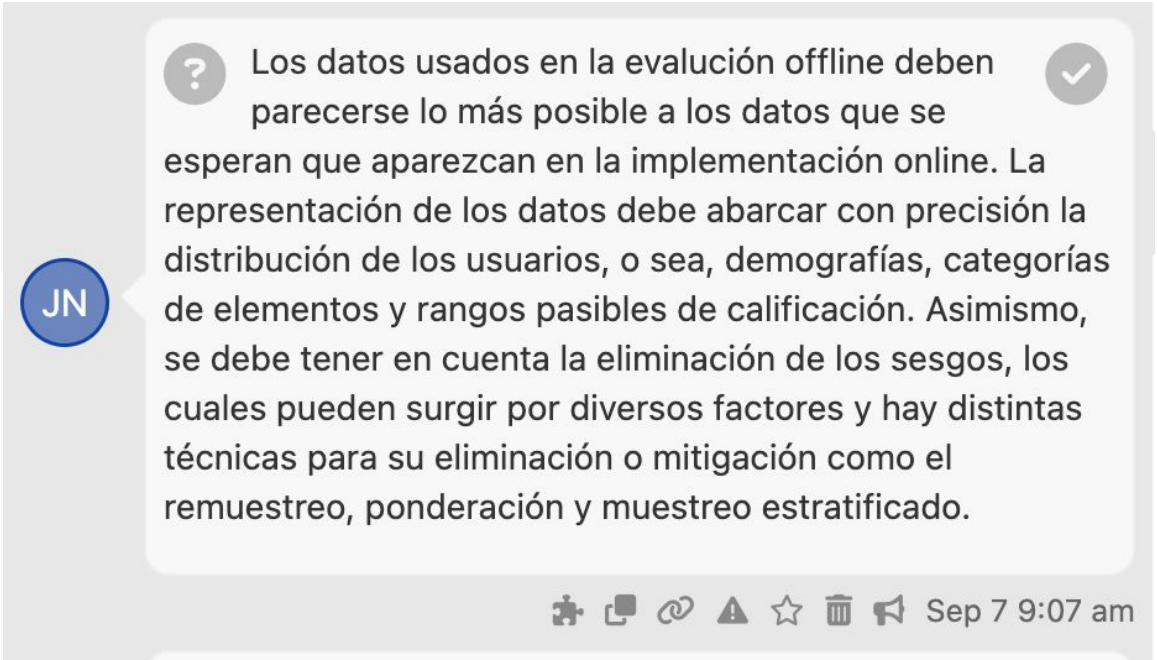
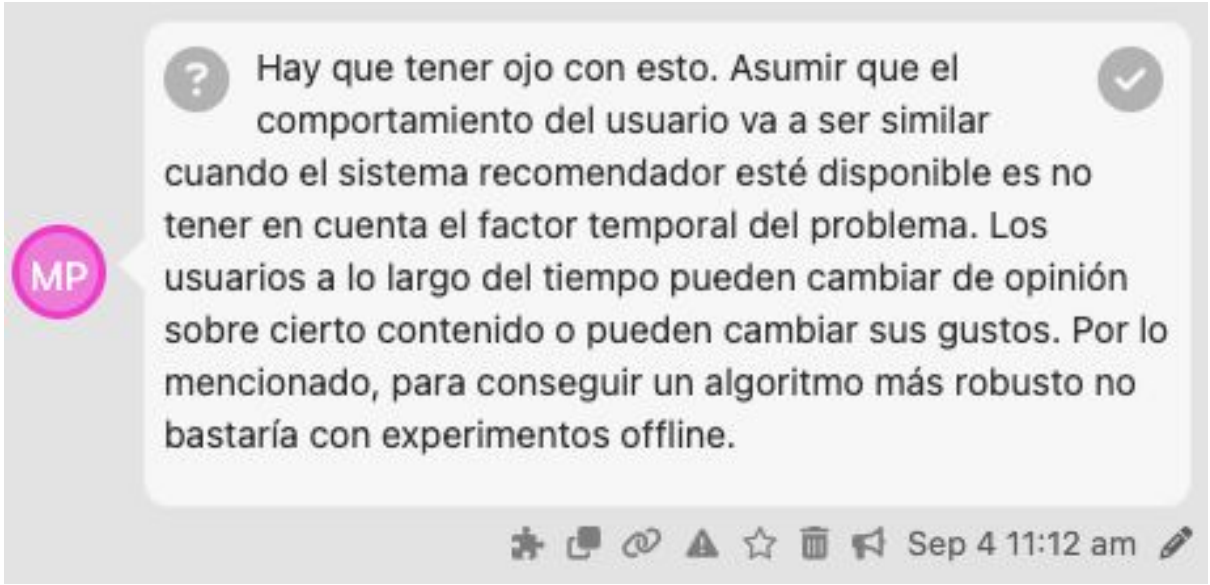
Sistemas Recomendadores
IIC-3633

