
SVD – dejando solo dos dimensiones

		d_1	d_2	d_3	d_4	d_5	d_6	
	1	<u>-0.75</u>	<u>-0.28</u>	<u>-0.20</u>	<u>-0.45</u>	<u>-0.33</u>	<u>-0.12</u>	
	2	<u>-0.29</u>	<u>-0.53</u>	<u>-0.19</u>	0.63	0.22	0.41	
	3	0.28	<u>-0.75</u>	0.45	<u>-0.20</u>	0.12	<u>-0.33</u>	
	4	0.00	0.00	0.58	0.00	<u>-0.58</u>	0.58	
	5	<u>-0.53</u>	0.29	0.63	0.19	0.41	<u>-0.22</u>	



LDA

- Latent Dirichlet Allocation (Blei et al, 2003)
- Relación con LSI



LDA – Processo Generativo e Inferencia

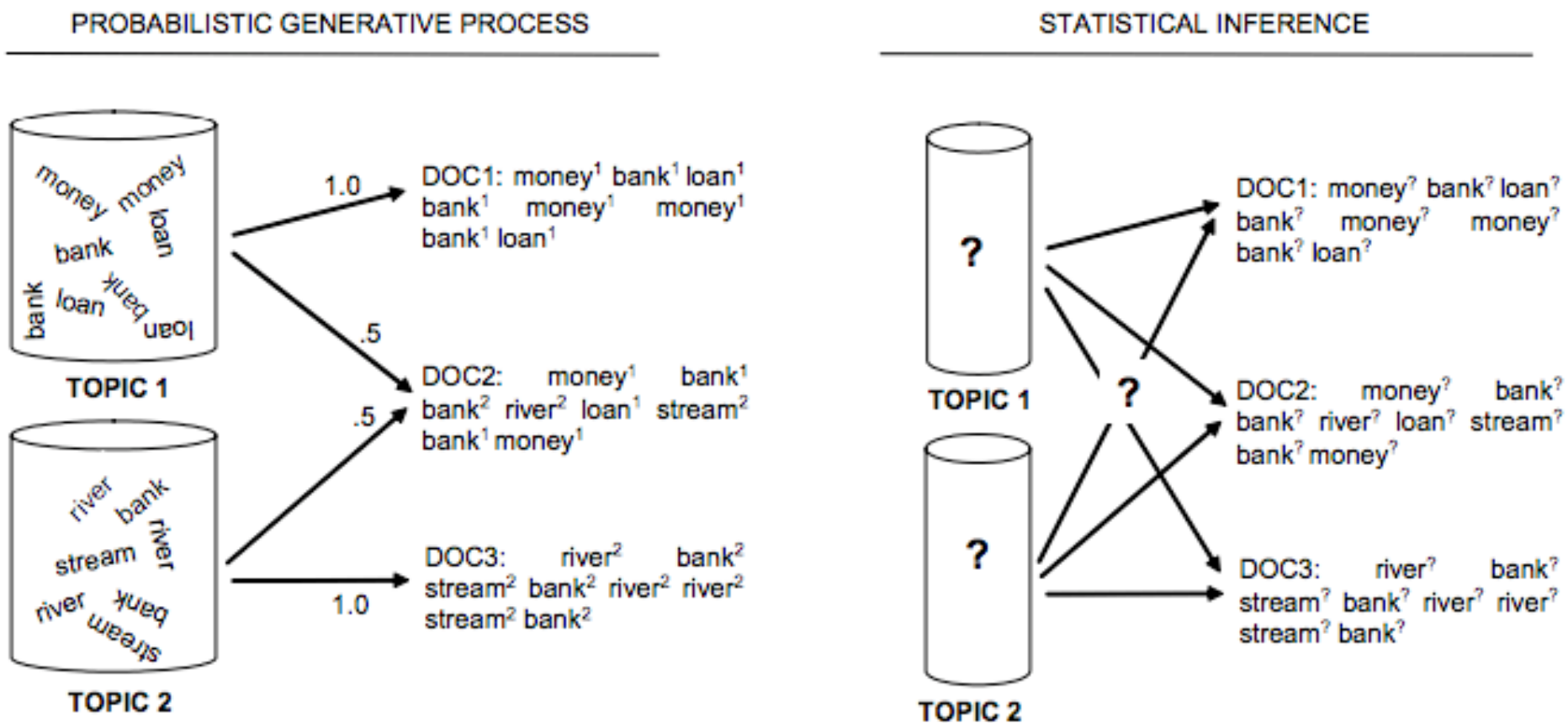
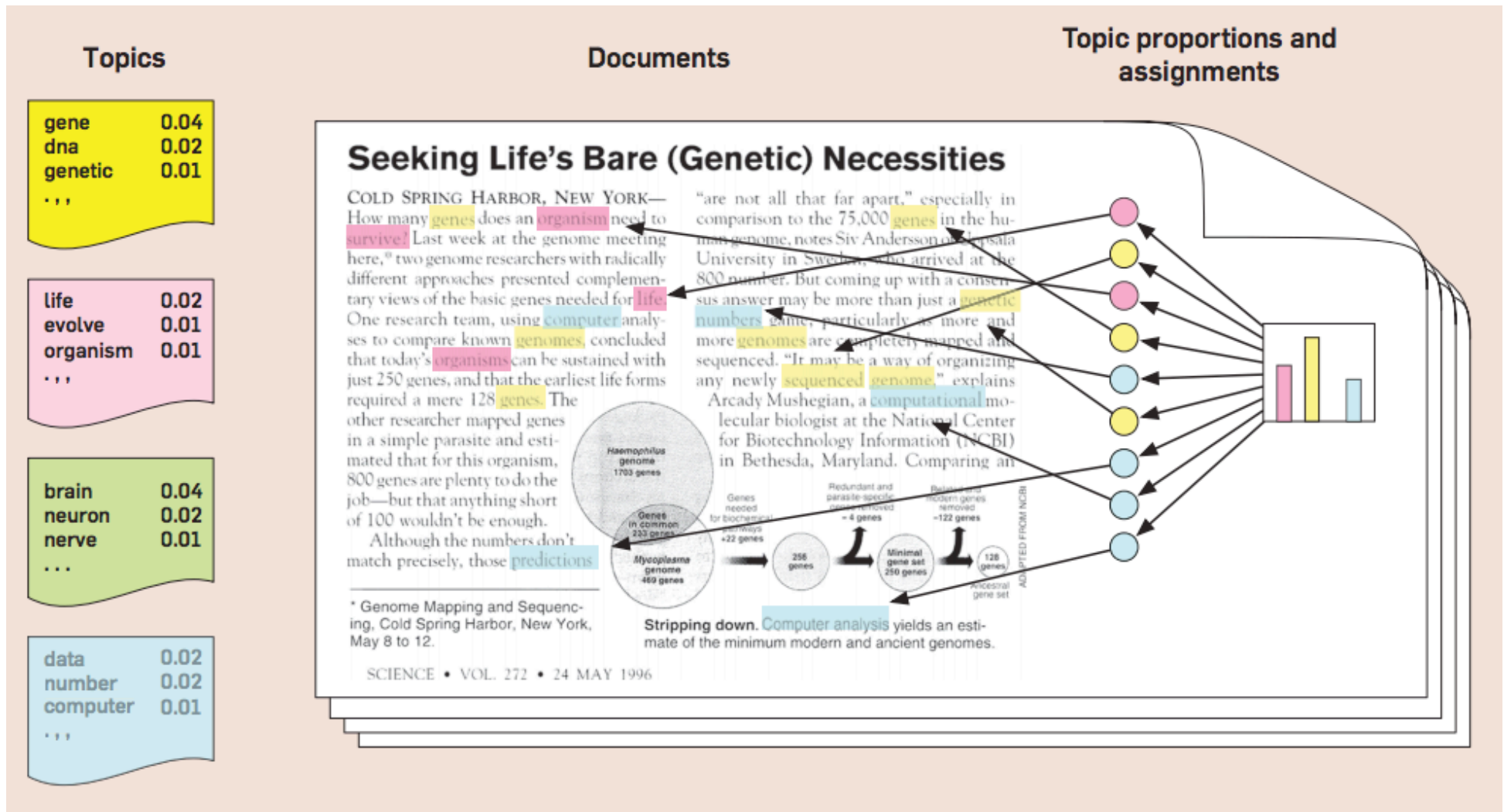
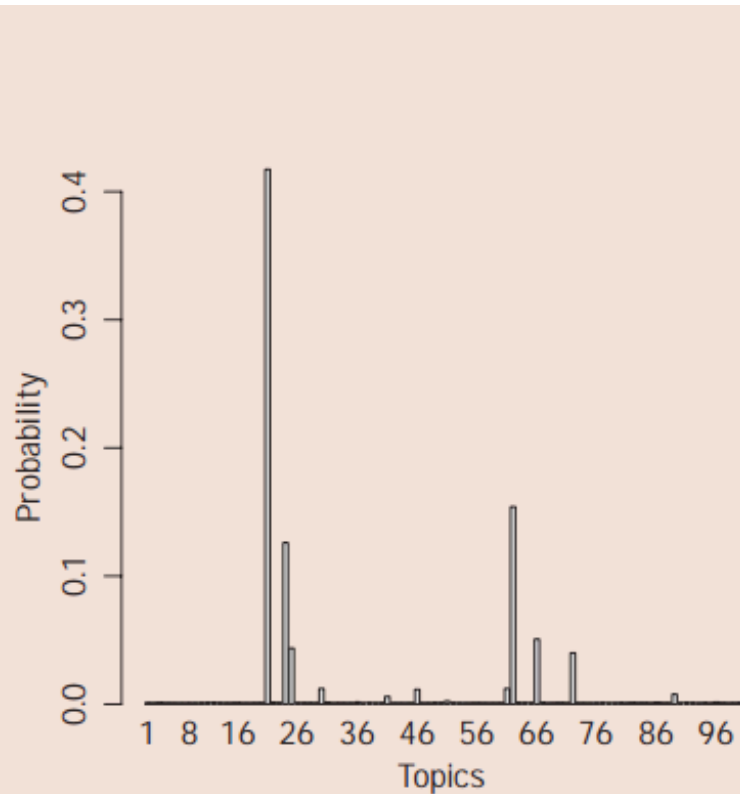


Figure 2. Illustration of the generative process and the problem of statistical inference underlying topic models

LDA – a nivel de Documento



LDA - A nivel de Corpus



“Genetics”

human
genome
dna
genetic
genes
sequence
gene
molecular
sequencing
map
information
genetics
mapping
project
sequences

