UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Implementación de un modelo de aprendizaje automático para la recomendación de ítems.

TRABAJO DE TITULACIÓN

AUTOR: Ojeda Ureña, Carlos Francisco.

DIRECTORA: Valdiviezo Díaz, Priscila Marisela, Mgs.

LOJA – ECUADOR

2017

Septiembre, 2017

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magíster.

Valdiviezo Díaz Priscila Marisela.

DOCENTE DE TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Implementación de un modelo de aprendizaje automático para la recomendación de ítems realizado por Ojeda Ureña Carlos Francisco, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, Marzo de 2017

f)……

Mgs. Priscila Marisela Valdiviezo Díaz

DECLARACIÓN DE AUDITORÍA Y CESIÓN DE DERECHOS

"Yo Ojeda Ureña Carlos Francisco declaro ser autor (a) del presente trabajo de titulación: **Implementación de un modelo de aprendizaje automático para la recomendación de ítems**, de la Titulación de Sistemas Informáticos y Computación, siendo Priscila Marisela Valdiviezo Díaz directora del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad"**.**

f).....

Autor: Ojeda Ureña Carlos Francisco

Cedula: 1105403024

DEDICATORIA

Con inmenso amor dedico este proyecto a mis padres Carlos Ojeda y Carmen Ureña, quienes con su esfuerzo, sacrificio, confianza e infinito amor han permitido que llegue a concluir con esta etapa de mi vida y por ser el motor para levantarme y seguir adelante.

A mis hermanos Jefferson, Jean y Yajaira quienes me han impulsado a lograr mis metas y por ser un ejemplo a seguir. A mis sobrinos Marlon y Carlos quienes han alegrado mi vida desde el inicio de su existencia.

Este título va con mucho cariño para ustedes, por estar presente en todo momento a lo largo de mi carrera profesional. A parte de la vida son lo mejor que Dios me regalo, los amo con todo mi ser.

Ojeda Ureña Carlos Francisco

AGRADECIMIENTO

Agradezco primeramente a Dios, por regalarme la vida y ser mi guía en todo momento para forjar mi futuro.

A mis padres y hermanos por confiar en mí, y por la paciencia que han tenido en ver cumplida esta meta. Mil gracias, Dios les pague y les múltiple todo lo que han hecho por mí.

A mi tutora de Tesis Magíster. Valdiviezo Díaz Priscila Marisela quien con sus conocimientos y experiencia ha sabido guiarme para culminar el presente trabajo de titulación.

A la titulación de Sistemas Informáticos y Computación, y a mis profesores quienes me impartieron sus conocimientos en el transcurso de la carrera.

Muchas gracias a todos, por ser parte de mi formación profesional.

Ojeda Ureña Carlos Francisco

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

RESUMEN

El presente trabajo de fin de titulación presenta una implementación de un sistema de recomendación de ítems. Se emplea el conjunto de datos de MovieLens que contiene películas se hace referencia a las películas como ítems. Se extrae información de contenido de las películas, para extraer características de las películas y recomendar películas similares. Se aplica una serie de pre-procesamientos a la información de contenido de las películas. Seguido se extrae tópicos mediante una técnica de aprendizaje automático contemplada dentro de los modelos probabilísticos denominada Latent Dirichlet Allocation (LDA). Almacenando la distribución de tópicos para cada película.

PALABRAS CLAVE: sistemas de recomendación, Latent Dirichlet Allocation, aprendizaje automático.

ABSTRACT

The present work of finalizing the presentation presents an implementation of a system of recommendation of items. The MovieLens data set that contains movies is used to refer to movies as items. Content information is extracted from the movies, for more features of the movies and recommendations. A series of pre-processing is applied to the content information of the films. Follow-up of the application of an automated learning technique within the probabilistic models called Dirichlet Latent Assignment (LDA). Storing the distribution of topics for each movie.

KEYWORDS: system recommender, Latent Dirichlet Allocation, maching learnig.

CAPITULO I: VISIÓN DEL PROYECTO

El presente capítulo brinda una visión general del proyecto, los objetivos que se plantean y los resultados que se persiguen; además de brindar detalle de la estructura del presente trabajo de titulación.

1.1. Introducción

Los sistemas de recomendación (en adelante SR) actúan como un filtro a la información que perciben los usuarios de internet, intentando predecir el impacto que un ítem tendría sobre estos usuarios. Nacen como respuesta a la sobre carga de información que sufren los usuarios de sistemas de servicios en línea como: Netflix, MovielLens, Amazon, YouTube, Facebook, entre otros. Éstos sistemas integran diferentes técnicas de Filtrado para limitar la cantidad de información que reciben sus usuarios.

Los diferentes tipos de filtrado que brindan los sistemas de recomendación poseen ventajas y desventajas, una tendencia que ha impactado es la combinación de métodos de filtrado. Mediante constante investigación en el campo de Machine Learning (en adelante ML), diferentes técnicas se implementan para mejorar la calidad de las recomendaciones que reciben los usuarios dentro de un SR; debido a que las técnicas que aporta el ML intentan plasmar la realidad mediante técnicas estadísticas; los modelo probabilísticos del ML están en auge combinándolos para mejorar los SR.

SR son concebidos para mitigar el impacto negativo de la sobrecarga de información que se produce en internet, mediante técnicas de filtrado. Brindando disponibilidad a los usuarios sobre ítems atractivos y relevantes para ellos. Tomando información de los usuarios e ítems existentes en el SR. Éstos implementan técnicas que permiten personalizar las necesidades e intereses de un usuario en particular o generalizar a comunidades de usuarios.

El presente trabajo de titulación se analiza técnicas probabilísticas de recomendación para implementar un modelo de recomendación basado en la información de ítems.

