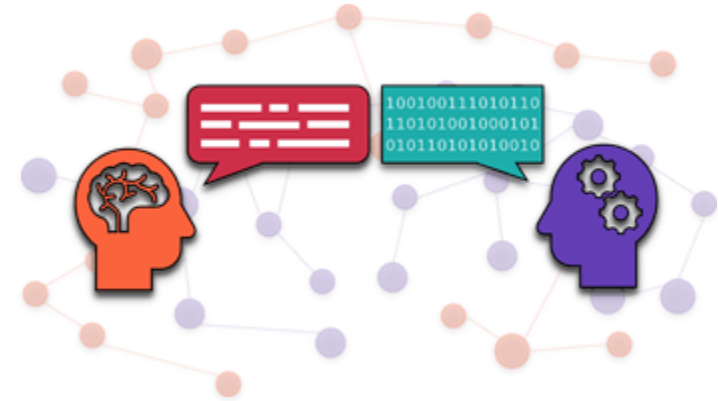


:: U3 ::

Procesamiento del Lenguaje Natural

Uso de modelos lingüísticos



Curso 2023-24

Tabla de contenidos

1. Introducció
2. Tècniques y modelos lingüístics
 - a. Bag-of-words + Naïve Bayes
 - b. TF-IDF
 - c. Topic Modeling
3. Modelos generativos
 - LLMs (Large Language Models)
 - Word Embeddings
 - LangChain
4. Aplicaciones prácticas de PLN
 - Desarrollo de chatbots (RASA)



Tabla de contenidos

1. Introducció
2. Tècniques y modelos lingüístics
 - a. Bag-of-words + Naïve Bayes
 - b. TF-IDF
 - c. Topic Modeling
3. Modelos generativos
 - LLMs (Large Language Models)
 - Word Embeddings
 - LangChain
4. Aplicaciones prácticas de PLN
 - Desarrollo de chatbots (RASA)

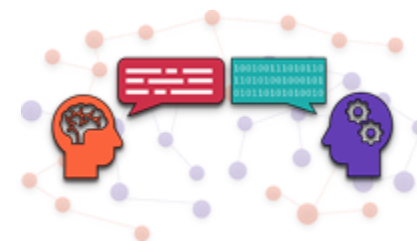


1. Introducció



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Los modelos lingüísticos son modelos **probabilísticos** del lenguaje que se basan en **representaciones numéricas del lenguaje**.
- Las aplicaciones de estos modelos son múltiples:
 - traducción automática,
 - chatbots,
 - análisis de sentimientos,
 - generación de texto o código, de resúmenes, ...



2. Uso de modelos lingüísticos

- **Ejemplo:** “No soy sabio por saber que no sé nada”

¿Qué nos diría un modelo probabilístico basado en palabras con letra inicial “s”?

¿Y si el modelo se basara en la probabilidad de encontrar unas palabras junto a otras?



1. Introducción



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

¿Qué tienen en común estos 3 tipos de problemas?

- Identificar si el remitente de un correo electrónico anónimo pertenece a una lista de contactos previa.
- Averiguar si cierto correo electrónico es o no SPAM.
- Determinar si un producto comercial está bien valorado o no.



1. Introducción



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

¿Qué tienen en común estos 3 tipos de problemas?

- Identificar si el remitente de un correo electrónico anónimo pertenece a una lista de contactos previa.
- Averiguar si cierto correo electrónico es o no SPAM.
- Determinar si un producto comercial está bien valorado o no.