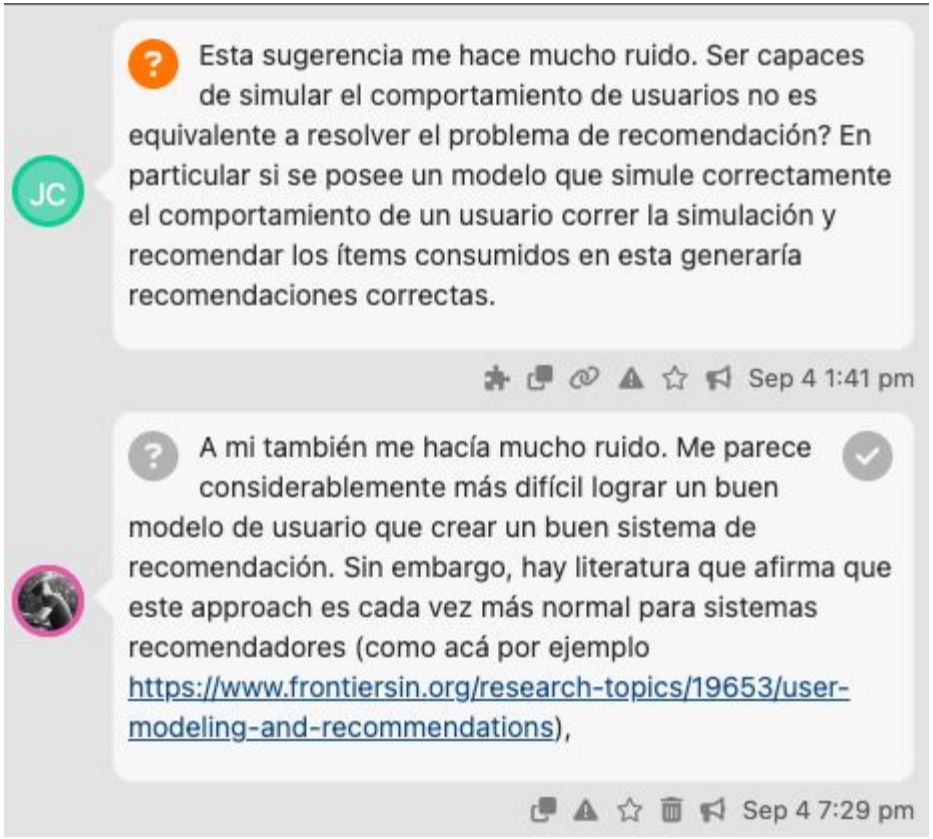
Recomendación basada en contenido
Parte 1



Hay que poner atención a como hacer la particion train, val y test para recomendacion.



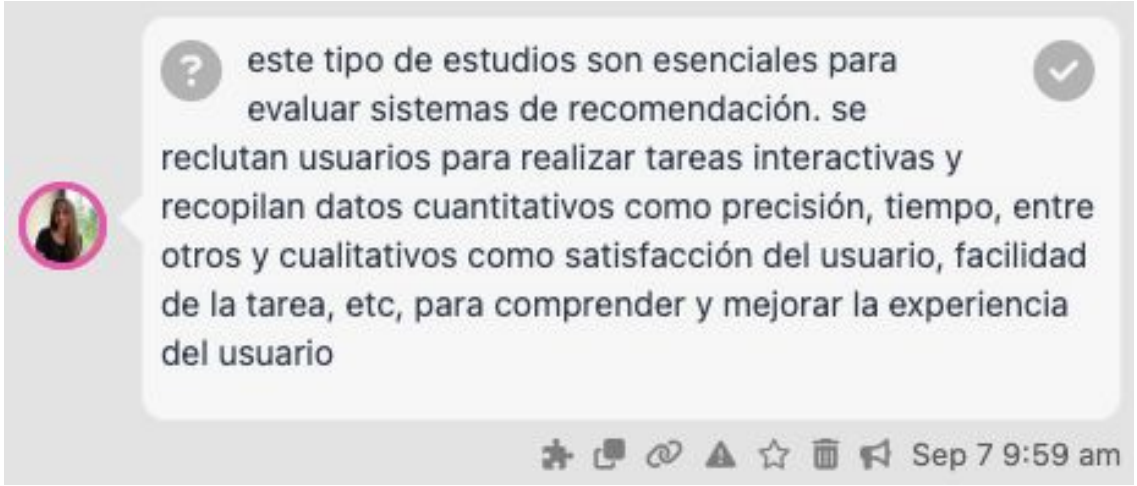
El factor temporal en recomendacion y cambios de preferencias es complejo.