1.2. Problemática y justificación

Los métodos de filtrado tradicional enfocan las recomendaciones basándose en la similitud de gustos de los usuarios registrados en el sistema de servicios en línea; sin tomar el contenido que caracterizan a los ítems que los usuarios califican (en adelante se utiliza el término califican, valoración y ranking; haciendo alusión al mismo significado). Obteniendo como resultado recomendaciones que no plasma los gustos del usuario a quien se emite una recomendación de parte del SR. Dentro de este punto la investigación se centra en implementar un modelo que el contenido que caracteriza a los ítems.

Los Sistemas de Recomendación basados en Filtrado Colaborativo poseen un amplio campo de investigación e implementación; pero presentan varias desventajas como:

- El espacio latente aprendido no es fácil de interpretar, para los usuarios por lo cual un usuario no comprende porque se le recomienda un ítem.
- La Matriz de Factores solo utiliza información de los usuarios plasmado mediante los rankings a ítems, y no se puede generalizar a ítems sin calificación de parte de los usuarios; este último es de vital importancia para los sistemas actuales debido a la incorporación constante de nuevos ítems en el sistema(Peralta Costoya, 2013).

Los datos a emplear en el desarrollo del presente trabajo de titulación con el modelo a seleccionar se enmarcan dentro de un servicio de alquiler de películas. Para lo cual se toma el conjunto de datos MovieLens. Se realiza una serie de experimentos con estos datos debido a que este conjunto de datos; es utilizado ampliamente en el campo de estudio de los SR.

Las diferentes técnicas disponibles para trabajar con información de contenido y descubrir representaciones latentes en el contenido de los ítems se desarrolla con algoritmos de Machine Learning, dentro de este campo se trabaja con modelos probabilísticos como Probabilistic Topic Models una de sus variantes amplia mente investigada e implementa es Latent Dirichlet Allocation (en adelante LDA). Los resultados obtenidos con el modelo LDA son parte de una aproximación hibrida denominada Collaborative Topic Regression (CTR) que se está investigando dentro del grupo de Inteligencia Artificial de la UTPL, esta aproximación combina métodos basados en Matriz de Factores como lo es el método Probabilistic Matrix Factorization (en adelante PMF), con LDA. La aproximación hibrida denominada CTR es capaz de cubrir las ventajas del Filtrado Colaborativo tradicional trabajando con la inferencia de variacional para situar la información de contenido de los ítems en función de tópicos. Para brindar recomendación dentro y fuera de la matriz de predicciones cubriendo las desventajas del FC tradicional combinándolo con el Filtrado basado en contenido con la técnica LDA.

1.3. Objetivos

Ésta sección detalla los objetivos específicos como general del presente trabajo de titulación:

Objetivo general:

- Implementar un modelo de aprendizaje automático para la recomendación de ítems.

Objetivos específicos:

- Investigar parámetros y técnicas utilizadas en el proceso de recomendación .
- Analizar y seleccionar técnicas que mejor se ajusten a un modelo de recomendación.
- Caracterizar parámetros del modelo de recomendación a implementar.

- Experimentar con algoritmos de aprendizaje automático.
- Analizar los resultados obtenidos de algoritmos de aprendizaje automático.
- Implementar y evaluar el modelo seleccionado.

1.4. Proceso para el desarrollo del trabajo de titulación

El proceso a seguir para la ejecución del presente trabajo de titulación se detalla a continuación:

- Marco conceptual y estado del arte en sistemas de recomendación, aprendizaje automático, clarificando conceptos clave y trabajos relaciones.
- Experimentación con técnicas de aprendizaje automático investigadas con la finalidad de comprender los modelos, y evaluar con datos experimentales de una web de alquiler de películas como MovieLens, cuantificando la calidad de las recomendaciones de las técnicas a evaluar.
- Selección de técnica que presente mejores resultados en las recomendaciones a usuarios contemplando a usuarios e ítems.

1.5. Estructura del documento

El presente trabajo de titulación sigue la siguiente estructura:

- En el capítulo I detalla la visión general del proyecto.
- El capítulo II brinda un marco conceptual y estado del arte en sistemas de recomendación, aprendizaje automático y métricas de evaluación de sistemas de recomendación.
- En el capítulo III presenta la implementación de un sistema de recomendación utilizando la técnica Topic Model con la variante LDA, utilizando el conjunto de datos de MovieLens dentro del lenguaje de programación R.
- En el capítulo IV presenta el análisis de resultados de la implementación del modelo LDA con los parámetros que caracterizan el modelo.

CAPITULO II: ESTADO DEL ARTE

El presente capítulo detalla conceptos de sistemas de recomendación, técnicas de aprendizaje automático y métricas de evaluación de recomendaciones, así como también, se describe la estructura correspondiente a las técnicas, algoritmos y algunos conceptos que permitan clarificar el presente trabajo de titulación. Se realiza la descripción de algunos trabajos relacionados que mantienen similitud con este trabajo, con el fin de tener una referencia para la selección de la técnica a implementar.

2.1. Aprendizaje automático

2.1.1 Aprendizaje automático supervisado

Uno de los puntos fuertes del aprendizaje está ligado al grado de supervisión que percibe, pudiendo ser brindado por un experto en el dominio de los datos, dotando al aprendiz (puede ser un algoritmo) de la retroalimentación sobre una base de conocimiento previo que es importante para el aprendizaje del aprendiz. Este tipo de aprendizaje se implementa cuando se desea realizar predicciones, donde cada categoría o clase están predefinidas, de tal manera que cada ejemplo se asocie con una clase o categoría. Este tipo de aprendizaje se denomina supervisado por la presencia de clases que encaminan el aprendizaje en el conjunto de entrenamiento, se menciona algunos algoritmos de aprendizaje supervisado como Bayes, Support Vector Machine, entre otros. Como método de comprobación se define un Gold Standard, que será utilizado para hacer una comparación clara del modelo, se procederá a realizar pruebas para constatar cómo fueron clasificados los datos, estos se pueden clasificar como Correctamente Clasificado (por sus siglas en Ingles TP), Erróneamente Clasificado (FP), con los valores presentados tanto de TP como de FP será evaluado el modelo, mediante métricas como Recall, Precisión, etc.(Alvarado, 2015).