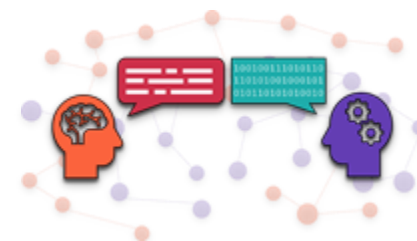
... tienen respuesta aplicando un análisis basado en Naïve Bayes + "Saco de Palabras" (Bag of Words)



2. Uso de modelos lingüísticos

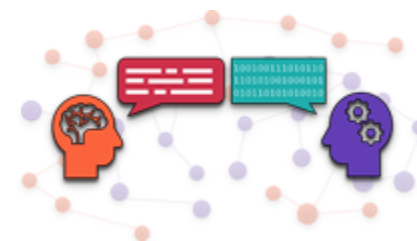
- **Saco de palabras** (bag-of-words)

- El modelo de *saco de palabras* es una técnica de procesamiento de lenguaje natural que se utiliza para representar documentos de texto
- **Objetivo del modelo BoW:** tratar un documento como una "bolsa" (conjunto) de palabras, sin tener en cuenta la estructura gramatical y el orden de las palabras en el texto, sino simplemente contando la **frecuencia de cada palabra en el documento.**
- Este modelo sirve de base para la aplicación de técnicas de análisis estadístico (por ej. *Naive Bayes*)



2. Uso de modelos lingüísticos

- El modelo **BoW** se construye en base a 3 elementos ...
 - i) Tokenización
 - ii) Diccionario de características
 - iii) Vectorización



2. Uso de modelos lingüísticos

- **Diccionarios BoW**

Una forma intuitiva de pensar cómo implementar el modelo BoW en Python sería usar un diccionario

The squids jumped out of the suitcases.



2. Uso de modelos lingüísticos

> Diccionario de características

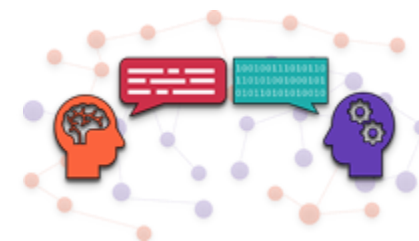
- las claves corresponden a los términos
- los valores indican la posición del término en el conjunto del texto

Ejemplo:

> documento: “No soy sabio por saber que no sé nada”

> diccionario de características:

[“No”: 0, “ser”: 1, “sabio”: 2, “por”: 3, “saber”: 4, “que”: 5, “nada”: 6]





2. Uso de modelos lingüísticos

> Vectorización

Supone construir un vector con las características (frecuencia de aparición de cada término) para cierto documento de prueba.



2. Uso de modelos lingüísticos



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **Ejemplo:**

- documento: “No soy sabio por saber que no sé nada”



2. Uso de modelos lingüísticos

- **Ejemplo:**

- documento: “No soy sabio por saber que no sé nada”
- diccionario de características (**datos entrenamiento**):

[“No”:0, “ser”:1, “sabio”:2, “por”:3, “saber”:4, “que”:5, “nada”:6]



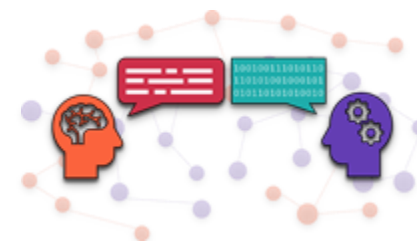
2. Uso de modelos lingüísticos

- **Ejemplo:**

- documento: “No soy sabio por saber que no sé nada”
- diccionario de características (**datos entrenamiento**):
[“No”:0, “ser”:1, “sabio”:2, “por”:3, “saber”:4, “que”:5, “nada”:6]
- vector de características (**datos de prueba**):

“**Eres** un tipo **sabio**”

[0, **1**, **1**, 0, 0, 0, 0]



2. Uso de modelos lingüísticos

- BoW mediante **CountVectorizer**:

```
from sklearn.feature_extraction.text import CountVectorizer

documentos_entrenamiento = ["Diez divertidos delfines nadaron hacia destinos desconocidos.",
                             "¿Quizás encuentren otros diez delfines divertidos?",
                             "¡Encuentra mis delfines con una función, por favor!"]
texto_prueba = ["Otros cinco delfines encuentran otro delfín lejano."]

bow_vectorizador = CountVectorizer()

# Creamos el diccionario BoW mediante el método "fit" del objeto "CountVectorizer"
bow_vectorizador.fit(documentos_entrenamiento)

# Mostramos las palabras encontradas
print("Palabras: ")
print(bow_vectorizador.get_feature_names_out())
```



