

Estudio sobre Sistemas de Recomendación y Predicción basados en el procesamiento del lenguaje natural

Hugo Ferrando Seage

25 de julio de 2017

Universidad Europea de Madrid
Escuela de Arquitectura, Ingeniería y Diseño

Introducción

Los recomendadores son una parte esencial de cualquier servicio de Video on Demand (VOD). Tanto Netflix como Movistar+, Amazon, Hulu y HBO cuentan con sus propios sistemas.

También existen webs que usan sus recomendadores, como IMDb o FilmAffinity. Incluso existen servicios comerciales que se dedican a productivizar su sistema de recomendación, como Jinni.

Existen tres grandes tipos de sistemas de recomendación:

- Filtrado colaborativo
- Filtrado por contenido
- Sistemas híbridos

Consiste en emparejar usuarios que tengan gustos similares y recomendar en base a esos datos.

Normalmente se representa usando una matriz bidimensional donde las filas representan usuarios y la columnas representan productos.

Los usuarios deben puntuar los contenidos, o se pueden usar otras métricas.

Consiste en la creación de un modelo que determina la similitud entre productos en base a algún criterio.

Ese criterio puede ser cualquier elemento del producto. Para películas puede ser el género. Para restaurantes el tipo de cocina. Etc.

Usan una combinación de ambas técnicas para complementar las recomendaciones.

Objetivos

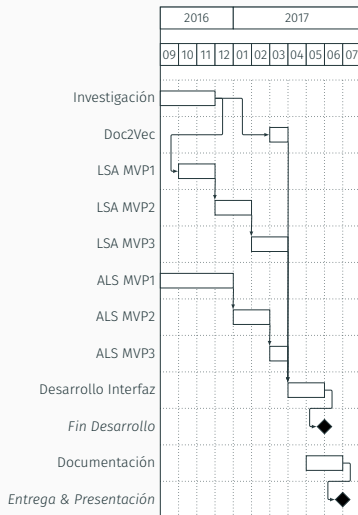
Objetivos

- Construir un recomendador de películas
- Crear el modelo en base a tres algoritmos
 - LSA
 - Doc2Vec
 - E-Modelo
- Optimizar modelos
- Crear una interfaz desde donde poder probarlos

Metodología

Metodología

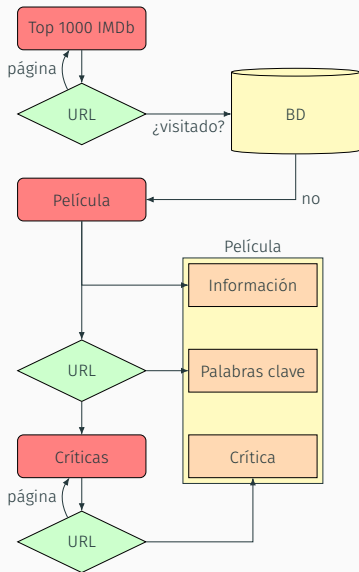
La metodología usada ha sido ágil, basada en MVPs.



Descarga de datos

Para entrenar los modelos es necesario obtener una gran cantidad de textos. Para ello se ha creado un crawler usando Spidy, que descarga información y críticas de las top 1000 películas de IMDb.

Descarga de datos



Limpieza de textos

Antes de crear los modelos es necesario hacer un pretratado de los textos.

Zeus is a Greek God.

NNP	VBZ	DT	NN	NNP
Zeus	is	a	Greek	God.

Zeus is a country deity.

Es necesario eliminar los nombres propios para que no se relacionen películas con personajes que tienen el mismo nombre.

`is a country deity.`

country deity

counti deiti

LSA

Latent Semantic Analysis trata de extraer conceptos de cada texto y analizar la relación entre documentos.

Term Frequency-Inverse Document Frequency calcula lo relevante que es cada palabra del vocabulario dentro de cada texto.

	says	just	room	dead	asks	ship	mother
$tfidf =$							
The Matrix	0,39	0,16	0,19	0,01	0,25	0,79	0,27
Alien	0,12	0,12	0,06	0,46	0,21	0,07	0,83
Serenity	0,46	0,55	0,15	0,55	0,22	0,27	0,11
Casablanca	0,00	0,60	0,51	0,00	0,00	0,60	0,00
Amelie	0,41	0,00	0,35	0,83	0,00	0,00	0,00

El siguiente paso es descomponer la matriz en valores singulares.