Area de sistemas de recomendacion y modelamiento de usuario están relacionadas, pero recomendación apoya a la primera.

Modelar el comportamiento de usuario se puede utilizar para diversas tareas:

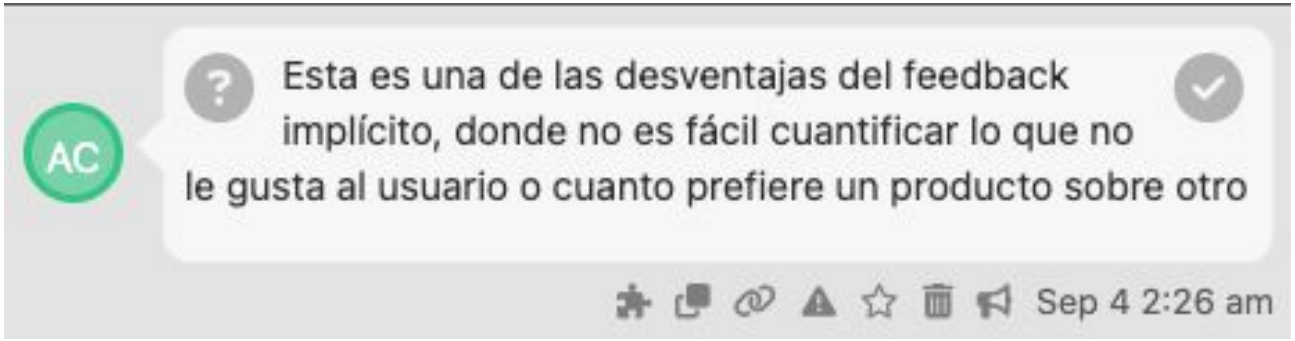
- Link prediction
- Recomendación
- Clustering



Ya la evaluación de la tarea de recomendar bien puede pasar el segundo plano.

Evaluar lo que viene después es lo relevante:

- tiempo de respuesta
- satisfacción
- experiencia de usuario
- confianza / explicabilidad



El feedback implicito asume que algo que no ha sido consumido por el usuario es menos preferido.

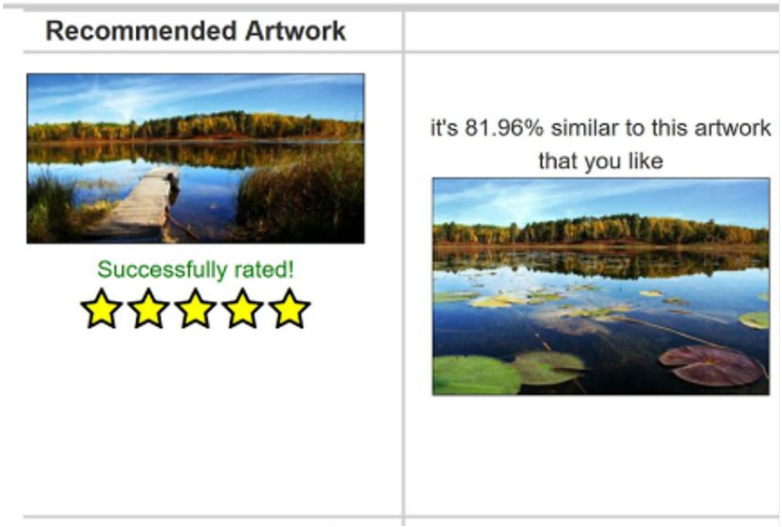
Esto se puede mitigar en parte con recomendación basada en contenido.

Esta clase

1. Recomendación basada en contenido en recomendación
2. Representación de texto

Recomendación basada en contenido

- El filtrado colaborativo funciona bien pero tiene el problema de *cold-start* y *new item.*
- Tenemos **el filtrado basado en contenido** como alternativa.



Recomendación basada en contenido de metadata de ítems

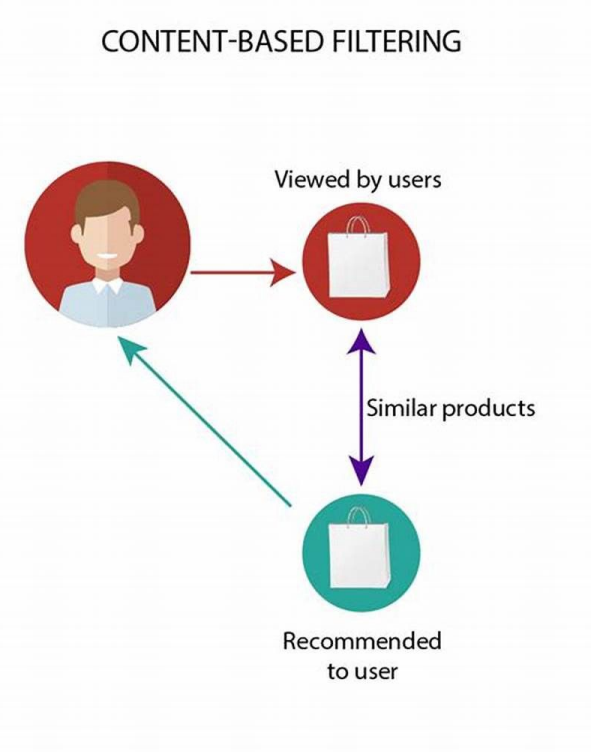
Movie	Action	Comedy	Adventure	…	…
Wolf of Wall Street	0	1	0		
Fargo	1	1	0	…	…
…					