2. Uso de modelos lingüísticos

- BoW mediante **CountVectorizer**:

Palabras:

```
['con' 'delfines' 'desconocidos' 'destinos' 'diez' 'divertidos'  
'encuentra' 'encuentren' 'favor' 'función' 'hacia' 'mis' 'nadaron'  
'otros' 'por' 'quizás' 'una']
```



2. Uso de modelos lingüísticos

- BoW mediante **CountVectorizer**:

```
# Mostramos el diccionario BoW
print("\nDiccionario BoW (palabras + índices): ")
print(bow_vectorizador.vocabulary_)

----

Diccionario BoW (palabras + índices):
{'diez': 4, 'divertidos': 5, 'delfines': 1, 'nadaron': 12, 'hacia': 10, 'destinos': 3, 'desconocidos': 2,
'quizás': 15, 'encuentren': 7, 'otros': 13, 'encuentra': 6, 'mis': 11, 'con': 0, 'una': 16, 'función': 9,
'por': 14, 'favor': 8}
```



2. Uso de modelos lingüísticos

- BoW mediante **CountVectorizer**:

```
# Generamos y mostramos la forma vectorizada el corpus de entrenamiento
bow_vector = bow_vectorizador.transform(documentos_entrenamiento)
print("\nVector completo para el texto de entrenamiento: ")
print(bow_vector.toarray())
```

Vector completo para el texto de entrenamiento:

```
[[0 1 1 1 1 1 0 0 0 0 1 0 1 0 0 0]
 [0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1]
 [1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 1]]
```



2. Uso de modelos lingüísticos

- BoW mediante **CountVectorizer**:

```
# Generamos el vector para la palabra "divertidos"
print("\nVector para 'divertidos': ")
print(bow_vectorizador.transform(['divertidos']).toarray())

----

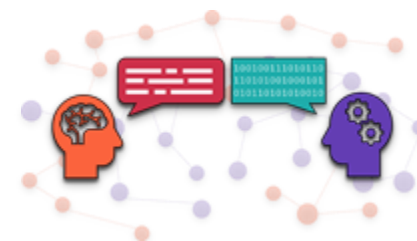
Vector para 'divertidos':
[[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]]
```



2. Uso de modelos lingüísticos

● Ejercicio :: Más sobre Countvectorizer()

- En el ejemplo anterior, hemos usado *Countvectorizer* **sin** argumentos.
- Implementa ahora una variante del script previo en la que el modelo descarte ciertas **stopwords** que se pasarán como argumento.
- Consulta [la página de Scikit-Learn sobre Countvectorizer](#) para indagar cómo hacerlo.
- Examina qué cambios ocurren entre ambas versiones.



2. Uso de modelos lingüísticos

- **Ventajas de BoW:**

- **Simplicidad:** BoW es fácil de entender e implementar.
- **Eficiencia computacional:** es computacionalmente eficiente y rápido de calcular.
- **Versatilidad:** se puede utilizar en diversas tareas de procesamiento de lenguaje natural.
- **Robustez ante ruido:** maneja texto con errores o redundancias sin afectar significativamente su desempeño.
- **Interpretabilidad:** fácil de interpretar, permitiendo examinar directamente las palabras que contribuyen más a la representación del documento.



2. Uso de modelos lingüísticos

- **Inconvenientes de BoW:**

- a. baja capacidad predictiva**

BoW no es un modelo muy preciso para la predicción del lenguaje, ya que la probabilidad de cierta palabra se relaciona sólo con aquellas utilizadas más frecuentemente.