2.1.2 Aprendizaje automático no supervisado (o algoritmos de descubrimiento de conocimiento)

En el método de aprendizaje no supervisado, la retroalimentación de parte de un experto en el conjunto de datos, no está disponible. Este tipo de aprendizaje es implementado para extracción de información que contenga un grado de utilidad partiendo de colecciones de datos. En este tipo de aprendizaje también se crea un Gold Standard, pero no siempre es posible realizar una evaluación del conjunto de datos, debido a que no se conoce a priori el número de clases o categorías, en los cuales se va a clasificar los datos. Razón por la cual el desempeño del modelo debe ser necesariamente evaluado a posteriori con un experto en el dominio de los datos de forma manual con los resultados presentados por el modelo(Alvarado, 2015).

2.2. Sistemas de recomendación

Los sistemas en línea orientados a ofertar productos y/o servicios han crecido en gran escala creando un problema de sobre carga de información a sus usuarios. Éste problema no se limita a un conjunto concreto de sitios, sino en todo internet. Como respuesta a éste problema nacen los sistemas de recomendación los cuales son sistemas inteligentes capaces de realizar recomendaciones actuando como filtros a la información que recibe el usuario, la información se puede personalizar atendiendo a los gustos de cada usuario. Para lo cual necesitan conocer los gustos de los usuarios partiendo de éstos realizan predicciones sobre posibles ítems que contengan información que al usuario le interesa o guste. Los gustos de cada usuario son plasmados con rankings a ítems existentes dentro del sistema. El tipo de ítems a recomendar por los SR es muy variado dependiendo del tipo del proveedor de productos y/o servicios se cita algunos de los más conocidos como: Netflix, Facebook, YouTube y MovieLens.

La tarea de averiguar los gustos de los usuarios y encontrar aquellos ítems que se ajusten mejor a los gustos requiere el uso de técnicas de Machine Learning (o aprendizaje automático en español). Dentro del campo del ML, encontramos los modelos probabilístico que son los que mejor se adaptan a este caso en particular de determinar los gustos en basa a distribuciones de probabilidades (Ortega Requena, 2015).

El constante cambio en los sistemas dentro de internet ha forzado un constante proceso de investigación y evolución de los sistemas de recomendación en los últimos años. Donde en un inicio se toma la información de forma explícita (o proporcionada de forma manual por los usuarios), a tenerla de manera implícita (por medio de valoración de usuarios con gustos similares). A incorporar información de redes sociales e internet de las cosas. La finalidad es mejorar la calidad de las recomendaciones tratando de plasmar la realidad lo más fielmente posible en las recomendaciones que reciben los usuarios. Sacrificando la universalidad de los métodos y aumentando el coste computacional de los algoritmos (Fernando Ortega, 2015).

Los SR actúan como filtro de la información que perciben los usuarios, estos intentan predecir el impacto que tendrá un ítem sobre un usuario, al cual se emite una recomendación. Además los SR ayudan a interactuar a los usuarios dentro del sistema. Evitando aislar a los usuarios, mejorando la interacción con el SR (Fernando Ortega, 2013).

2.3. Clasificación de los sistemas de recomendación

Los sistemas de recomendaciones tienen una clasificación bastante amplia dependiendo del contexto donde se utilizan a continuación se presentan tres tipos de filtrado los cuales poseen un campo de investigación en constante evolución como son:

2.3.1. Filtrado colaborativo (en adelante FC)

Los SR que implementan FC parten de una matriz de rankings que se crea con información explicita de los usuarios por medio de los rankings emitido sobre los ítems existentes en el sistema que han gustado al usuario. Las recomendaciones producidas plasman las preferencias del usuario. Este método es el más preciso entre los distintos tipos de filtrado existentes. Dando como resultado una área de investigación en constante evolución. Aunque tiene un alto costo computacional para poner en funcionamiento. Él FC toma la idea, de cuando una persona desea una recomendación sobre algún producto o servicio le solicita a otra persona con gustos similares (inferidos) le brinde una recomendación en base a los gustos en común que ambas personas poseen. La fortaleza de éste método es la participación de los usuarios de ahí nace la idea del filtrado colaborativo.(Fernando Ortega, 2015; Peralta Costoya, 2013). La Tabla 1 presenta un ejemplo de una matriz de rankings de usuarios hacia ítems existentes es el sistema.

Tabla 1. Matriz de rankings usuarios/ítems.

Usuarios	**Ítems**			
	Ítem 1	Ítem 2	Ítem 3	Ítem 4
Usuario1	4	3	0	1
Usuario 2	0	0	1	4
Usuario 3	5	5	0	0
Usuario 4	4	4	5	3

Elaborado por: Autor.
Fuente:(Moya García, 2015).

La Tabla 1 presenta una matriz de rankings donde se tiene en las columnas ítems y en sus filas usuarios produciendo en sus intersecciones entre una columna y una fila esta la calificación que un usuario a emitido sobre un ítem determinado. Partiendo de esta matriz el filtrado colaborativo puede predecir recomendaciones en base a los gustos similares que poseen los usuarios.

Problemas del filtrado colaborativo:

El filtrado colaborativo presenta algunas ventajas y desventajas que se exponen a continuación:

Ventajas:

- El contenido no importa, solo toma los rankings de otros usuarios, haciendo una ventaja este aspecto.

- Puede aplicarse a cualquier tipo de ítem o producto(Pablo Castells, Fernando Díez, 2011).
- Permite introducir novedad, previa a la experiencia del usuario (Pablo Castells, Fernando Díez, 2011).
- Similitud a la popularidad global, pero personalizando a cada usuario(Pablo Castells, Fernando Díez, 2011).

