Introducción a Text Mining y a Natural Language Processing

Analizando texto con Machine Learning

German Rosati
german.rosati@gmail.com

UNTREF - UNSAM - CONICET

21 de noviembre de 2019

Introducción

¿Qué es NLP/ Text Mining?

- Conjunto de algoritmos de machine learning que se usan sobre datos de texto, no estructurados
- Área de investigación científica llamada Natural Language Processing, una subdisciplina de machine learning/ciencias de la computación que trata de emular la interpretación humana de textos.

Introducción

¿Qué es NLP/ Text Mining?

- Datos no estructurados de texto
- Parte crítica del pipeline: preprocesamiento
- Principales técnicas son Parsing, tokenización, sentence splitting, lemmatización, y part of speech tagging.
- Cuanto más se cuida el proceso de preprocesamiento, mejor calidad tienen las técnicas de análisis de texto.

Preprocesamiento de texto

¿Qué es un texto?

- Se puede pensar un texto como una secuencia de
  - caracteres
  - palabras
  - frases y entidades con sentido
  - oraciones
  - párrafos
  - documentos
  - ...

Preprocesamiento de texto

¿Qué es un texto?

- Parece natural pensar en un texto como una secuencia de palabras
- A su vez, una palabra es una secuencia **significativa** de caracteres
- En castellano, podemos delimitar una palabra a través de espacios y/o signos de puntación. En otros idiomas... no es tan simple.

Preprocesamiento de texto

Detección de oraciones

- Recorte de una secuencia de caracteres entre dos signos de puntuación. El signo debe estar acompañado por un espacio en blanco.
- Para determinar las abreviaciones en el texto se utiliza un diccionario específico para cada idioma (ejemplo Sr.)

- INPUT:

  El conjunto de estas relaciones de producción forma la estructura económica de la sociedad, la base real sobre la que se levanta la superestructura jurídica y política y a la que corresponden determinadas formas de conciencia social. El modo de producción de la vida material condiciona el proceso de la vida social política y espiritual en general. No es la conciencia del hombre la que determina su ser sino, por el contrario, el ser social es lo que determina su conciencia.

- OUTPUT:

  [El conjunto de estas relaciones de producción forma la estructura económica de la sociedad, la base real sobre la que se levanta la superestructura jurídica y política y a la que corresponden determinadas formas de conciencia social.]

  [El modo de producción de la vida material condiciona el proceso de la vida social política y espiritual en general.]

  [No es la conciencia del hombre la que determina su ser sino, por el contrario, el ser social es lo que determina su conciencia.]

Preprocesamiento de texto

Tokenizacion

- Proceso que divide una secuencia (por ejemplo,una oración) en los llamados *tokens*
- Un *token* puede ser pensada como una unidad útil para el procesamiento semántico
- Dependiendo del caso, podrán ser palabras, oraciones, párrafos, documentos, etc.
- Una forma de identificar *tokens* en idiomas modernos que utilizan un sistema de escritura occidental se realiza delimitando espacios en blanco con límites de palabra, entre comillas, paréntesis y puntuación.

Preprocesamiento de texto

Tokenizacion

- INPUT:

  No es la conciencia (...) la que determina su ser sino (...) el ser social lo que determina su conciencia.

- OUTPUT:

  [No], [es], [la], [conciencia], [la], [que], [determina], [su], [ser], [sino], [el], [ser], [social], [lo], [que], [determina], [su], [conciencia]

Preprocesamiento de texto

Tokenizacion - Problemas

- ¿Qué hacemos con
  1. ... números?
  2. ... signos de puntuación o siglas?
     - ¿RR.HH. es 1 token o 2?
  3. ... palabras similares con significados diferentes?
  4. ... abreviaturas?
  5. conceptos compuestos de varias palabras?
     - ¿Buenos Aires es 1 token o 2?
     - ¿estado del arte es 1 token o 3?