“Evolution”

evolution
evolutionary
species
organisms
life
origin
biology
groups
phylogenetic
living
diversity
group
new
two
common

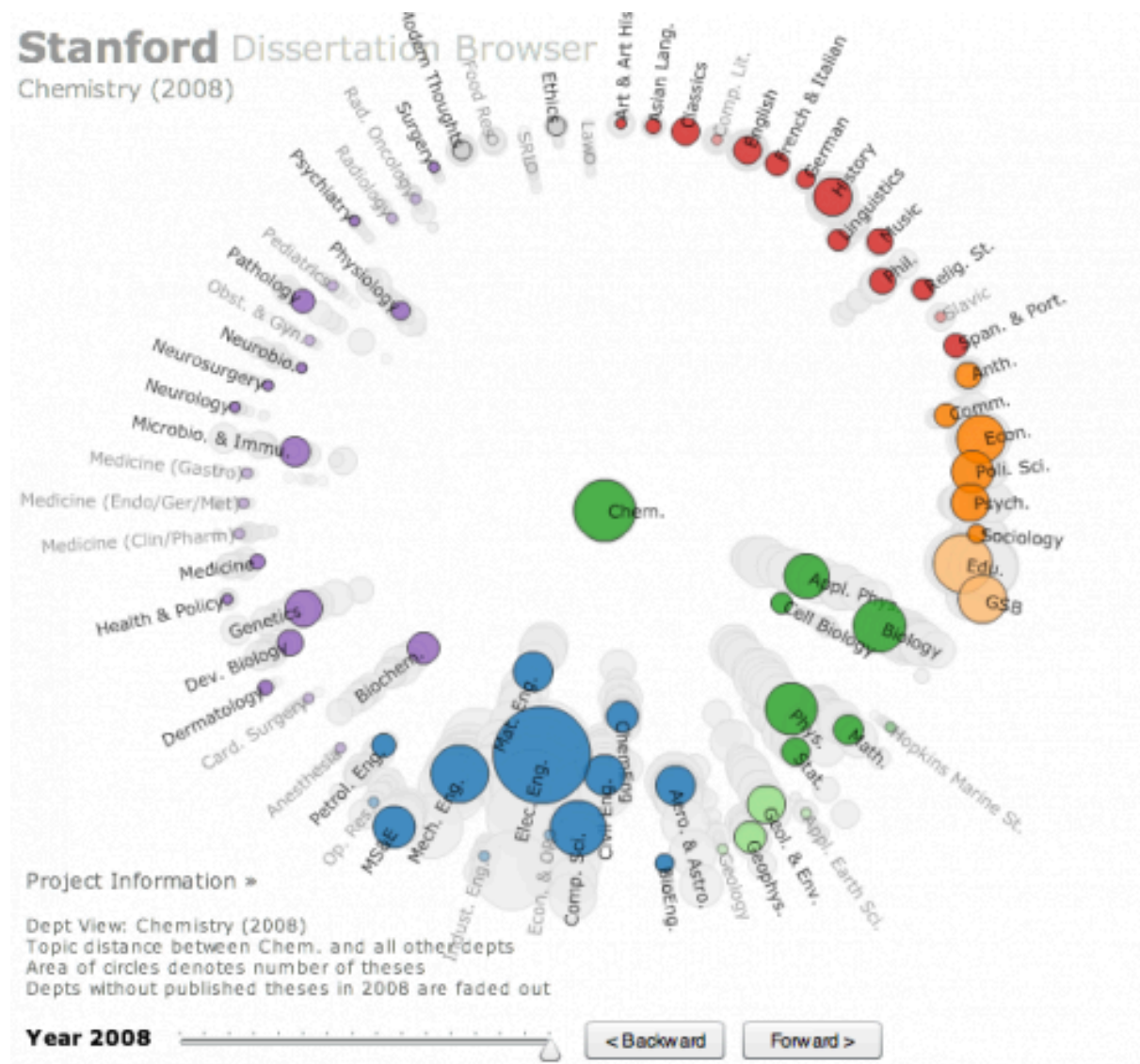
“Disease”

disease
host
bacteria
diseases
resistance
bacterial
new
strains
control
infectious
malaria
parasite
parasites
united
tuberculosis

“Computers”