Desventajas:

- Escasez (Sparsity): Es un problema común en el FC debido a que no todos los usuarios califican los ítems disponibles en el sistema. Dificultando calcular la similitud entre usuarios e ítems dificultando la tarea de predecir recomendaciones.
- Arranque en frio (Cold start): El problema de arranque en frio se produce por la existencia de un nuevo usuario o ítem en el sistema al ser nuevos estos no han emitido rankings en el caso del usuario y los ítems no han tenido rankings por los usuarios al relativamente nuevo en el sistema (Peralta Costoya, 2013).
- Transparencia (Transparency): El procesos de recomendación no es transparente para el usuario no se puede explicar la recomendación.

La clasificación del filtrado colaborativo se realiza atendiendo a como se obtienen las recomendaciones estas pueden ser de dos tipos como se describen en la sección 2.3.1.1 y 2.3.1.2.

2.3.1.1. Filtrado colaborativo basado en memoria

Las técnicas de FC basado en memoria toman la matriz de rankings de los usuarios a los ítems disponibles en el sistema para predecir rankings utilizando heurísticas computando todo en memoria(Peralta Costoya, 2013).

Uno de los algoritmos con mayor implementación es el método KNN (k-vecinos cercanos o en inglés k-nearest neighbors) en el FC basado en memoria. Que parte de calcular la similitud (inferir) entre usuarios con gustos similares al usuario a recomendar. Eligiendo los K vecinos más cercanos. Realizando predicciones sobre los rankings que emitieron los vecinos del usuario a recomendar para completar los espacios vacíos en la matriz de rankings. Partiendo de una matriz completa de rankings infiriendo rankings emite recomendaciones de los n ítems más prometedores al usuario a brindar una recomendación de parte del SR (Hernando, Moya, Ortega, & Bobadilla, 2013; Peralta Costoya, 2013)

La principal desventaja que presenta el método de vecindad KNN es condicionar los resultados de los SR, además presentan otros problemas como balance entre exactitud y

escalabilidad, que mientras más vecinos k se consideren mejor debería ser la predicción de los rankings; pero entre más usuarios n existen, mayor es el costo de encontrar los k vecinos más cercanos. Con lo cual lo hace muy poco sustentable en sitio con millones de usuarios registrados. La dispersión es otro problema la cual es producto de la baja densidad de los datos en el sistema. Los modelos basados en factorización matricial (en inglés Matrix Factorization) están en auge en la actualidad mitigando los problemas del método de KNN (Fernando Ortega, 2015; Parra, 2015c).

2.3.1.2 Filtrado colaborativo basado en modelo

El FC basado en modelo toma un modelo de la matriz de rankings para calcular la probabilidad de que un nuevo ítem sea interesante para un usuario al cual se emite una recomendación de parte del sistema (Peralta Costoya, 2013).

Existe una amplia variedad de técnicas para este tipo de filtrado, pero la técnica que ha presentado mejores resultados es la factorización matricial. La idea tras éste modelo es la existencia de factores latentes u ocultos, que permiten hacer predicciones de los rankings, además permiten obtener información de relaciones entre usuarios e ítems. Brindando buena escalabilidad al momento de predecir rankings, volviendo las predicciones más precisas y flexibles(Fernando Ortega, 2013; Ortega Requena, 2015; Parra, 2015a).

Los modelos Factorización Matricial (en adelante MF por sus siglas en inglés) como su nombre lo indica factorizan o dividen la matriz de rankings en dos nuevas matrices estas matrices resultantes son:

- La primera matriz resultante es denomina Matriz Usuario/Tema (o Tópico) la cual contiene la distribución de probabilidad de los usuarios en los temas.
- La segunda matriz denominada Matriz Ítem/Tema esta contiene la distribución de probabilidad de los ítems en los temas.

Para clarificar el concepto de Factorización Matricial la Figura 1 presenta como se factoriza la matriz de rankings en dos nuevas matrices.

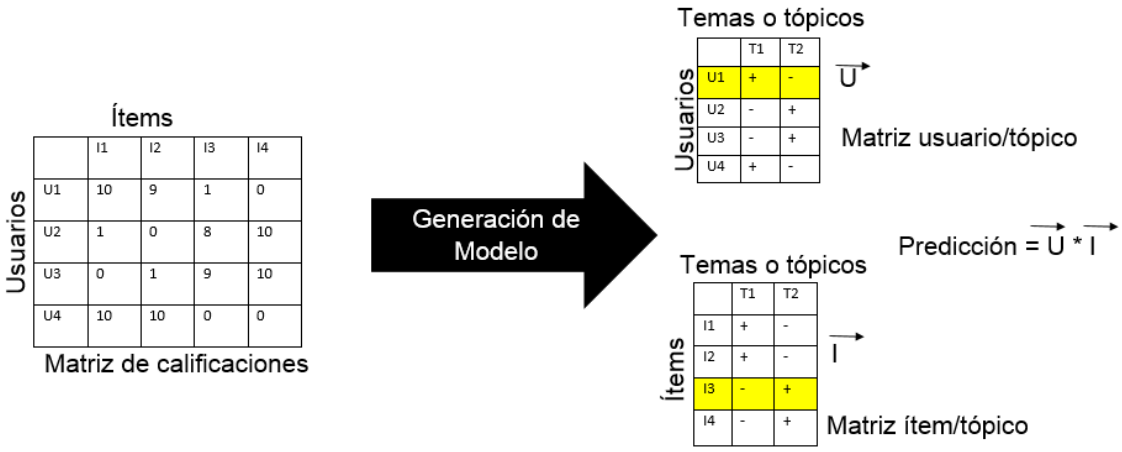


Figura 1. Matriz de rankings factorizada en dos nuevas matrices.
Fuente:(Fernando Ortega, 2013).