- En general, es útil poder reducir al mismo token diferentes formas de una palabra:
  - lobo, lobos => lob
  - compró, comprará => compr
- Dos formas de Normalización
  1. **Stemming**
     - Remueve y reemplaza sufijos para llegar a la forma raíz"de la palabra, generalmente, llamada *stem*
     - Generalmente, consta de reglas y heurísticas que sirven para truncar los sufijos.
  2. **Lemmatización**
     - Usa vocabularios y análisis morfológico.
     - Devuelve la forma base o de "diccionario"de una palabra, llamada *lemma*

Preprocesamiento de texto

Part-Of-Speech POS-Tagging

- Etiquetado de las palabras según el rol que cumplen dentro de una oración.
- Asigna a cada una de las palabras de un texto su categoría gramatical (sustantivo, adjetivo, adverbio, etc.)
- Requisito: establecer relaciones de una palabra con sus adyacentes dentro de una frase o de un párrafo.
- Un mismo token puede tener múltiples etiquetas POS, pero solo una es válida dependiendo del contexto.

Preprocesamiento de texto

Stopwords

- Exclusión de palabras muy comunes con poco valor para recuperar información del documento o corpus
- La cantidad de ocurrencias de una palabra en el texto determina si es o no una "stop word"
  - cuanto más ocurrencias existan menos relevancia tiene en el texto.
- Artículos, pronombres, preposiciones, y conjunciones.
- Reducir el tamaño del texto para analizar, eliminando aproximadamente el 30 % o 40 % de dichas palabras.

Vectorización de texto

Bag of Words

- ¿Cómo pasar del texto no estructurado a un formato que pueda ser procesado por un algoritmo o técnica de ML?
- Es necesario transformar los *tokens* en $X's$, en features.
- Representar un documento en alguna forma de espacio vectorial.
- Una forma de representación posible es el modelo **Bag of Words** o BOW.
- NO es la única, ni necesariamente la mejor...

Vectorización de texto

Construcción de la matriz

	good	movie	not	a	did	like
good movie	1	1	0	0	0	0
not a good movie	1	1	1	1	0	0
did not like	0	0	1	0	1	1

- BoW genera una representación de cada documento, en función de las palabras que este contiene.
- Algunas características:
  1. Es simple de generar
  2. Se asume que las palabras son ïndependientes"
  3. Los vectores son claramente no independientes
  4. La gramática y el orden de las palabras se pierden

Vectorización de texto

n-grams

- ...

	good movie	movie	did not	a	...
good movie	1	1	0	0	...
not a good movie	1	1	0	1	...
did not like	0	0	1	0	...

- Podemos intentar preservar algún orden las palabras, al menos, local
- **n-Grams**[2]
  1. Pares (*2-grams*), tripletas (*2-grams*), cuatrifectas (*4-grams*) de palabras
  2. Problema: la cantidad de X, features, crece exponencialmente $O(V^N)$

Vectorización de texto

Ponderando los *n-grams*

- La forma más simple es, como vimos, hacer un conteo de palabras
- En general, filtramos los *n-grams*
  - demasiado frecuentes (caso típico, stopwords, pero no solamente)
  - los demasiado poco frecuentes, porque probablemente se produzca alguna forma de overfitting
- Nos quedamos, entonces, con los *n-grams* de frecuencia media
- Aún así, este conteo de los *n-grams* no está normalizado
- Es importante NORMALIZAR...

Vectorización de texto

Ponderando los *n-grams*

- Podemos pensar en dos dimensiones de las frecuencias de los términos de un corpus...
  1. Un término t es más **importante** si es más frecuente en un documento d de un corpus C determinado.
  2. A su vez, t es más **informativo** del contenido de un documento d si está presente en pocos documentos y no en todos de C.
- Es decir, hay que mirar tanto la frecuencia de t a lo largo de todo el corpus C y al interior del documento d.
- Dos métricas para cuantificar ambas dimensiones... [3]