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Recomendación basada en contenido

Basado en **características de ítems con los que el usuario ha interactuado** puedo **saber ítems con los que va a interactuar** a futuro que comparte características con ellos.

Tengo que **buscar alguna forma como "agregar" los ítems consumidos por el usuario**.



Recomendación basada en contenido (ventajas /desventajas)

ventajas:

- soluciona parte de cold-start de usuarios e ítems.
- es más transparente, “porque te gusta este contenido te recomiendo esto”
- se puede combinar para agregar información adicional al filtrado colaborativo.

desventajas:

- poca diversidad, me va a recomendar más de lo mismo que he consumido.
- **usa vectores de texto pre-entrenados (estáticos) para una tarea distinta que de recomendación.**
- **habría que entrenar el modelo de lenguaje en conjunto con el recomendador.**

Recomendación basada en contenido de texto

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

Persona: edad, ciudad , estado civil , ….

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 deshonrosos; 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.

¿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.

Vocabulario

Un **vocabulario** es una secuencia ordenada de **palabras** con un identificador único (id).

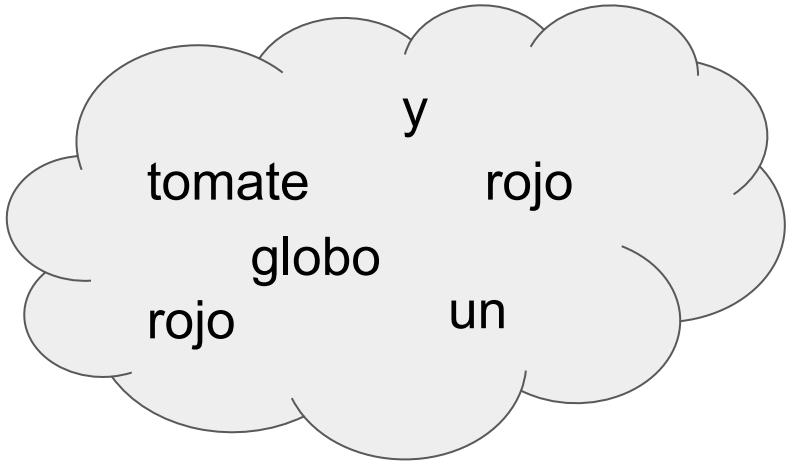
Ejemplo:

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)*

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

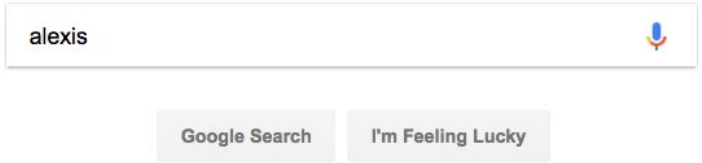
Pesos de las palabras (weighting)

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 = {alexis}



- Otra estrategia para “pesar” las palabras: logaritmo

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

si la palabra “alexis” aparece muchas veces en los documentos debería tener más importancia.

log10 se utiliza para suavizar este efecto.

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

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

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

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

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.***

Castigar si la palabra es muy frecuente en el corpus, puede ser una stop word.