La Figura 1 presenta como el método de factorización matricial crea dos nuevas matrices para predecir los valores faltantes en la matriz.

En los siguientes apartados se exponen algunas técnicas que se basan en la técnica de factorización matricial:

- **Singular Value Decomposition (SVD)**

SVD es una técnica con un campo de investigación amplio por lo que se ha implementado y evaluado, mostrando buenos resultados. La finalidad que persigue es obtener una serie de factores latentes que caractericen a los usuarios e ítems. Tomando como punto de partida los factores latentes para calcular la similitud entre usuarios o ítems.(Hernando et al., 2013)

- **Probabilistic Matrix Factorization (PMF)**

El principio tras utilizar modelos de factores latentes en la MF, es que las preferencias del usuario se determinan por un pequeño número de factores latentes o no observados. La finalidad es descubrir la atracción de un usuario hacia las características ocultas de los ítems que expliquen la valoración observada hacia estos ítems de parte de los usuarios a la cual se la denomina rankings, calificación o valoración. La denominación de ranking se denota por **"R"**. El usuario se identifica como **"i"**, este se representa por un vector latente $\boldsymbol{u_i} \in \boldsymbol{R^k}$ y el ítem, se identifica como **"j"**, que es representado como un vector latente $\boldsymbol{v_i} \in \boldsymbol{R^k}$. **"K"** es selecciona cumpliendo la siguiente condición $\boldsymbol{k \ll i,j}$**.** La denominación rankings $\boldsymbol{r_{ij}}$**,** está ligada por el producto interno del par de los vectores latentes, que se presenta en la ecuación 1.(Chong Wang, 2011; Orii, 2012; Wu et al., 2016)

Ecuación 1. Cálculo de vectores latentes.

$$r_{ij} = u_i^T v_j$$

Fuente:(Wu et al., 2016).

Partiendo de rankings **"R"**, el problema es calcular los vectores de características latentes **"u"** y **"v"**. Lo cual consiste en regular el error cuadrático minimizándolo.

La Figura 2 presenta el modelo grafico de la técnica Probabilistic Matrix Factorization, representando como se calcula los vectores de características **"u" y "v"**.

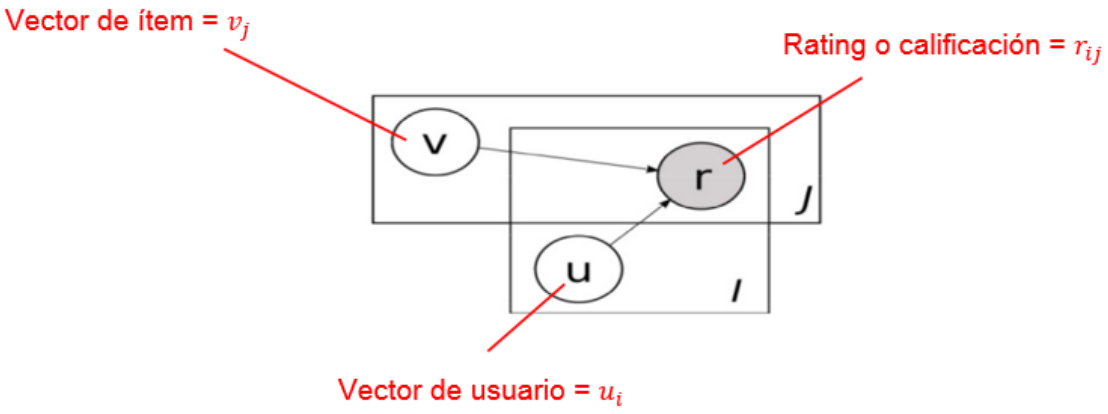


Figura 2. Modelo grafico de la técnica Probabilistic Matrix Factorization.
Fuente:("A Probabilistic Approach for Recommendation Looking at Collaborative Topic Regression as introduced," n.d.).

El modelo grafico probabilista de la técnica Probabilistic Matrix Factorization, se presenta en la Figura 3, como se calculan los vectores de características **"V_j" y "V_i".**

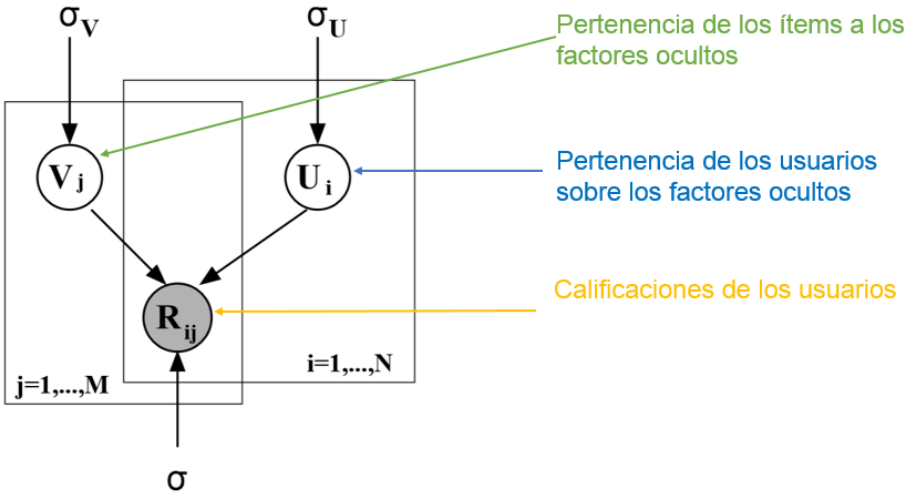


Figura 3. Modelo grafico probabilista de la técnica Probabilistic Matrix Factorization.
Fuente:(Fernando Ortega, 2013).

La Figura 4 presenta como se conforman los vectores de características tanto de Ítems como de rankings.