Vectorización de texto

Term Frequency -TF-

- $c(t, d)$ es el conteo çrudo"del t en el documento d
- $raw\_tf(t, d) = c(t, d)$
- Hasta aquí estamos en el esquema BoW crudo.
- Problemas:
  1. El largo de los documentos suele ser variable
  2. En general, la información acerca del sentido no çrece"de forma proporcional a la ocurrencia de t en un d
- Entonces, hay normalizaciones alternativas
  1. Binaria: $0, 1$
  2. $TF(t, d) = \frac{c(t,d)}{\sum_{t \in d} c(t,d)}$
  3. Log: $log\_tf = 1 + \log c(t, d)$

Vectorización de texto

Document Frequency -IDF-

- Insumo para una medida de la **informatividad** de un término a lo largo de C.
- $DF(t) = \log \frac{df(t)}{|C|}$
  donde
  - $df(t)$ es la cantidad de documentos en C que contienen el término t
  - $|C|$ es el tamaño del corpus C, es decir, el total de documentos en C
- Cuanto mayor es $DF(t)$ menor es la informatividad de un término. Entonces, se calcula su inversa (IDF):
- $IDF(t) = \log \frac{|C|}{df(t)}$

Vectorización de texto

Term Frequency-Inverse Document Frequency TF-IDF

- Entonces, $tf(t, d)$ es una propiedad del documento y $IDF(t)$ es una propiedad del corpus
- Combinamos ambas en una medida llamada Term Frequency-Inverse Document Frequency (TF-IDF)
- $TF\_IDF(t) = tf(t, d) \times IDF(t)$
- Valores altos de $tf(t, d)$ y valores altos de $IDF(t)$ -o sea, valores bajos de $DF(t)$ arrojan valores altos de $TF\_IDF(t)$.
- O sea, términos t frecuentes en d y poco frecuentes en C.

Vectorización de texto

Term Frequency-Inverse Document Frequency TF-IDF

	good movie	movie	did not	...
good movie	**0.17**	**0.17**	0	...
not a good movie	**0.17**	**0.17**	0	...
did not like	0	0	**0.47**	...

- Tenemos un BoW mejorado: en lugar de usar los conteos crudos de cada *n-gram*, usamos los pesos calculados con $TF\_IDF(t)$
- Normalizamos los resultados para cada fila usando la norma ℓ_2 o la norma ℓ_1

Ejercicio

Clasificador Automático para un Call Center

- Todos forman parte del equipo de Data Science del Gobierno de la Ciudad y desde la Defensoría del Pueblo les llega un requerimiento:

  "Tenemos un problema: tenemos muchas llamadas al call center y los operadores no tienen el tiempo suficiente para clasificarlas todas. ¿No se pueden armar algo con Data Science para ayudarnos?"

Topic Modelling

Generalidades

- Algoritmos para descubrir los temas que permean un colección de documentos.
- Pueden servir para organizar los corpus textuales a partir de los temas descubiertos
- Puede ser aplicados a conjuntos masivos de documentos (escalabilidad)
- palabras que pertenecen a un mismo tópico co-ocurren
- Utilizados para otro tipos de datos no textuales (datos genéticos, imágenes, redes sociales)**??**

- Coocurrencia:
  - palabras que pertenecen a un mismo tópico co-ocurren
  - no así las palabras que tienen que ver con distintos tópicos.
  - Probabilidad mayor de que “fútbol” y “pelota”, o “ciencias” y “medicina” sean palabras co-ocurrentes
  - Tópicos similares.

Topic Modelling

Latent Dirichlet Allocation -LDA-

Vamos a ver un modelo para tópicos (de los más usados) [1] Algunos supuestos clave

- Un documento se compone de muchos tópicos
- Cada documento se produce a partir de un procedimiento generador (enseguida lo vemos...)
- Un tópico está compuesto de palabras, específicamente, es una distribución de probabilidad a lo largo de un vocabulario
- Los tópicos preexisten a los documentos
- Hay que especificar una cantidad de tópicos previamente