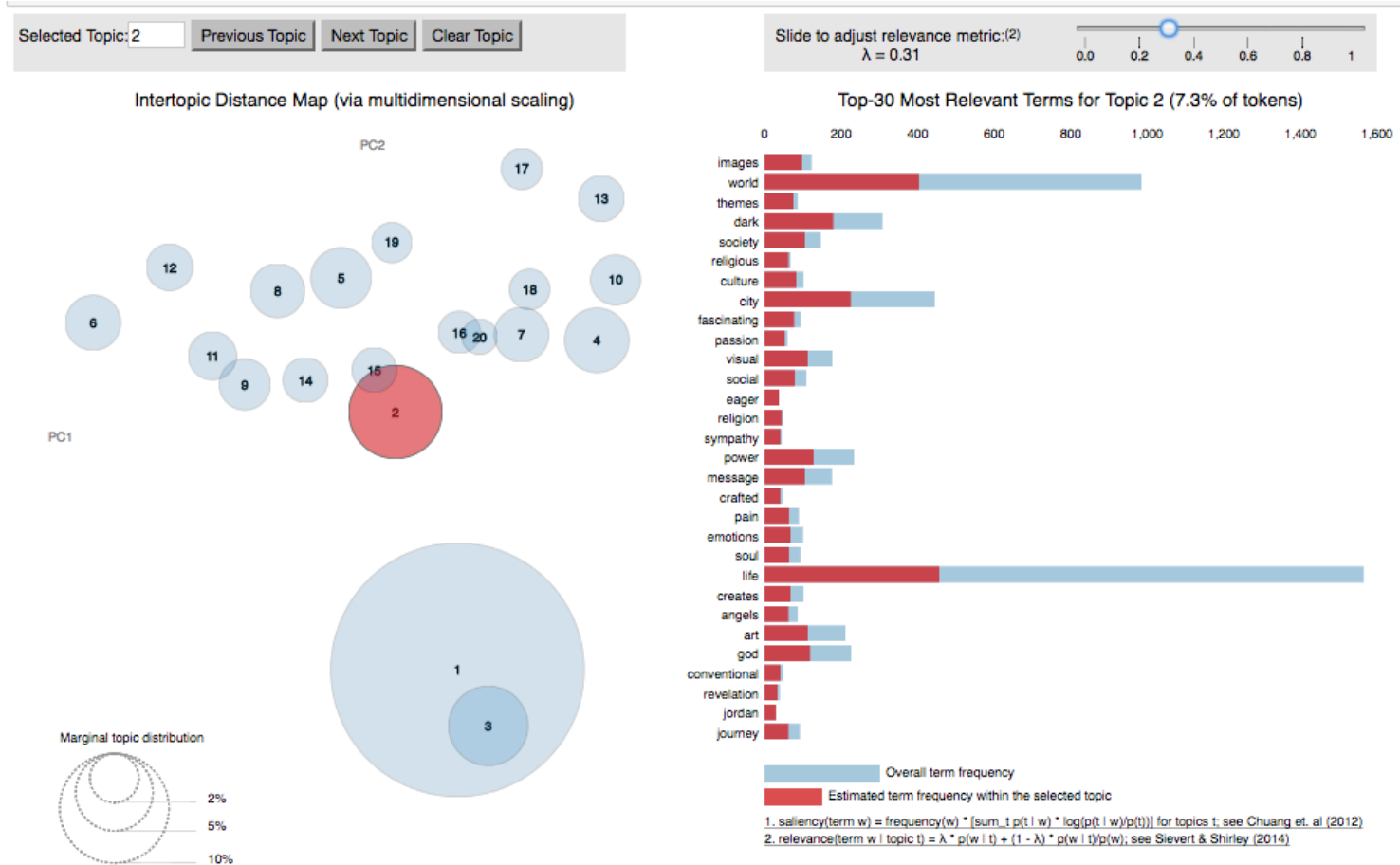
computer
models
information
data
computers
system
network
systems
model
parallel
methods
networks
software
new
simulations

Stanford dissertation browser



Interactivo: pyLDAvis

Out[12]:



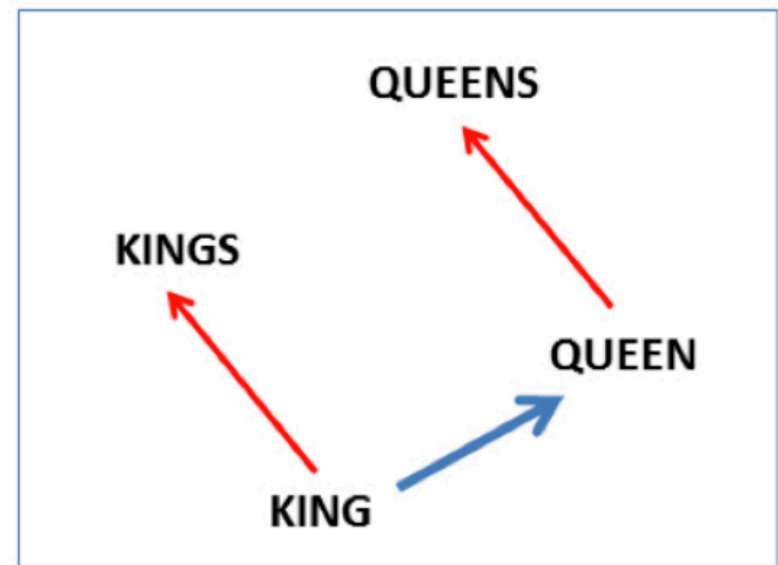
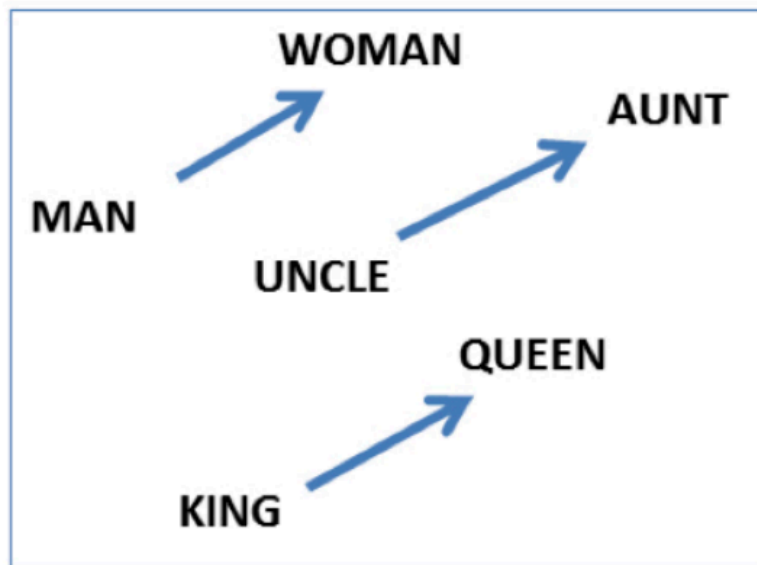
http://nbviewer.jupyter.org/github/bmabey/pyLDAvis/blob/master/notebooks/pyLDAvis_overview.ipynb