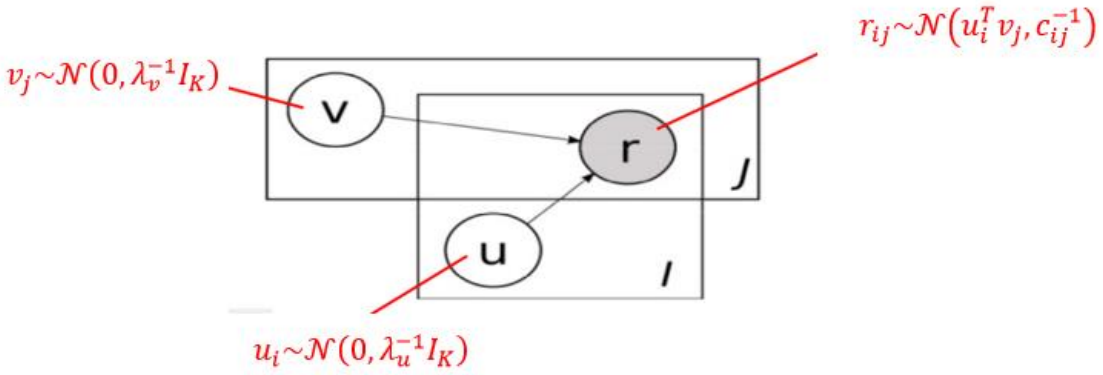


Figura 4. Modelo Probabilistic Matrix Factorization utilizando vectores.
Fuente:("A Probabilistic Approach for Recommendation Looking at Collaborative Topic Regression as introduced," n.d.).

En la técnica Probabilistic Matrix Factorization se modifica λ_u y λ_v, estos son parámetros regulables. Se puede tener un enfoque probabilístico en MF. Utilizando un modelo generativo simple usando un modelo lineal probabilista con el modelo Gaussiano de la siguiente manera(Orii, 2012; Wu et al., 2016):

- Para cada usuario **"i"**, modelar un vector latentes para el usuario $\boldsymbol{u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)}$
- Para cada ítem **"j"**, modelar un vector latentes para el ítem $\boldsymbol{v_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)}$
- Para cada par de usuario-ítem **("i", "j")**, modelar la calificación $\boldsymbol{r_{ij} \sim \mathcal{N}(u_i^T v_j,\ c_{ij}^{-1})}$

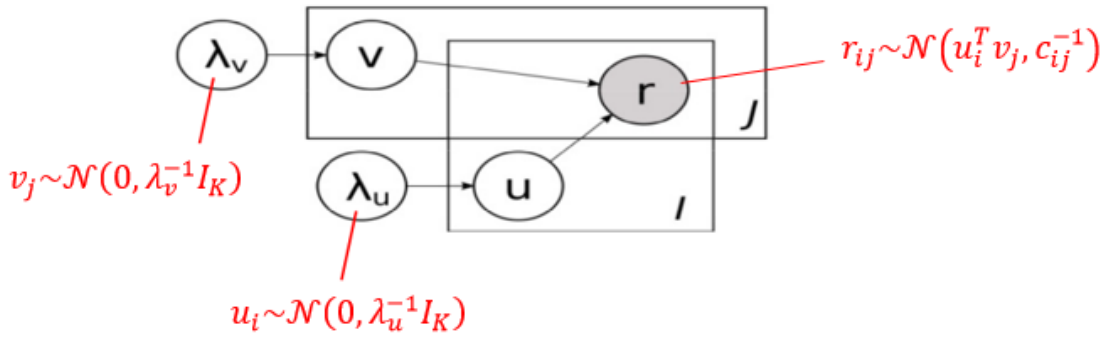


Figura 5. Modelo grafico Probabilistic Matrix Factorization utilizando parámetros regulables.
Fuente:("A Probabilistic Approach for Recommendation Looking at Collaborative Topic Regression as introduced," n.d.).

Donde $\boldsymbol{I_K}$, es una matriz identidad k-dimensionalidad, y "$\boldsymbol{c_{ij}}$", mide la confianza observada en $\boldsymbol{r_{ij}}$. El usuario **"i"** está interesado en el ítem **"j"** cuando $\boldsymbol{r_{ij}} = 1$, pero no le

interesa cunado $\boldsymbol{r_{ij}} = 0$. En consecuencia, se utiliza diferentes valores para $\boldsymbol{c_{ij}}$ en función del valor $\boldsymbol{r_{ij}}$ como se presenta(Orii, 2012):

$$c_{ij} = \begin{cases} a\ si\ r_{ij} = 1 \\ b\ si\ r_{ij} = 0 \end{cases}$$

Donde se debe cumplir la siguiente condición $\boldsymbol{a > b > 0}$, los parámetros **"a"** y **"b"** son ajustables, esto permite a la técnica PMF, trabajar con rankings observados o disponibles dentro de la matriz de rankings.

El Probabilistic Matrix Factorization trata de descomponer la matriz como en Singular Value Decomposition pero presenta dos diferencias principales estas son:

- Solo trabaja con los ítems con un valor de cero, por lo cual trabaja bien con una matriz escasa o poblada por ceros.
- Tiene una escala lineal con el número de observaciones.

(Chong Wang, 2011)Presenta dos desventajas principales:

- El espacio latente aprendido no es fácil de interpretar.
- Solo utiliza información de los usuarios, no generaliza hacia ha artículos no clasificados.

- **Collaborative topic regression (CTR)**

El modelo CTR mejora la técnica clásica de FC, que solo toma la información de retroalimentación de los usuarios hacia los ítems, que es la matriz rankings, incorporando la explotación de la información de contenido de los ítems logrando un desempeño prometedor en diferentes investigaciones como (Bhowmick, Prasad, & Kottur, 2014; Orii, 2012; Wu et al., 2016).