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

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

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)*

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

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

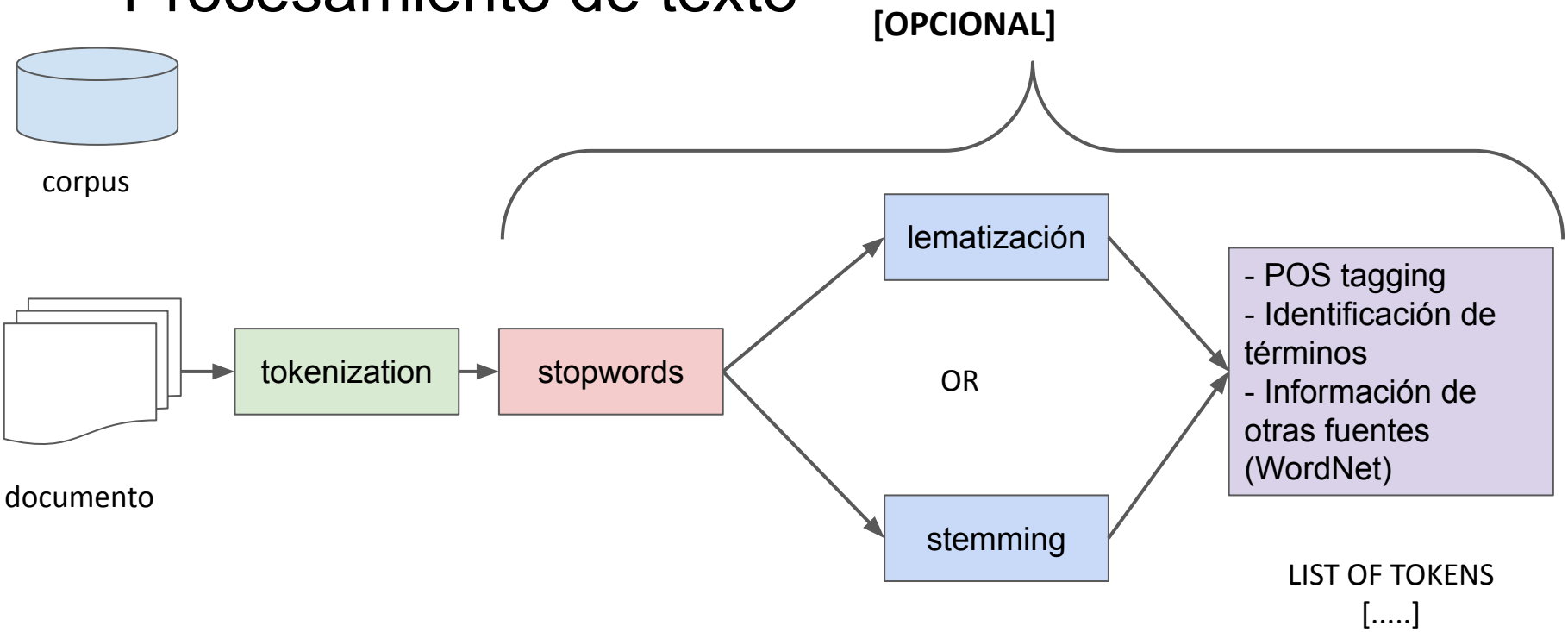
Documento 2: Un tomate rojo y un globo rojo.

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

Procesamiento de texto



por ejemplo:

- Para modelos de lenguaje basados en contexto puede influir remover stopword “NO”, pues cambia el sentido de la oración particular (contexto)
- A veces palabra original sin stemming puede tener sentido en otro contexto

Conceptos NLP: lematización

Lematización (reduce redundancia):

- reducir palabras a su raíz semántica.
- ejemplo: jugando, jugabais, jugamos → jugar
- ejemplo: resurrección → resucitar.

Conceptos NLP: stemming

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

Stemming (reduce redundancia):

- elimina el sufijo de una palabra.

Otro tipo de reducción de redundancia de palabras, pero en este caso elimina el sufijo de la palabra.

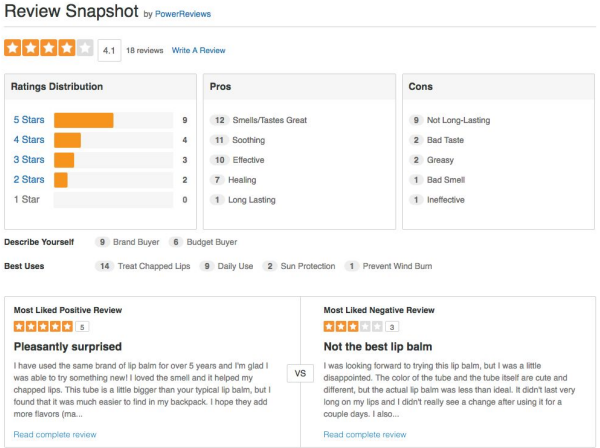
¿Cómo hacemos uso de esta información para recomendación?

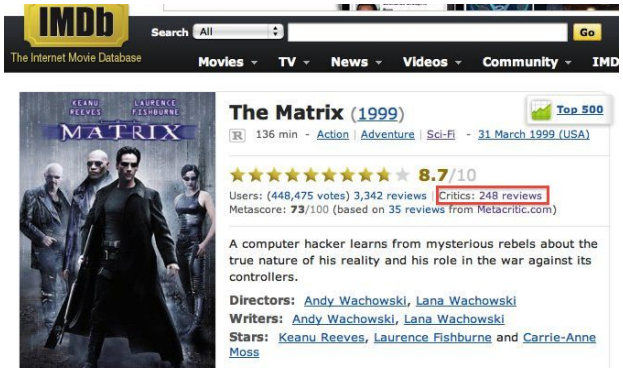
Características de contenido (texto información del usuario)

- Descripción del perfil del usuario.
- Descripción de ítems.
- Reviews de ítems.
- Tweets de ítems.
- Noticias de ítems.
- Cruce de BD, ej. IMDB.