- b. descontextualización**

Los términos en un modelo BoW carecen de contexto, lo que puede complicar la comprensión del significado de una palabra.

- c. tamaño del vocabulario**

Puede generar vectores de características largos y dispersos, aumentando la complejidad computacional y de almacenamiento.



2. Uso de modelos lingüísticos

- **Práctica 3.1**

- Realiza la práctica “**PLN - P3.1 :: El amigo misterioso**”, cuyo enunciado encontrarás en el moodle

- **Práctica 3.2**

- Realiza la práctica “**PLN - P3.2 :: Creación de un filtro de SPAM**”, cuyo enunciado encontrarás en el moodle

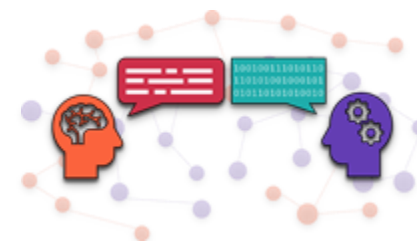
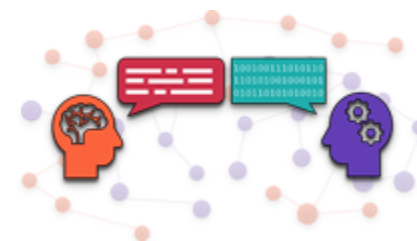


Tabla de contenidos

1. Introducció
2. Tècniques y modelos lingüístics
 - a. Bag-of-words + Naïve Bayes
 - b. **TF-IDF**
 - c. Topic Modeling
3. Modelos generativos
 - a. Word Embeddings
 - b. LangChain: integració con LLMs (Large Language Models)
 - c. Desarrollo de chatbots (RASA)



2. Uso de modelos lingüísticos

Término frecuencia / frecuencia inversa en documento (tf-idf)

- **tf-idf** es una técnica estadística que cuantifica la importancia de una palabra en un documento en función de la frecuencia con la que aparece en ese documento y en una determinada colección de documentos (corpus)
- Usos:
 - clasificar resultados en un motor de búsqueda
 - hacer resúmenes de texto
 - construir chatbots inteligentes
 - ...



2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

TF



Frecuencia de una palabra
en un documento

IDF



Frecuencia de una palabra
en un *corpus*



2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- La técnica asigna puntuaciones a las palabras:
 - si una palabra aparece con frecuencia en un documento, entonces debería ser más importante y relevante que otras palabras que aparecen menos veces y deberíamos darle a esa palabra una puntuación alta (**tf**).
 - ... pero si una palabra aparece muchas veces en un documento pero también en muchos otros documentos, probablemente no sea una palabra relevante, por lo que deberíamos asignarle una puntuación más baja (**idf**).



2. Uso de modelos lingüísticos

**Término
frecuencia /
frecuencia
inversa en
documento
(tf-idf)**

- Pasos en la creación del modelo:
 1. se calculan los valores de **frecuencia de un término** en un documento
 2. se calcula los valores de **frecuencia inversa de documento del término en el corpus**
 3. se combina (1) y (2) para crear la matriz **tf-idf**



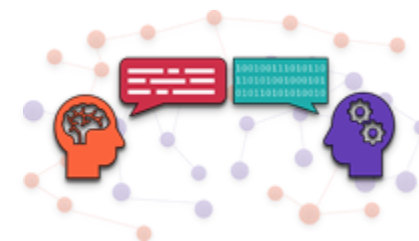
2. Uso de modelos lingüísticos

**Término
frecuencia /
frecuencia
inversa en
documento
(tf-idf)**

○ Pasos en la creación del modelo:

1. **TF(t,d)** (Frecuencia del término) mide cuántas veces aparece el término **t** en el documento **d**.

$$TF(t, d) = \frac{\text{Número de veces que } t \text{ aparece en } d}{\text{Número total de términos en } d}$$