Proceso generativo de un documento

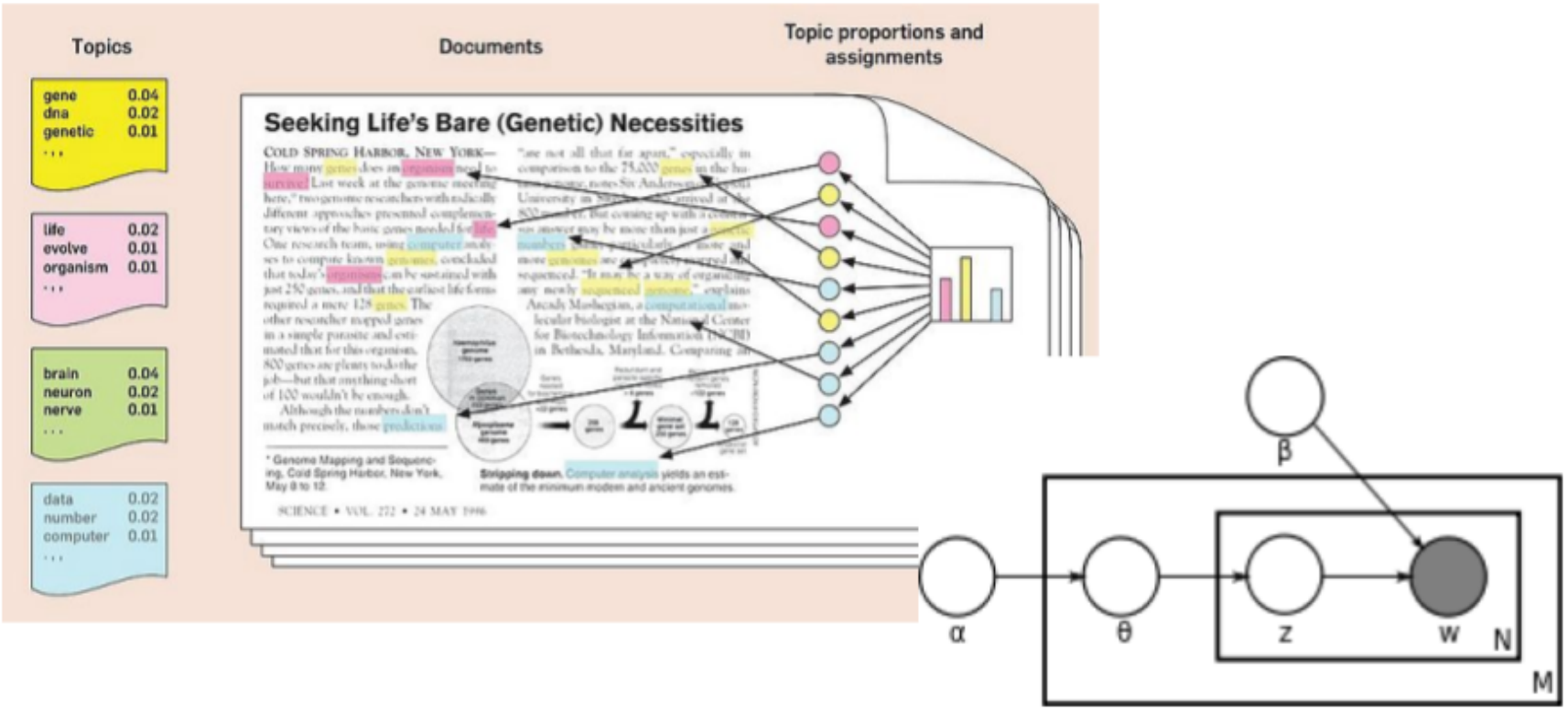
1. Se elige aleatoriamente una distribución a lo largo de tópicos. O sea, el documento d tiene 20 % del T1, 0 % del T2, ..., 80 % del Tn
2. Para cada palabra (w) del documento d
   1. seleccionar aleatoriamente un tópico de la distribución de tópicos
   2. seleccionar aleatoriamente una palabra del tópico correspondiente (cada tópico era una distribución sobre palabras)