Matriz Palabra-Concepto.

$$U = \begin{matrix} & \begin{matrix} \text{Sci-Fi topic} \\ \text{Romance topic} \\ \text{Ruido} \end{matrix} \\ \begin{matrix} \text{action} \\ \text{gun} \\ \text{shoot} \\ \text{run} \\ \text{love} \\ \text{peace} \\ \text{kiss} \end{matrix} & \begin{pmatrix} 0,13 & 0,02 & -0,01 \\ 0,41 & 0,07 & -0,03 \\ 0,55 & 0,09 & -0,04 \\ 0,68 & 0,11 & -0,05 \\ 0,15 & -0,59 & 0,65 \\ 0,07 & -0,73 & 0,67 \\ 0,07 & -0,29 & 0,32 \end{pmatrix} \end{matrix}$$

Matriz de relevancia de Conceptos

$$\Sigma = \begin{matrix} & \begin{matrix} \text{Sci-Fi topic} \\ \text{Romance topic} \\ \text{Ruido} \end{matrix} \\ \begin{pmatrix} 12,4 & 0 & 0 \\ 0 & 9,5 & 0 \\ 0 & 0 & 1,3 \end{pmatrix} \end{matrix}$$

Matriz Película-Concepto

$$V^T = \begin{matrix} & \begin{matrix} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{matrix} \\ \begin{matrix} \text{Sci-Fi topic} \\ \text{Romance topic} \\ \text{Ruido} \end{matrix} & \begin{pmatrix} 0,56 & 0,59 & 0,56 & 0,09 & 0,09 \\ 0,12 & -0,02 & 0,12 & -0,69 & -0,69 \\ 0,40 & -0,80 & 0,40 & 0,09 & 0,09 \end{pmatrix} \end{matrix}$$

Los conceptos menos relevantes se pueden eliminar

$$V^T = \begin{matrix} & \begin{matrix} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{matrix} \\ \begin{matrix} \text{Sci-Fi topic} \\ \text{Romance topic} \end{matrix} & \begin{pmatrix} 0,56 & 0,59 & 0,56 & 0,09 & 0,09 \\ 0,12 & -0,02 & 0,12 & -0,69 & -0,69 \end{pmatrix} \end{matrix}$$

$$\cos \left(\begin{pmatrix} 0,56 \\ 0,12 \end{pmatrix}, \begin{pmatrix} 0,59 \\ -0,02 \end{pmatrix} \right) = 0,97 \quad (1)$$

Figura 1: Alta similitud entre Matrix y Alien

$$\cos \left(\begin{pmatrix} 0,56 \\ 0,12 \end{pmatrix}, \begin{pmatrix} 0,09 \\ -0,69 \end{pmatrix} \right) = -0,08 \quad (2)$$

Figura 2: Baja similitud entre Matrix y Amelie

Doc2Vec

Word2Vec es un algoritmo creado por Google en 2013. Es conceptualmente similar a LSA, pero teniendo en cuenta cada palabra dentro de su contexto.

Es decir, calcula la probabilidad de que una palabra esté en la vecindad de otra palabra en el vocabulario.

Word2Vec

En primer lugar se guardan las parejas de palabras dentro de una ventana.

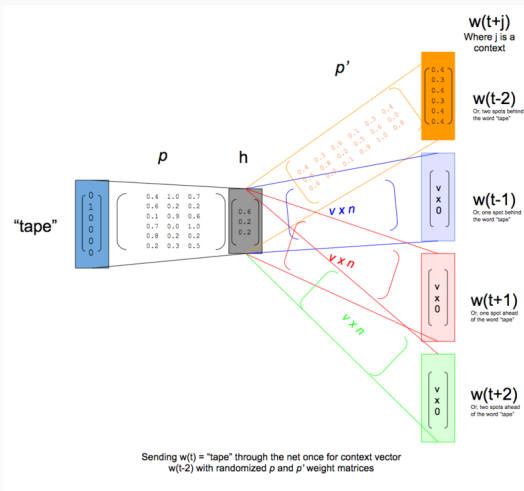
Source Text	Training Samples
The quick brown fox jumps over the lazy dog	('the', 'quick') ('the', 'brown')
The quick brown fox jumps over the lazy dog	('quick', 'the') ('quick', 'brown') ('quick', 'fox')
The quick brown fox jumps over the lazy dog	('brown', 'the') ('brown', 'quick') ('brown', 'fox') ('brown', 'jumps')
The quick brown fox jumps over the lazy dog	('fox', 'quick') ('fox', 'brown') ('fox', 'jumps') ('fox', 'over')

Las palabras del vocabulario se convierten a vectores one-hot.