User profile: soluciona parcialmente “cold start” usuarios que no han participado en la plataforma.

Contenido texto items





¿Cómo funciona?

- Tengo que representar el texto como vectores y hacer uso de estos vectores para la recomendación.

Ejemplos:

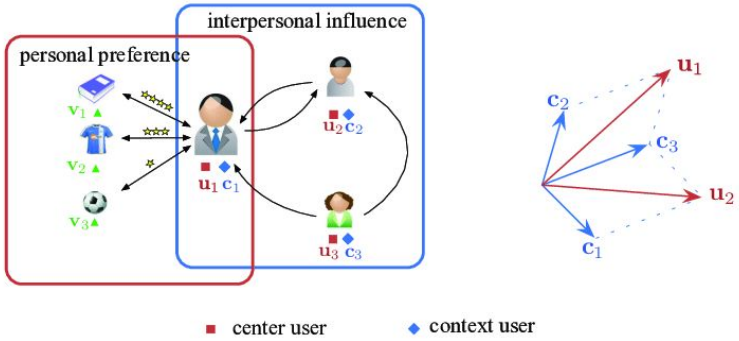
- Representar a cada usuario como:
  - vector del texto de todas las reviews que ha hecho.
  - vector del texto de su perfil.
  - vector del texto de los tweets de los ítems con los que ha interactuado

- Representar cada ítem como:
  - vector de texto de sus reviews
  - vector de texto de su descripción

Recomendación

Como **vector del usuario va a tener las mismas dimensiones que los vectores de ítems** puedo buscar las más cercanas con alguna métrica de similaridad.

También puedo buscar usuarios cercanos en el espacio y recomendar los ítems con los que otros han interactuado (como filtrado colaborativo).



Recipe Recommendation System Using TF-IDF

Shubham Chhipa[1], *Vishal Berwal*[2], *Tushar Hirapure*[3] *and Soumi Banerjee*[4]

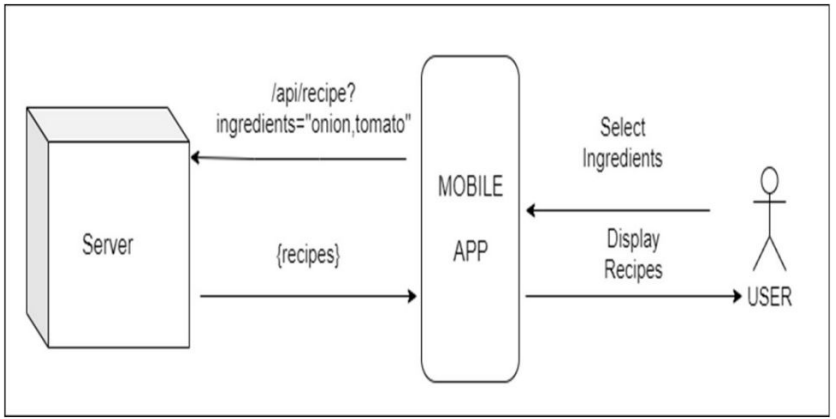
[1]*Department of Information Technology, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai.*
[2] *Department of Information Technology, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai.*
[3] *Department of Information Technology, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai.*
[4] *Department of Information Technology, Ramrao Adik Institute of Technology, D.Y. Patil Deemed to be University, Navi Mumbai.*

Abstract - A Recipe Recommendation System is being proposed in this following paper. Food recommendation is a new area, with few systems that are focus on analysing and user preferences and constraints such as ingredients available at their side being deployed in real settings in the form of web application or mobile application [4]. The proposed model is a mobile application which allows users to search recipes using ingredients available at them including vegetables. For this work we have find a dataset which is a collection of Indian cuisines recipes and apply the content-based recommendation using Term Frequency – Inverse Document Frequency (TF-IDF) and Cosine Similarity [1]. This application gives the recommendation of Indian recipes based on ingredients available at them and allows users to filter out the recipes on course type, diet type, etc.

Recomendación de recetas

Perfil del usuario se crea a partir de:

- preferencias alimentarias.
- tipo de comida.
- dietas.
- almuerzo, cena, desayuno.



Representa recetas con TF-IDF

Busca las más cercanas

Usuario puede editar ingredientes y dar feedback al modelo.