De manera similar que en LDA, en CTR cada ítem **"j"** es una proporción de tópico **"θ_j",** que es empleada para obtener palabras. Un enfoque ingenuo seria emplear **"θ_j"**, para representar el ítem como un vector latente en la ecuación(Orii, 2012):

Ecuación 2. Modelo CTR.

$$r_{ij} \sim \mathcal{N}\left(u_I^T \theta_j, c_{ij}^{-1}\right)$$

Fuente: (Wang & Blei, 2011).

Este enfoque no se toma en CTR en lugar del enfoque presentado en la Ecuación 2. Se explota la información del usuario para obtener un ítem ajustado con un vector latente **"v_j".**

La representación gráfica del modelo CTR, se ilustra en la Figura 7.

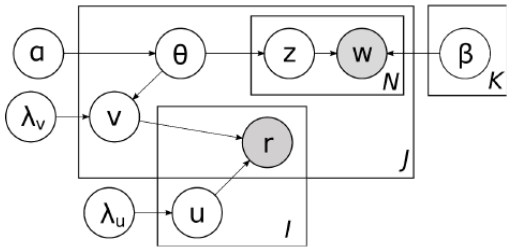


Figura 6. Representación gráfica del modelo CTR.
Fuente: (Orii, 2012).

CTR explota la información de contenido de los ítems para mejorar el filtrado colaborativo tradicional. El modelo CTR es un método que ha logrado un desempeño prometedor mediante la integración exitosa de la información de retroalimentación de los usuarios y la información de contenido que caracterizan a los ítems. El modelo CTR combina las ventajas del filtrado basado en contenido y filtrado colaborativo mediante la combinación de métodos probabilísticos como Probabilistic Matrix Factorization y Probabilistic Topic Model. Parte de los rankings de los usuarios a los ítems del sistema con la adición de información de contenido que caracterizan a los ítems mejorando sustancialmente la calidad de las recomendaciones que perciben los usuarios (Wu et al., 2016).

CTR representa a los usuarios con tópicos de interés y asumen que los ítems (documentos) con distribuciones de tópicos generados con LDA. Además CTR incluye una variable latente **"ϵ_j",** que es deducida de la distribución de tópicos **"θ_j"**, cuando modela los rankings de los usuarios. **"ϵ",** puede capturar el ítem preferido para un usuario en particular basándose en sus rankings hacia otros ítems. Esta es una innovación importante en comparativa con los modelos de PLSA, en donde la desviación respecto a la preferencia de los usuarios a la distribución tópico con el ítem no puede ser capturada (Wu et al., 2016).

La Figura 8 presenta el modelo de CTR y como incorpora la información de contenido combinando las ventajas del filtrado colaborativo con el filtrado basado en contenido en una aproximación hibrida de recomendación.

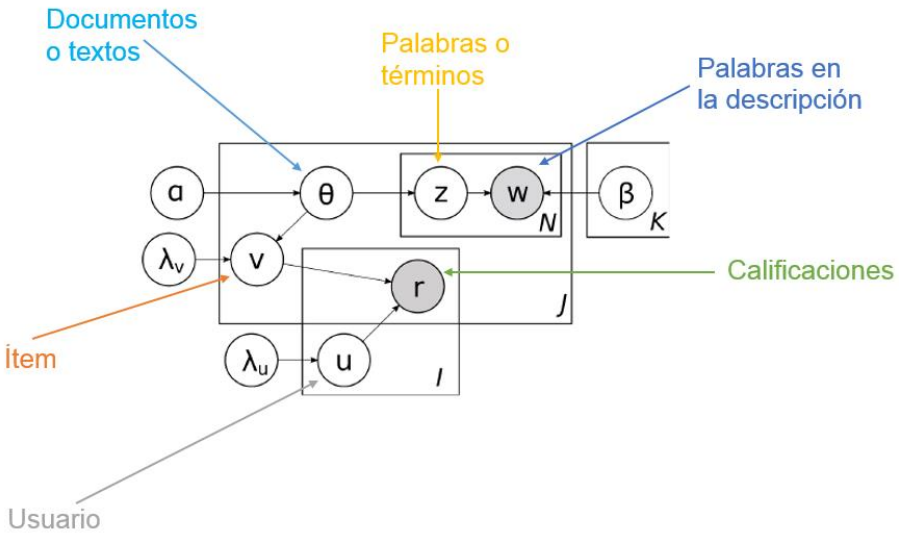


Figura 7. Modelo Collaborative topic regression explicado sus componentes.
Fuente:(Fernando Ortega, 2013).

La Figura 8 presenta todos los componentes del modelo hibrido de recomendación CTR que integra información de contenido de los items con los rankings.

2.3.2. Filtrado basado en contenido (en adelante FBC)

El FBC parte de la información de contenido de los ítems disponiendo de técnicas de aprendizaje automático para encontrar características en la información por medio de distribución de tópicos. Con las distribuciones de tópicos obtenidas se puede realizar recomendaciones de ítems con similitud. Los sistemas que implementa éste tipo de filtrado, surgen de la idea de recomendar partiendo de ítems que gustaron al usuario en el pasado por ejemplo: si al usuario X, le gusta la película de Transformers I, el sistema brinda recomendaciones como Transformes II. Partiendo de las características que ambas películas poseen en común en este caso en particular el guion de ambas películas es similar al tratarse de robots alienígenas (Peralta Costoya, 2013).

Se dispone de métodos filtrado basado en contenido como:

Term Frequency-Inverse Document Frequency (por sus siglas en ingles TF-IDF), signa un peso a cada término y/o palabra, midiendo de esta forma la similitud entre ítems como similitud entre pares de vectores. Clasificadores Bayesianos, como también técnicas de ML como Clustering, árboles de decisión y Redes Neuronales Artificiales. Éstos métodos se basan en identificar un modelo aprendido de la información subyacente y luego utilizar éste modelo para calcular probabilidades de si un ítem es atractivo para un usuario(Peralta Costoya, 2013).