Otro modelo de lenguaje: Word Embeddings

Word2vec Embeddings (Mikolov, 2010)

$\text{vector}('king') - \text{vector}('man') + \text{vector}('woman') \approx \text{vector}('queen')$

$\text{vector}('Paris') - \text{vector}('France') + \text{vector}('Italy') \approx \text{vector}('Rome')$



¿Cómo obtener vectores de palabras?

- Modelo word2vec: Skip-gram o continuous bag-of-words (CBoW)

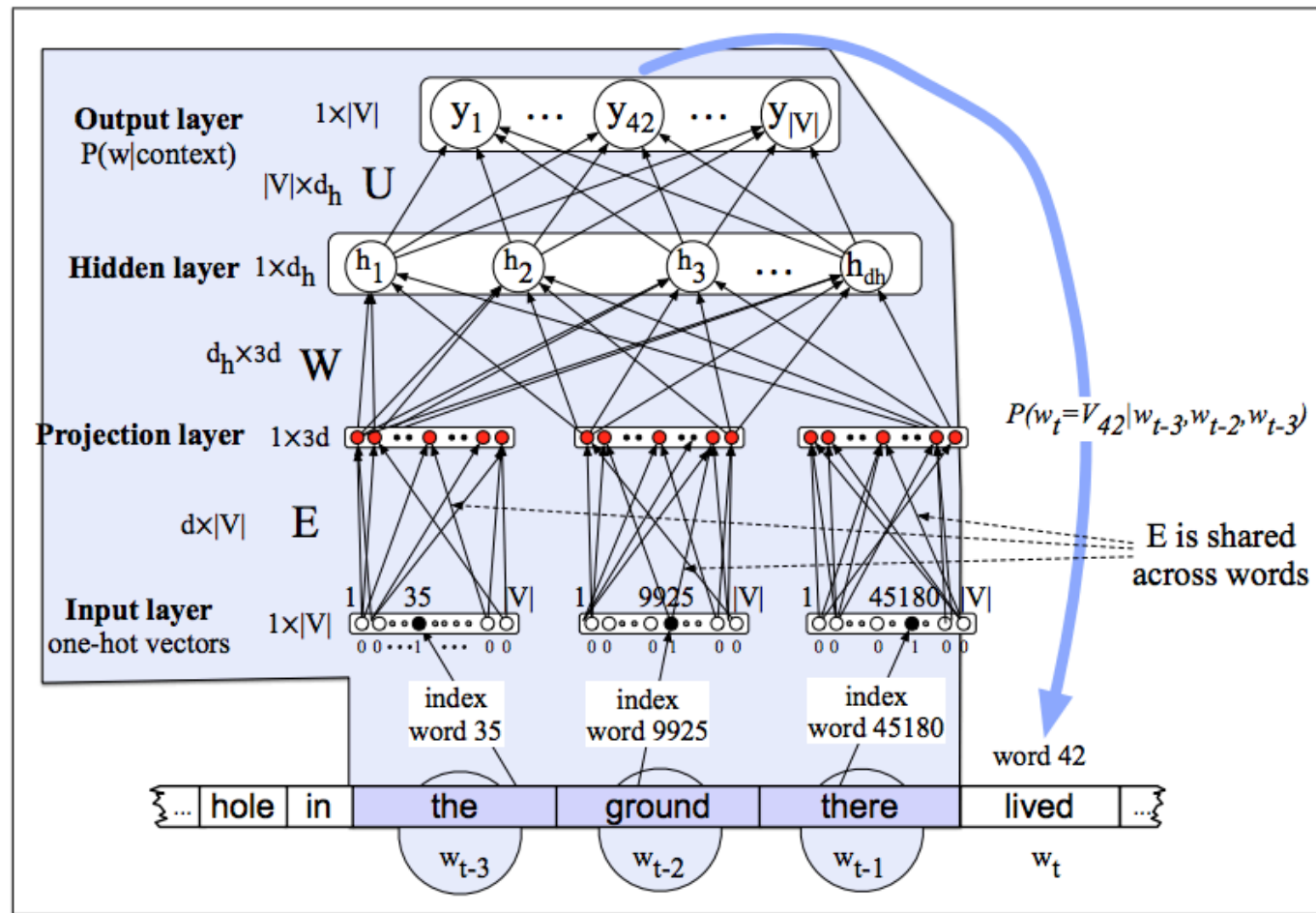


Figure 7.13 Learning all the way back to embeddings. Notice that the embedding matrix E is shared among the 3 context words.