Palabra	Posición por orden alfabético	Vector
fox	2/3	[0, 1, 0]
dog	1/3	[1, 0, 0]
zebra	3/3	[0, 0, 1]

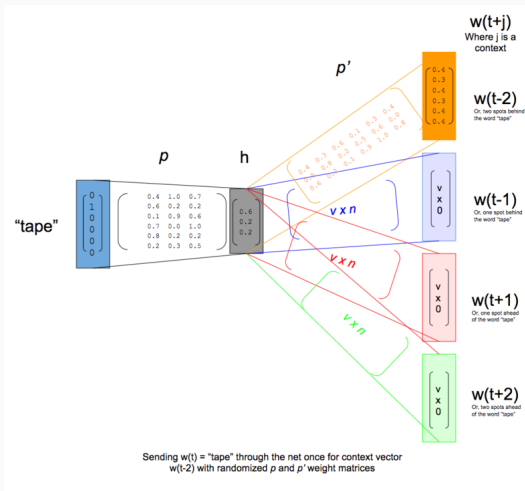
Word2Vec

Con los datos obtenidos se entrena una red neuronal con una capa oculta.



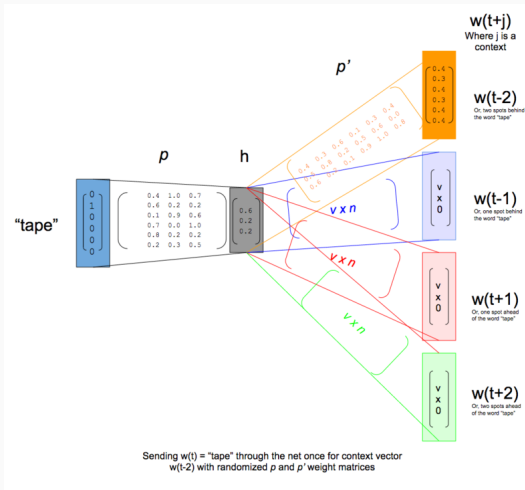
Word2Vec

El tamaño de la capa de input es igual al número de palabras en el vocabulario. La función de activación es lineal.



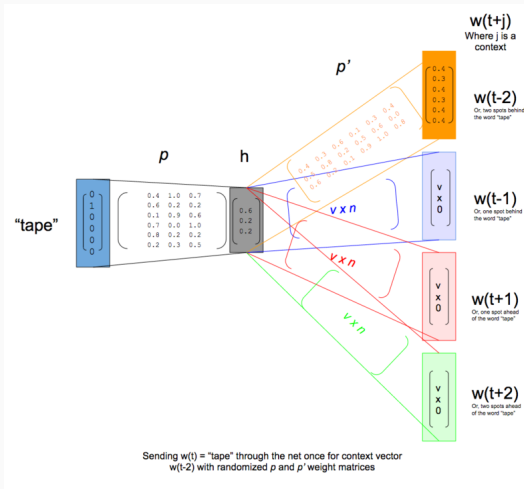
Word2Vec

h tiene tantas neuronas como componentes se quieran extraer.



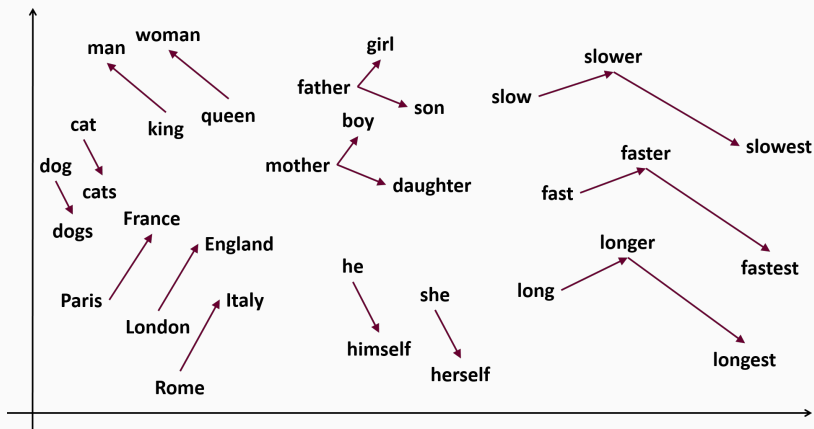
Word2Vec

La capa output tienen el mismo número de vectores que el tamaño de la ventana.