Dataset

KANISHK JAIN · UPDATED 3 YEARS AGO

93 | New Notebook | Download (10 MB)

6000+ Indian Food Recipes Dataset

Food Recipes Dataset generated by crawling archanaskitchen.com.

https://www.kaggle.com/datasets/kanishk307/6000-indian-food-recipes-dataset/code

Resultados cualitativos

Using Lunch Filter

Sr No	Recommendations
1	Aloo Ke Gutte Recipe
2	Niramish Aloo Recipe

Using Side Dish Filter

Sr No	Recommendations
1	Konkani Style Batata Recipe
2	Parsi Style Chas Payelo Sakarkand Recipe
3	Odisha Style Aloo Recipe

Chinese Cuisine Recipes

Sr No	Recommendations
1	Idli Manchurian Recipe With Oriental Twist
2	Chilli Baby Corn Manchurian Recipe
3	Chilli Paneer Momo Recipe
4	Spicy Schezwaan Noodles

En esta clase

1. Recomendación basada en contenido en recomendación
2. Representación de texto

Ideas de proyectos

Musica: Datos Spotify para RecSys challenge 2018

- El siguiente paper describe el RecSys challenge 2018, los equipos que enviaron y sus repos, datasets y métricas

- https://arxiv.org/pdf/1810.01520.pdf

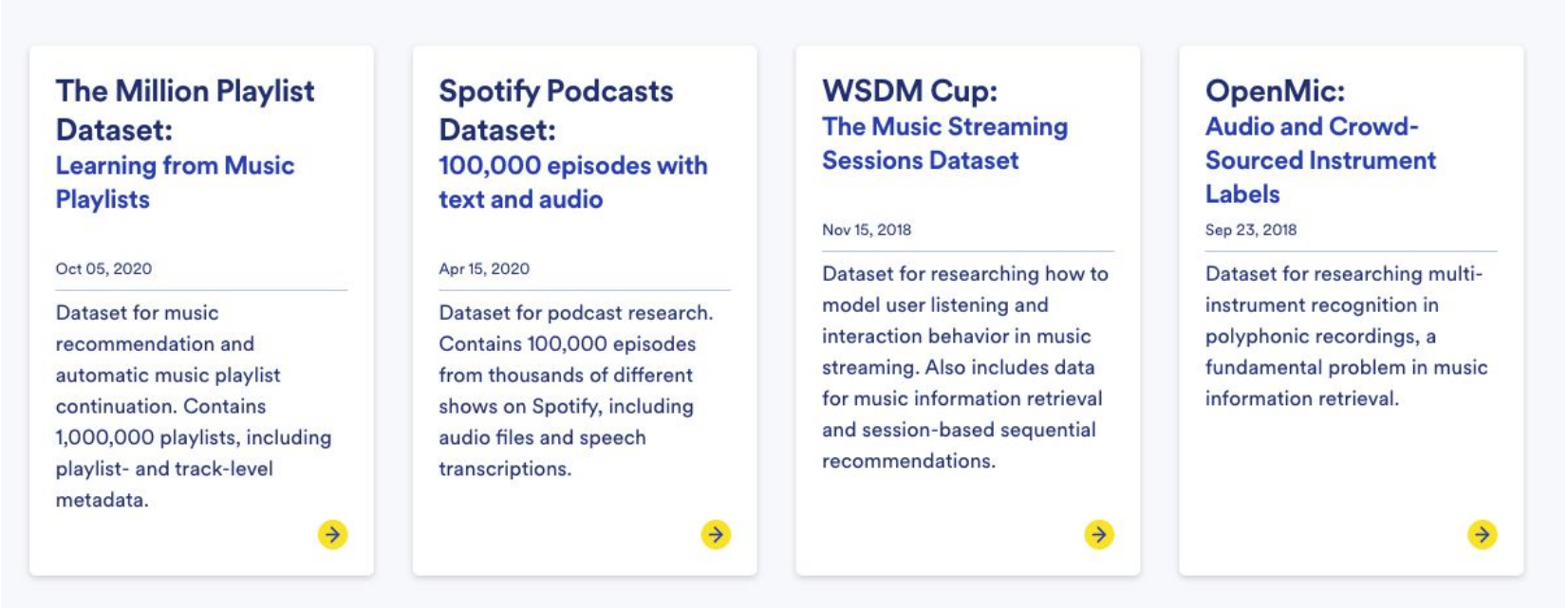
Table 1. Basic statistics of the Million Playlist Dataset.