2. Uso de modelos lingüísticos

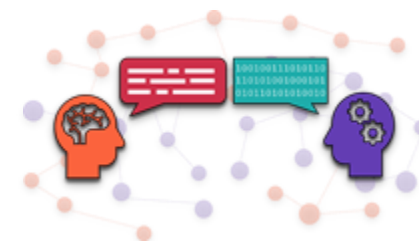
**Término
frecuencia /
frecuencia
inversa en
documento
(tf-idf)**

- Pasos en la creación del modelo:

2. mide la rareza del término ***t*** en el conjunto de documentos ***D***.

$$\text{IDF}(t, D) = \log \left(\frac{\text{Número total de documentos en } D}{\text{Número de documentos que contienen } t+1} \right)$$

Al agregar 1 en el denominador, se evita la división por cero en el caso de que el término no aparezca en ningún documento.



2. Uso de modelos lingüísticos

Término

frecuencia /

frecuencia

inversa en

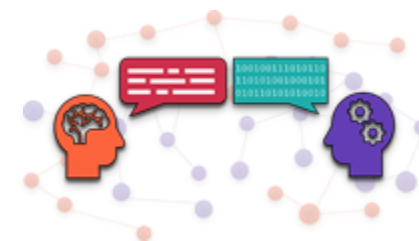
documento

(tf-idf)

- Pasos en la creación del modelo:

3. se combina (1) y (2) para crear la
matriz tf-idf

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

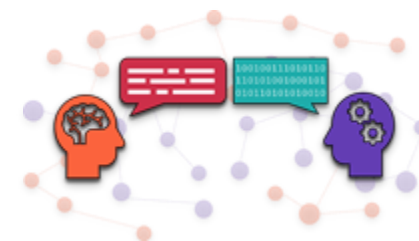


2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** partimos del siguiente *corpus*

Document	Text Data
D1	My name is Naftal.
D2	My car is a Hyundai.
D3	The car I drive is a Hyundai Sonata.
D4	My car is a Sonata model by Hyundai !!

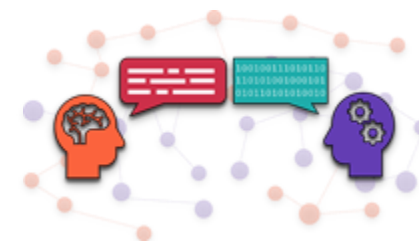


2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** preprocesamos su contenido

Document	Text Data
D1	name naftal
D2	car hyundai
D3	car drive hyundai sonata
D4	car sonata model hyundai



2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** se crea una matriz **tf** para el corpus ...

Document	name	naftal	car	hyundai	drive	sonata	model
D1	1	1	0	0	0	0	0
D2	0	0	1	1	0	0	0
D3	0	0	1	1	1	1	0
D4	0	0	1	1	0	1	1

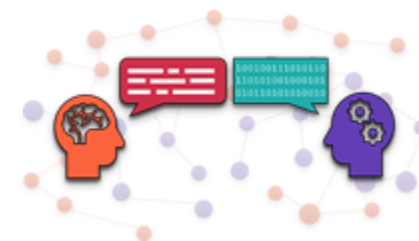


2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** ... normalizando sus valores

Document	name	naftal	car	hyundai	drive	sonata	model
D1	1/2	1/2	0	0	0	0	0
D2	0	0	1/2	1/2	0	0	0
D3	0	0	1/4	1/4	1/4	1/4	0
D4	0	0	1/4	1/4	0	1/4	1/4



2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** se calculan los valores **idf**

name	naftal	car	hyundai	drive	sonata	model
0.602	0.602	0.125	0.125	0.602	0.301	0.602

$$\text{IDF}(t, D) = \log \left(\frac{\text{Número total de documentos en } D}{\text{Número de documentos que contienen } t+1} \right)$$