Word2Vec

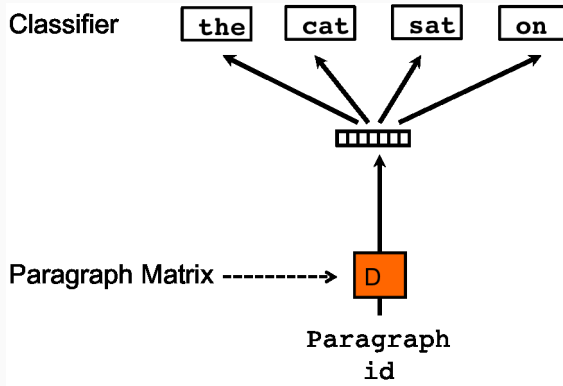
Los resultados son vectores que se pueden comparar usando la similitud del coseno de cada vector.



Palabra clave		Palabras similares			
mafia	dealer	banker	mob	mexican	drug
axe	expedition	undercover	biker	employee	outlaw
hair	clothes	coat	eyes	teeth	makeup
airplane	announcer	sanctuary	engine	accident	ambushed
plant	retrieve	investigate	thwart	unearth	escape
air	automatic	oil	swinging	oxygen	ocean

Doc2Vec

Word2Vec trabaja a nivel de palabras. Doc2Vec extiende el algoritmo para hacer comparaciones entre documentos.



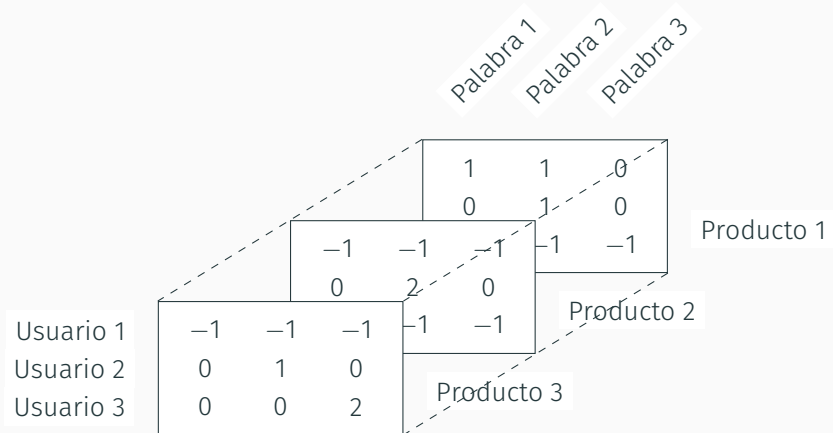
E-Modelo

Modelo de predicción de tokens híbrido.

Combina el filtrado colaborativo con los features extraídos de un filtrado por contenido.

En primer lugar extraemos tokens como se ha hecho en el pre-filtrado de textos (usando Bitext).

E-Modelo



E-Modelo

	Producto 1			Producto 2			Producto 3		
	Palabra 1	Palabra 2	Palabra 3	Palabra 1	Palabra 2	Palabra 3	Palabra 1	Palabra 2	Palabra 3
Usuario 1	1	1	0	-1	-1	-1	-1	-1	-1
Usuario 2	0	1	0	0	2	0	0	1	0
Usuario 3	-1	-1	-1	-1	-1	-1	0	0	2