Proceso generativo de un documento un poco más formamente

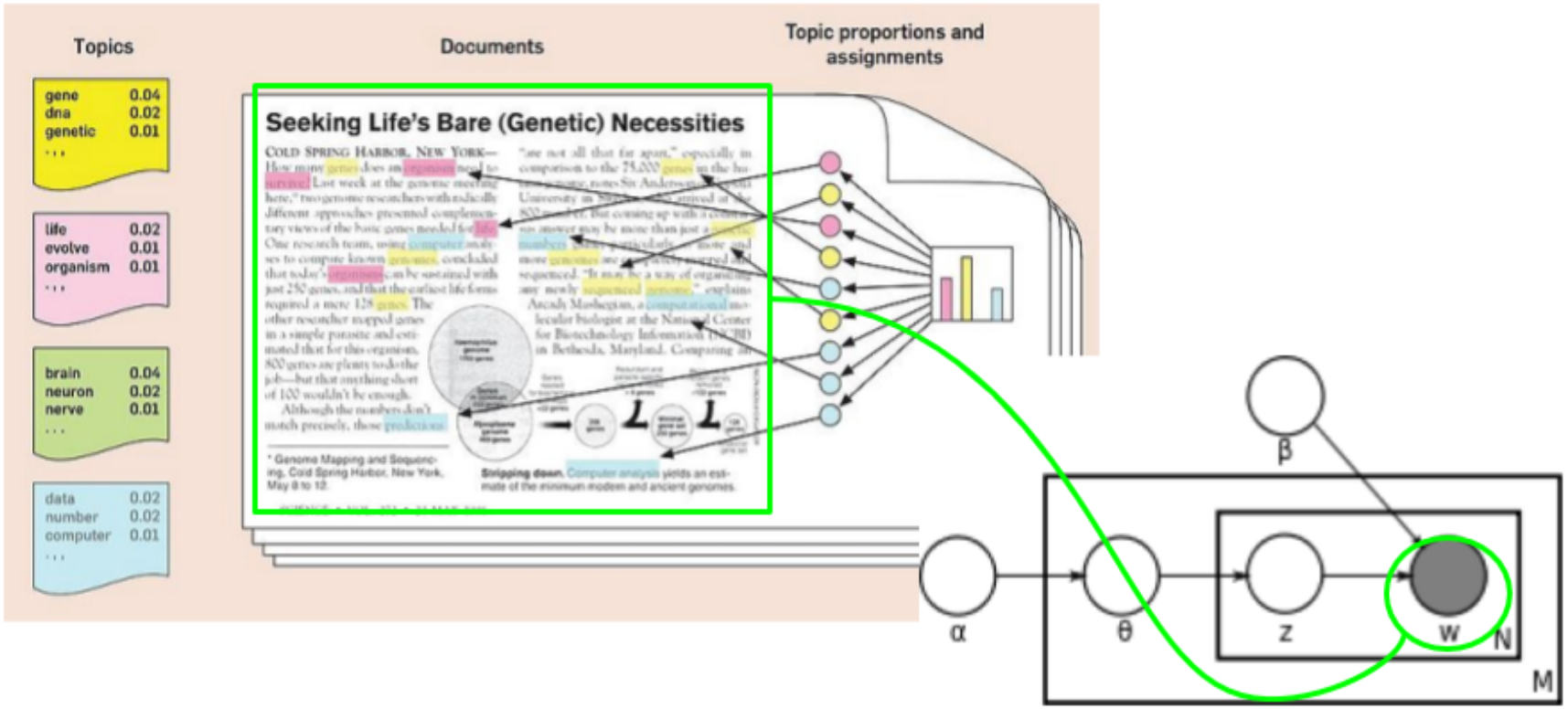
- Alguna notación breve...
  - $\beta_{1:K}$ son los tópicos, donde cada β_k es una distribución sobre el vocabulario
  - θ_d es la proporción de tópicos para el documento d
  - $\theta_{d,k}$ es la proporción del tópico k para el documento d
  - $z_{d,n}$ es la asignación de tópicos para la palabra n en el documento d
  - w_d son las palabras observadas en el documento d
- El objetivo, entonces, es estimar la siguiente probabilidad condicional:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \quad (1)$$