Visualización de Texto, Ejemplos

- Nubes de Palabras (Wordle)
- ChronoText
- Revisionist
- WordTree

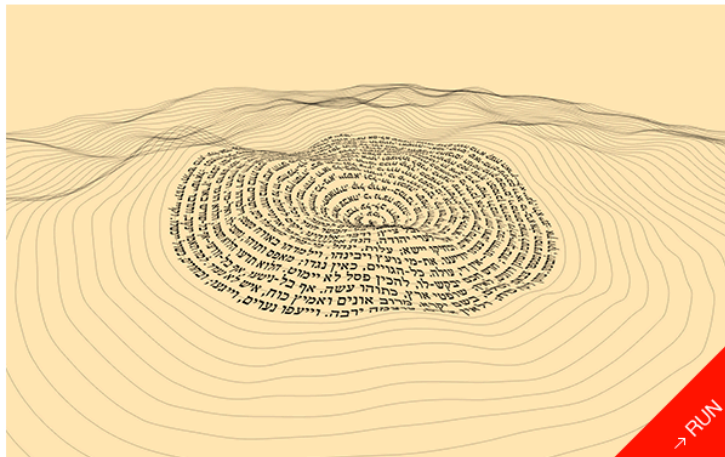
Nube de palabras

●Tweets de la encuesta CASEN 2011



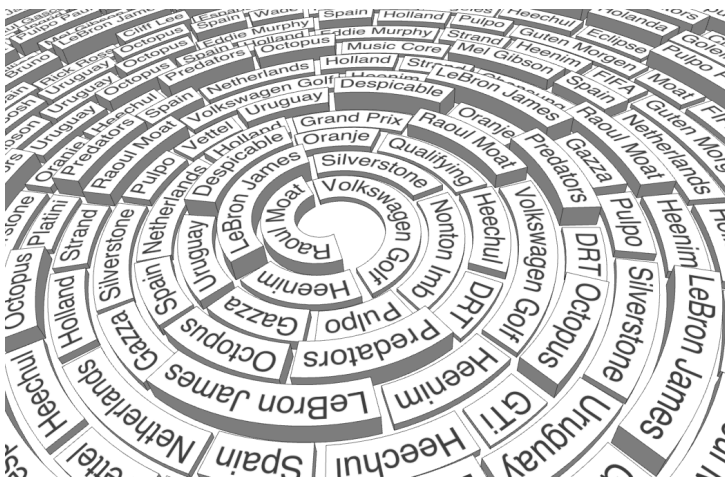
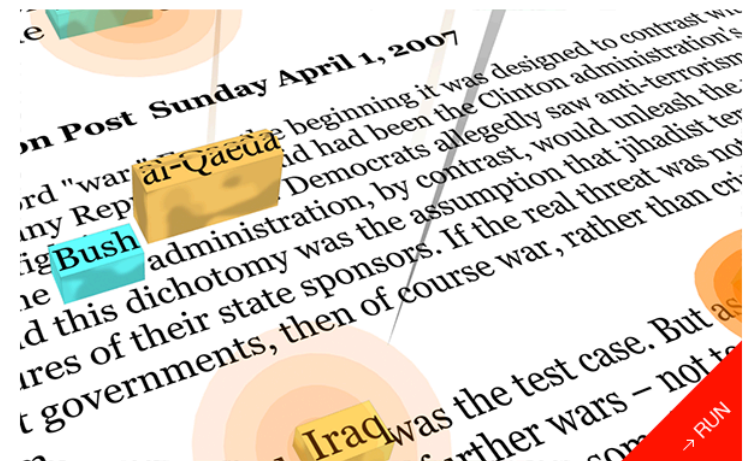
ChronoText

TOPOGRAPHIC TEXT



THE WAR OF THE WORDS

Two armies clash on the semantic battlefield. Each camp is composed of groups of elevated keywords. Each group is firing at one or more groups from the opposite camp. Each time a keyword is hit, its elevation decrease.