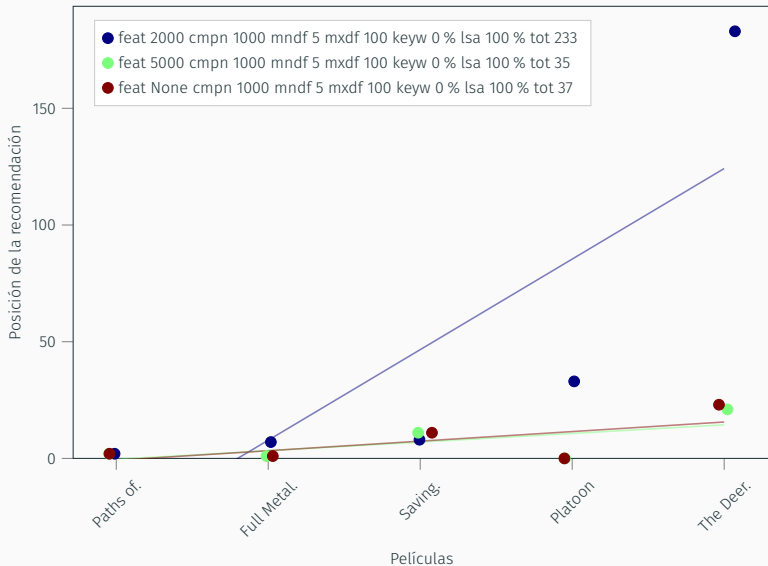
Optimización

Para cada modelo hay unos parámetros que se pueden ajustar.

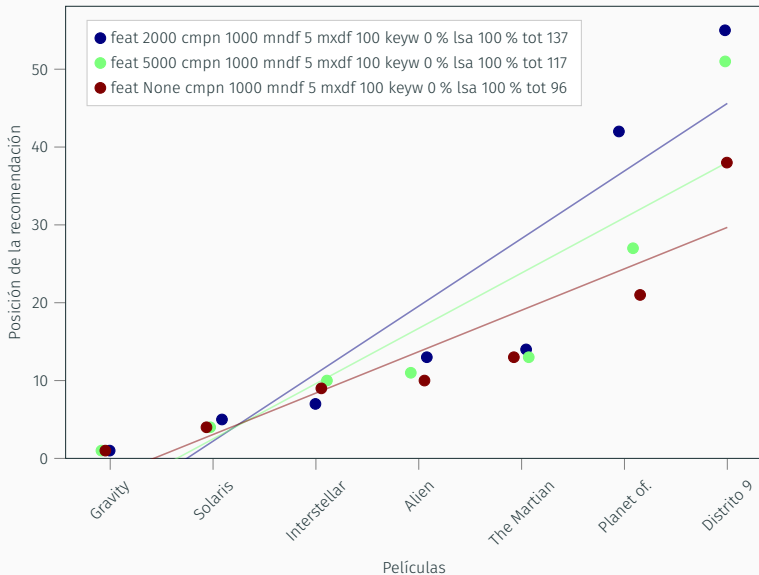
- Número de 'features' TF-IDF
- Número de componentes LSA
- Frecuencia Mínima de Documentos
- Frecuencia Máxima de Documentos