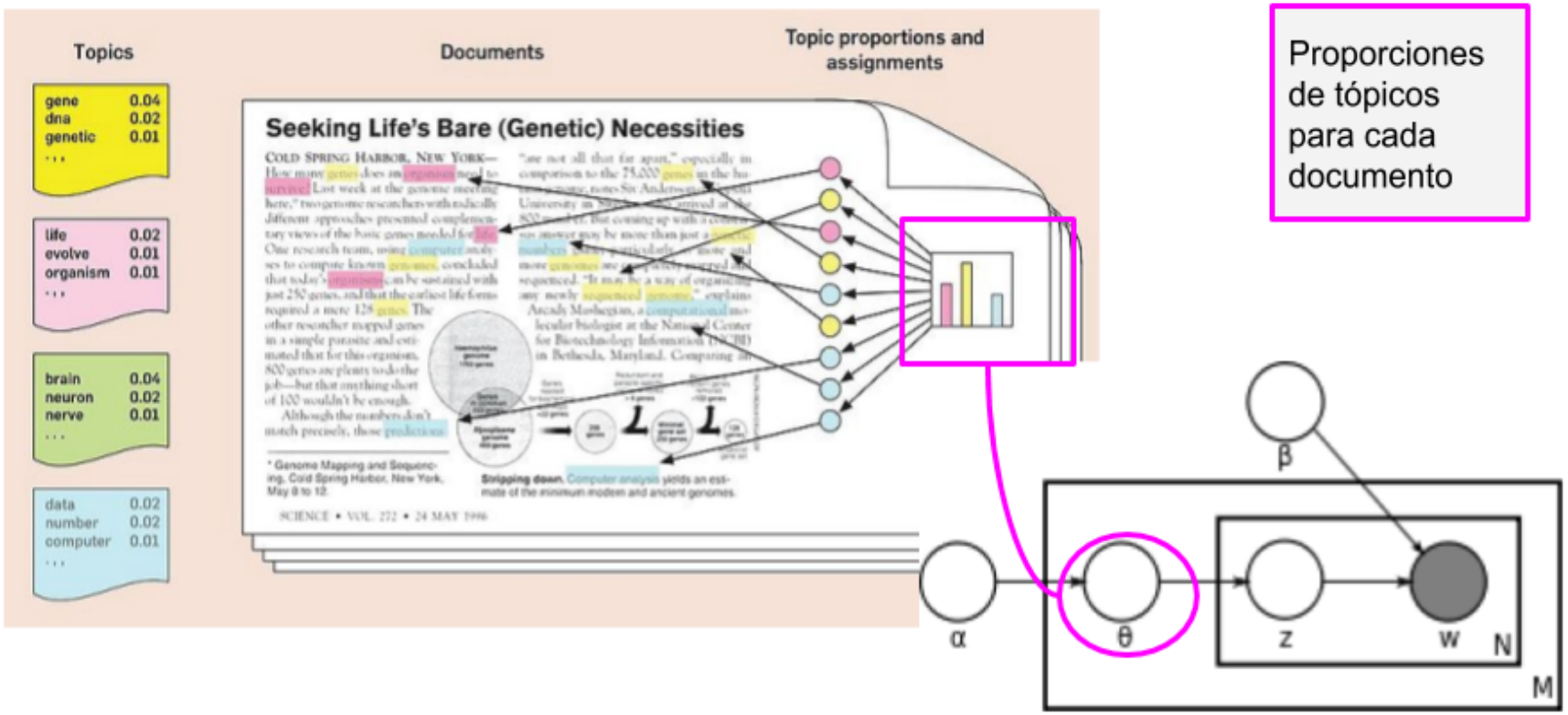
Gráficamente...