2. Uso de modelos lingüísticos

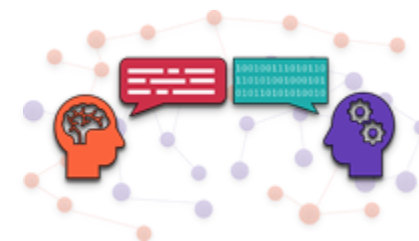
> Término frecuencia / frecuencia inversa en documento (tf-idf)

- **Ejemplo:** se calcula la matriz **tf-idf** asociada

$$1/2 * 0.602$$



Document	name	naftal	car	hyundai	drive	sonata	model
D1	0.301	0.301	0	0	0	0	0
D2	0	0	0.0625	0.0625	0	0	0
D3	0	0	0.03125	0.03125	0.1505	0.075	0
D4	0	0	0.03125	0.03125	0	0.075	0.1505



2. Uso de modelos lingüísticos

> Término frecuencia / frecuencia inversa en documento (tf-idf)

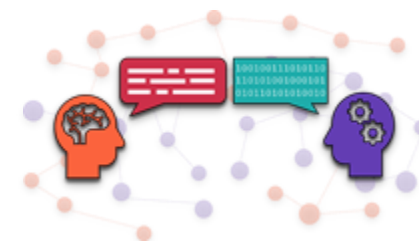
- **Ejemplo:** se calcula la matriz **tf-idf** asociada

Document	name	naftal	car	hyundai	drive	sonata	model
D1	0.301	0.301	0	0	0	0	0
D2	0	0	0.0625	0.0625	0	0	0
D3	0	0	0.03125	0.03125	0.1505	0.075	0
D4	0	0	0.03125	0.03125	0	0.075	0.1505



Document	Text Data
D1	name naftal
D2	car hyundai
D3	car drive hyundai sonata
D4	car sonata model hyundai

- los valores para un término en cierto documento dan una idea de su importancia relativa en el conjunto
- cuanto más alto sea éste mayor relevancia se le puede suponer



2. Uso de modelos lingüísticos

**Término
frecuencia /
frecuencia
inversa en
documento
(tf-idf)**

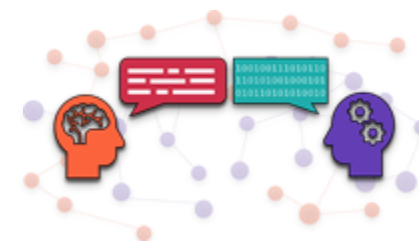
- La librería Scikit-Learn proporciona 3 clases para implementar este modelo:

1. TfidfVectorizer

Para obtener la **matriz tf-idf**

1. CountVectorizer + TfidfTransformer

Para obtener **tf** e **idf** por separado



2. Uso de modelos lingüísticos

- **Práctica 3.3**

- Realiza la práctica **“PLN - P3.3 :: Analizando noticias de los medios”**, cuyo enunciado encontrarás en el moodle

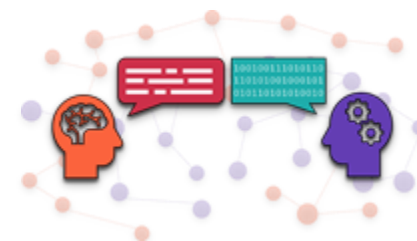


Tabla de contenidos

1. Introducció
2. Tècniques y modelos lingüístics
 - a. Bag-of-words + Naïve Bayes
 - b. TF-IDF
 - c. **Topic Modeling**
3. Modelos generativos
 - a. Word Embeddings
 - b. LangChain: integració con LLMs (Large Language Models)
 - c. Desarrollo de chatbots (RASA)



2. Uso de modelos lingüísticos

- **Clasificación temática (“topic modeling”)**

- El **topic modeling** es una técnica no supervisada de NLP, capaz de detectar y extraer de manera automática relaciones “ocultas” entre términos en grandes volúmenes de información.
- Un **tema** (tópico) ...
 - es un conjunto de palabras que suelen aparecer juntas en los mismos contextos.
 - expresan una **distribución de probabilidades** de aparición de las distintas palabras del vocabulario.