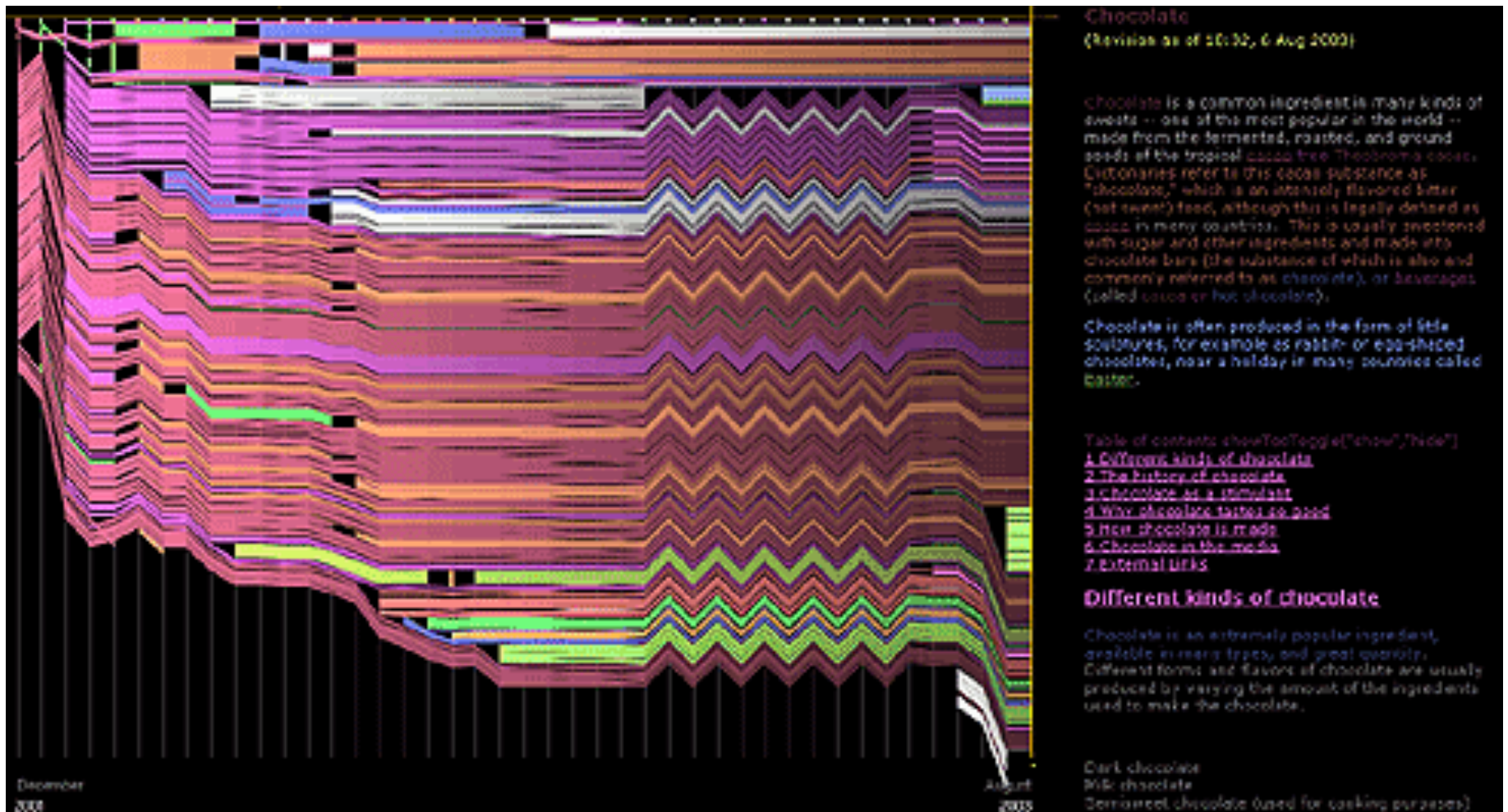


More than two thousand years earlier, another poem was composed: **the song of the Sea**, in the Biblical Book of Exodus, celebrating Pharaoh's defeat in the Sea of Reeds. Quoting Wikipedia:

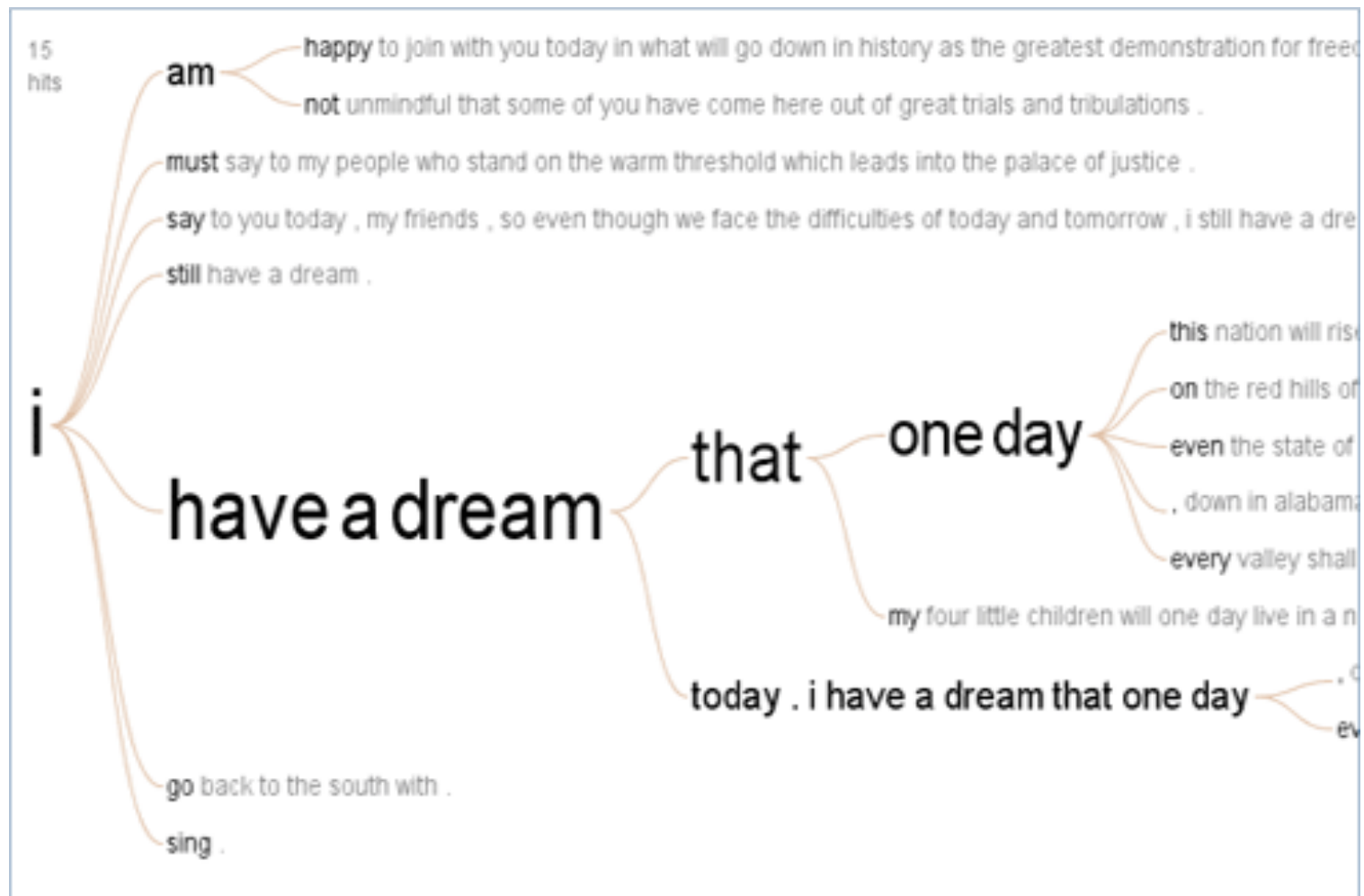
Revisionist

[illegible]

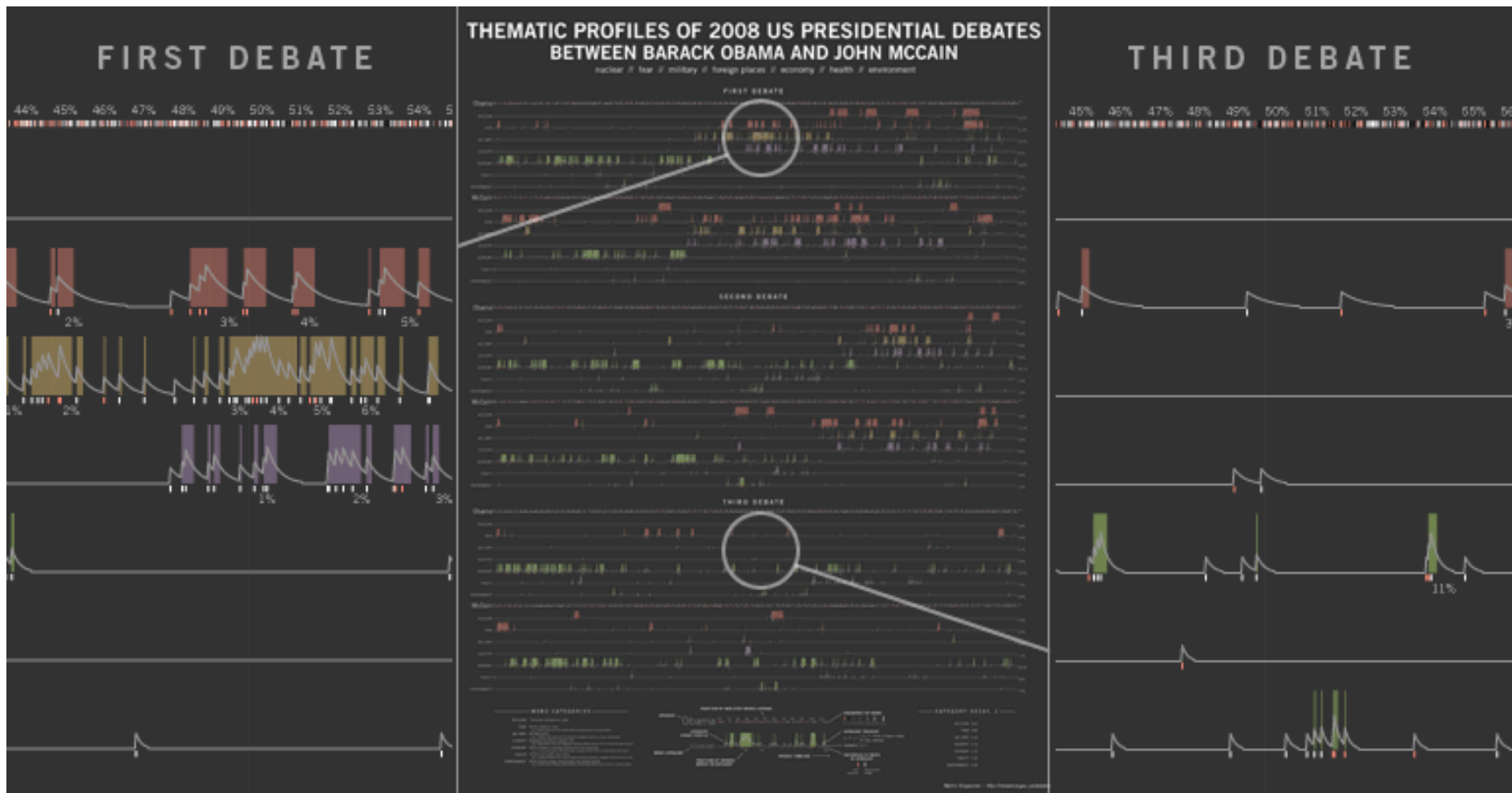
History Flow



WordTree



Análisis Léxico de discurso presidencial



<http://mkweb.bcgsc.ca/debates/>

Consultas: dparra@ing.puc.cl