- Size
- Window
- Minimum Word Count
- Iteraciones

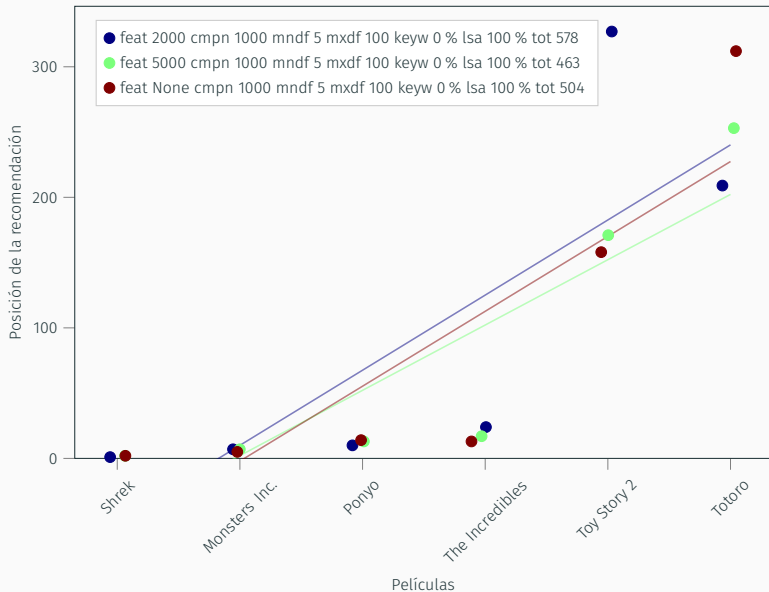
Optimización LSA



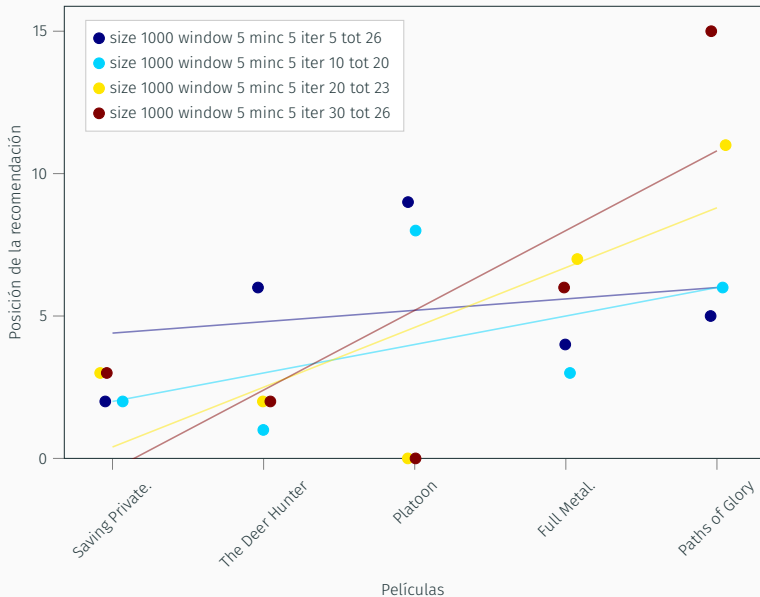
Optimización LSA



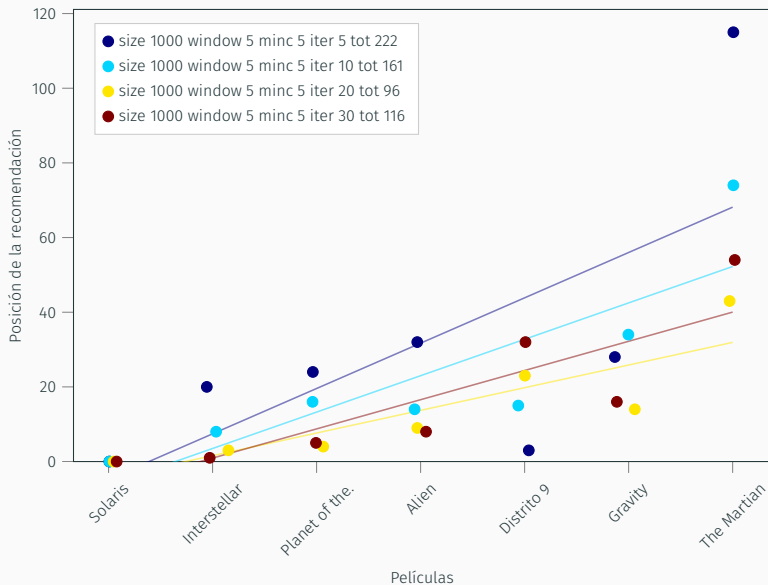
Optimización LSA



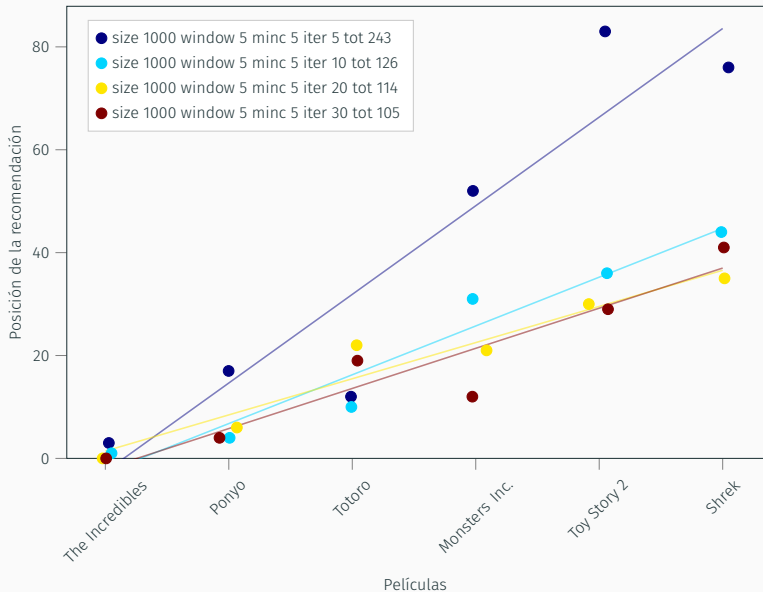
Optimización Doc2Vec



Optimización Doc2Vec



Optimización Doc2Vec



Demo

<https://moviepepper.hugofs.com>