Property	Value
Number of playlists	1,000,000
Number of tracks	66,346,428
Number of unique tracks	2,262,292
Number of unique albums	734,684
Number of unique artists	295,860
Number of unique playlist titles	92,944
Number of unique normalized playlist titles	17,381
Average playlist length (tracks)	66.35

- Idea de proyecto: Investigar una forma novedosa de recomendar con este mismo dataset

Otros datasets de spotify

https://research.atspotify.com/datasets/



Reproducir un paper

- No todos los paper publicados tienen una implementación pública
- Una muy buena idea de proyecto es implementar el modelo del paper ver si es posible replicar resultados
- Posteriormente, si la replicación fue posible, testear lo mismo pero con otros datasets para ver si el resultado es consistente (reproducible)

Ejemplos: DVBPR y Attentive Collaborative Filtering para recomendación de imágenes

Datasets

Kaggle

https://www.kaggle.com/search?=recommender+in+%3Adatasets+sortBy%3Adate

Julian Macaulay

https://cseweb.ucsd.edu/~jmcauley/datasets.html

MeLi Data Challenge 2020

Mercado Livre Data Challenge 2020

https://www.kaggle.com/datasets/marlesson/meli-data-challenge-2020


```
{'user_history': [
        {'event_info': 2443411,
         'event_timestamp': '2019-09-27T10:43:47.778-0400',
         'event_type': 'view'},

        {'event_info': 2443411,
         'event_timestamp': '2019-09-27T10:47:21.195-0400',
         'event_type': 'view'},

        {'event_info': 'NUMEROS RESIDENCIAIS INOX',
         'event_timestamp': '2019-09-27T10:47:43.019-0400',
         'event_type': 'search'},

        {'event_info': 'NUMEROS RESIDENCIAIS INOX 2',
         'event_timestamp': '2019-09-27T10:48:45.530-0400',
         'event_type': 'search'},

        {'event_info': 522000,
         'event_timestamp': '2019-09-27T10:49:19.537-0400',
         'event_type': 'view'}
    ],
'item_bought': 2659106}
```

Recommendation of COVID-19 articles using Deep Knowledge-Aware Network

Ivania Donoso-Guzmán
indonoso@uc.cl
Pontificia Universidad Católica de Chile
Santiago, Chile

ABSTRACT

In the last few years the number of papers that are published every day has increased enormously. Keeping up with the current state of the art has become a very difficult task for researchers. A recent work presented the use of Deep Knowledge-Aware network for paper recommendation. In this work, we expand its results by using SciBERT, BioBERT and Word2Vec as word embedding models and MeSH, MAG, and L-OVE as knowledge graphs to create the entity embeddings. Our results show that the knowledge graph does not influence the results, but the word embedding model can affect the model outcome.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

of information makes it difficult for researchers to keep up with the latest articles in their fields.

There have been efforts to create recommendation systems that could help researchers find relevant works more efficiently. These systems have mostly relied on interactions with applications (likes, bookmarks, saves) [10] and normally use a content based approach or a knowledge base graph that relies on author-paper relations. Most recently the tutorial [11] presented the use of Deep Knowledge-Aware Network (DKN) [16] for paper recommendation.

In this work we aim to expand the tutorial [11] by using different knowledge graphs, to create the entity embeddings, and also use different models to create the word embeddings. We want to understand what characteristic should have the embeddings for the tasks of user-item and item-item recommendation. The research questions can be summarized as:

Matrix Factorization–User KNN Ensemble Approach for Automatic Playlist Continuation

Thomas Muñoz
Pontificia Universidad Católica de Chile
Santiago, Chile
tfmunoz@uc.cl

Rodolfo Palma
Pontificia Universidad Católica de Chile
Santiago, Chile
rdpalma@uc.cl

ABSTRACT

Automatic playlist continuation is a contemporary problem in music consumption. Since playlists elaboration is more time consuming for the user, playlist continuation is an uprising way of exploring music. Due to this importance, the ACM Recommender Systems Challenge 2018 was focused on evaluating automatic playlists continuation systems, using the Million Playlist Dataset provided by Spotify. In this paper we present a two-stage model towards computing recommendations for a subset of playlists of the dataset. We first focus on a fast retrieval stage that aims to reduce the candidate songs to be searched in the second stage, which reranks these tracks. The final recommendations are an ensemble of this two systems. This simple ensemble system would have reached the 30th place in the challenge main track out of over 100 teams.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

eficientemente el tamaño del espacio de búsqueda de canciones relevantes. La segunda etapa consiste en un algoritmo de vecinos cercanos, cuyo tiempo de ejecución se ve drásticamente reducido por el filtrado de la primera etapa.

Luego de ejecutar diversos experimentos, se puede observar que el modelo propuesto mejora significativamente métricas, como NDCG, en comparación al *baseline most popular*. Además, se considera que el modelo es competitivo, debido a que si se hubiese enviado al *ACM RecSys Challenge 2018* hubiese obtenido el lugar N°30 en el *track* principal.

2. *AUTOMATIC PLAYLIST CONTINUATION*

A continuación se definirá la tarea sobre la cuál se aplicará el sistema recomendador. En primer lugar, se define una lista de reproducción como una secuencia ordenada de canciones con el propósito de ser escuchadas en conjunto por algún usuario. La tarea de continuar automáticamente una lista de reproducción (APC, por su sigla en inglés) consiste en ir agregando una o más canciones a una lista