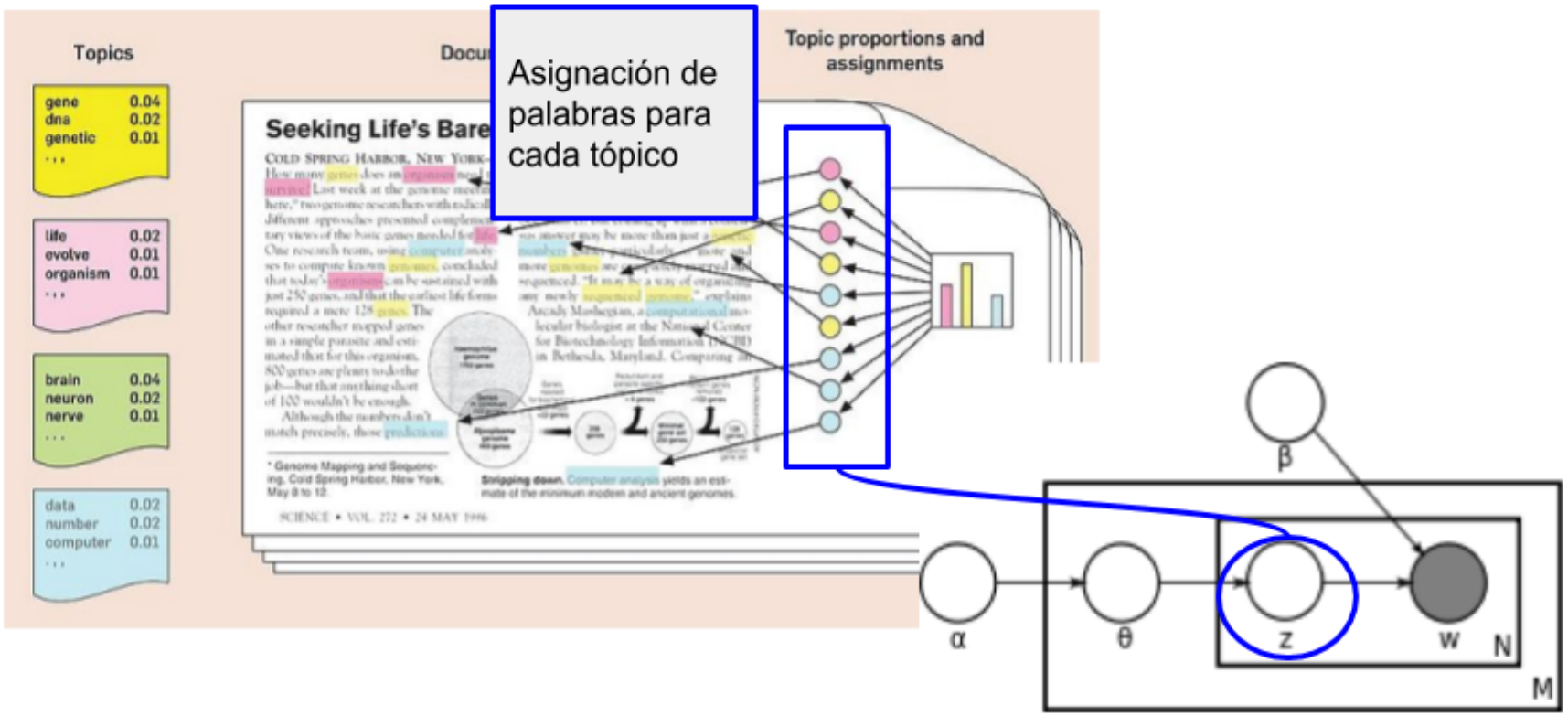
Gráficamente...