2. Uso de modelos lingüísticos



Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, these **predictions** "are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson, a Uppsala University in Sweden biologist, arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic numbers** game, particularly as more and more **genomes** are completely mapped and sequenced. "It may be a way of organizing any newly **sequenced genome**," explains Arach Muehligian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments



2. Uso de modelos lingüísticos

- **Ejemplo:** aplicación de LDA (*Latent Dirichlet Allocation*, algoritmo que implementa el *Topic Modeling*) para la identificación de temas en un conjunto de noticias de los medios generalistas

Resultados del modelo LDA

Lista de Tópicos

Tópico 0
violencia: 0.057
género: 0.045
mujer: 0.037
maltrato: 0.36
...

Violencia de género

Tópico 1
presidente: 0.075
diputados: 0.057
gobierno: 0.041
españa: 0.39
...

Política

...

Lista de Documentos

Sevilla registra 300 denuncias por
violencia de género

Tópico 0: 0.4334
Tópico 4: 0.1266
Tópico 1: 0.0623
...

La Xunta revela a los 5 diputados
territoriales

Tópico 1: 0.3934
Tópico 6: 0.0555
Tópico 9: 0.0467
...

...

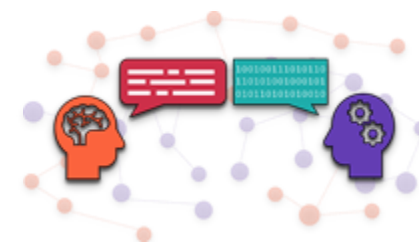
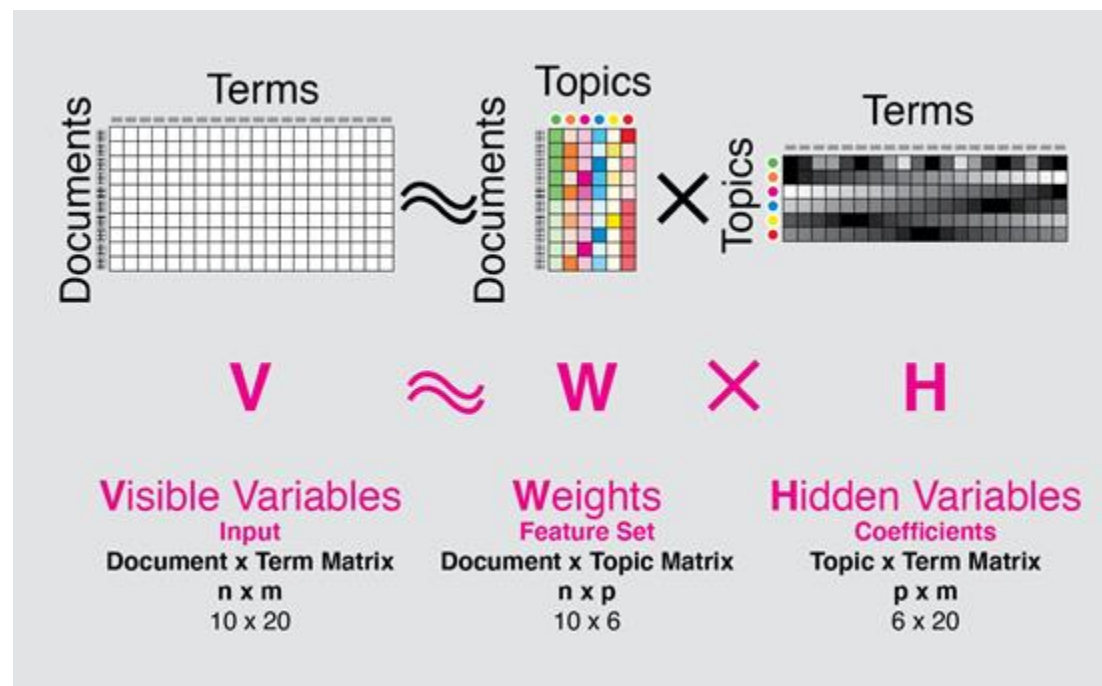