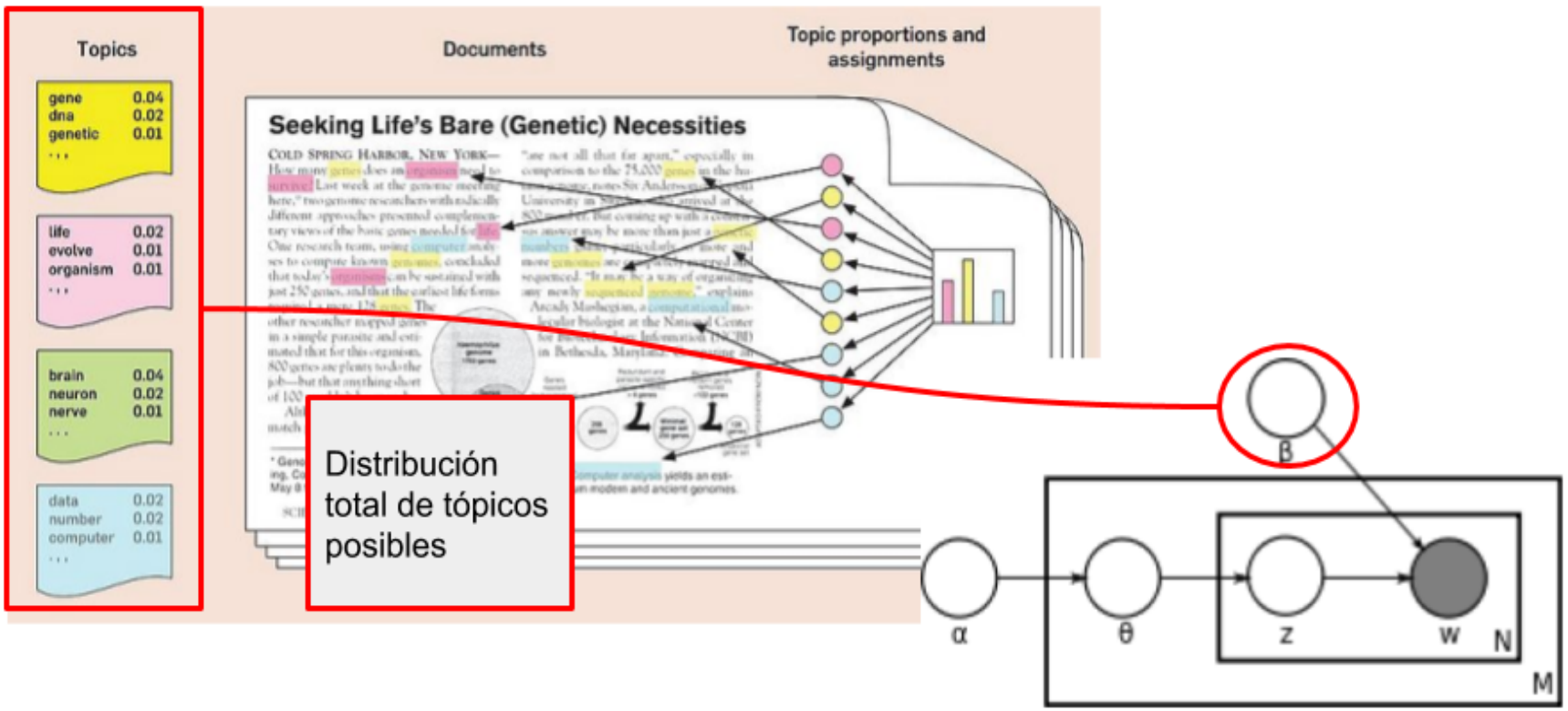
Gráficamente...



Gráficamente...



Gráficamente...



Topic Modelling

Ejercicio - Detección de tópicos en discursos presidenciales

- A partir del dataset de discursos presidenciales construiremos un detector de tópicos basado en LDA.

Referencias bibliográficas I

Blei, D. M.
Probabilistic topic models.
Commun. ACM 55, 4 (Apr. 2012), 77–84.

Feldman, R., and Sanger, J.
Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.
Cambridge University Press, New York, NY, USA, 2006.

Wiedemann, G.
Text Mining for Qualitative Data Analysis in the Social Sciences. A Study on Democratic Discourse in Germany.
Springer, 2015.