2. Uso de modelos lingüísticos

- Scikit-Learn implementa **LDA** generando lo que se denomina **matriz de términos del documento** (**V** en la imagen)

... que descompone en ...

- matriz de documento-temas** (**W** en la imagen)
- matriz de tema-términos** (**H** en la imagen)

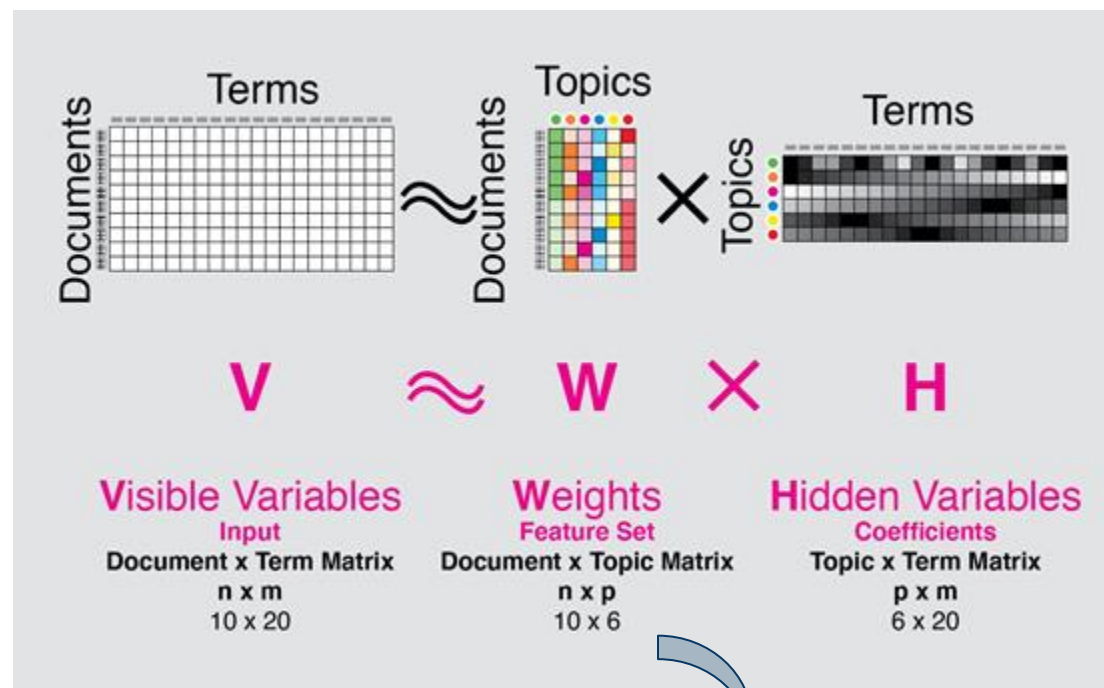


2. Uso de modelos lingüísticos

- Scikit-Learn implementa **LDA** generando lo que se denomina **matriz de términos del documento** (**V** en la imagen)

... que descompone en ...

- matriz de documento-temas** (**W** en la imagen)
- matriz de tema-términos** (**H** en la imagen)



Objetivo

- Cada término hará una contribución positiva a un conjunto de temas ...
- Cada tema puede ser descrito por una combinación de términos





2. Uso de modelos lingüísticos

- **Práctica 3.4**

- Realiza la práctica “**PLN - P3.4 :: Topic modeling vs K-Means**”, cuyo enunciado encontrarás en el moodle