Se han considerado Modelos Temáticos (Blei, 2012) que permiten describir la estructura semántica de una colección de documentos sobre la base de un análisis bayesiano jerárquico del texto en los documentos mediante un modelo probabilístico como Latent Dirichlet Allocation (Peralta Costoya, 2013).

El FBC parte de la matriz de apariciones de las palabras en los ítems (palabras/ítems) esta matriz se presenta en la Tabla 2, donde cada palabra está contenida en diferentes ítems.

Tabla 2. Matriz de palabras/ítems.

palabras	**Ítems**				
	Ítem1	Ítem2	Ítem3	Ítem4	Ítem5
baloncesto	3	0	2	0	0
política	0	0	4	4	2

Fuente:(Parra, 2015b).

Las recomendaciones presentadas a los usuarios por métodos de FBC, parten de la idea que si a un usuario le gusta el ítem1 presente en la Tabla2. Que contiene información sobre baloncesto le podría interesar el ítem3 que también posee información del mismo tipo.

El FBC presenta algunas ventajas y desventajas entre las que se encuentran:

Ventajas:

- Si los ítems poseen información de contenido suficientemente robusta, se evita el problema de arranque en frio.
- La representación del contenido es variado permitiendo emplear diversas técnicas de procesamiento de texto, uso de información semántica, inferencias, etc.
- El sistema provee transparencia, utilizando el contenido de los ítems para explicar las recomendaciones(Peralta Costoya, 2013).
- Recomendar ítems nuevos, que aún si no han sido valorados por ningún usuario.

Desventajas:

- Tiende a la sobre-especialización, recomendara ítems similares a los que ya han consumido los usuarios creando una tendencia al “filter bubble”, por lo que no hay novedad en las recomendaciones.
- Enfrentan problemas de semántica, polisemia y sinonimia(Peralta Costoya, 2013).
- Solo toma en consideración términos, no existe significancia entre un ítems bien redactado de otros(Peralta Costoya, 2013).

- Problema del nuevo usuario donde es necesario un número considerable de ítems previamente consumidos por el usuario para hacer una recomendación(Peralta Costoya, 2013).

El FBC representa el contenido de los ítems como bolsa de palabras; de esta forma se representa cada ítem (documento) como un Vector Space Model (en adelante VSM).La Figura 8 presenta la descripción grafica de bolsa se palabras.

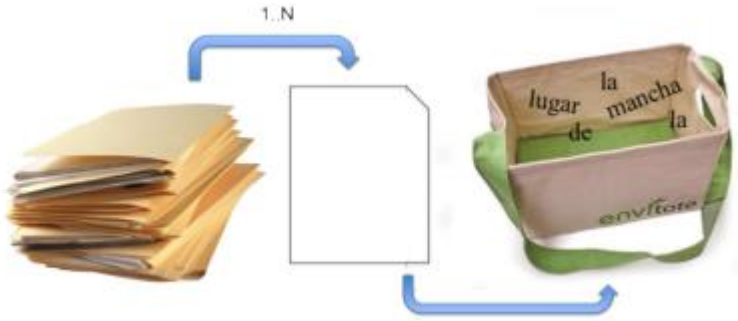


Figura 8. Descripción grafica de bolsa de palabras.
Fuente:(Parra, 2015b).

La representación del contenido VSM, como punto de partida se toma los ítems; estos contienen información de contenido de los ítems, esta colección de documentos se denomina corpus; este puede ser representado con una matriz donde las filas son términos y las columnas son documentos. Cada documento se representa como una vector, el peso de cada palabra para ese documento puede darse en base a la frecuencia del término en el documento. Puede darse el caso que un término aparezca en solo unos pocos documentos podría ser descriptivo, pudiendo considerarse la matriz inversa “Inverse Document Frequency” y combinarla con la matriz “Term Frequency” denominada TF-IDF.(Parra, 2015b)

Partiendo de la matriz se pueden realizar representaciones semánticas del contenido; existen diferentes opciones dentro del trabajo de fin titulación se enfoca a inferir representaciones semántica como (Latent semantic indexing, Latent dirichlet allocation).

2.3.2.1. Filtrado basado en contenido basado en memoria

En este tipo de filtrado existen diversas técnicas, estas técnicas incorporan métodos no probabilísticos del Machine Learning. A continuación se exponen algunas técnicas como:

- **Vectores de palabras**

Esta técnica realiza las recomendaciones basándose en el cálculo de distancias entre cada par de vectores de apariciones de palabras en los ítems. (Hernando et al., 2013)

Para lo cual parte de la matriz de palabras/ítems, que se presenta en la Tabla 3, y se calcula de la distancia Euclidiana entre los ítems el resultado se presenta en la Tabla 4.

Tabla 3. Matriz palabras/ítems.

palabras	ítems				
	ítem1	**ítem2**	**ítem3**	**ítem4**	**ítem5**
baloncesto	3	0	2	0	0
política	0	0	4	4	2

Fuente:(Moya Garcia, 2015).

Tabla 4. Calculo de distancia Euclidiana.

	ítem 1	**ítem 2**	**ítem 3**	**ítem 4**	**ítem 5**
ítem 1	0	3	4.1	5.0	3.6
ítem 2	3	0	4.5	4	2
ítem 3	4.1	4.5	0	2	2.8
ítem 4	5	4	2	0	2
ítem 5	3.6	2	2.8	2	0

Fuente: (Moya Garcia, 2015).

La calidad de las recomendaciones está ligada al cálculo de la distancia que existe entre cada par de vectores. Razón por que se debe buscar el método que mejor resultado presente en el cálculo de distancia entre vectores siendo esta su principal limitante.

- **Latent semantic indexing (LSI)**

También conocido como Latent Semantic Analysis (LSA), es una técnica de factorización matricial basada en la técnica matemática del SVD, parte de la matriz de apariciones de palabras/ítems. Para extraer una serie de factores latentes que caracterizan a las palabras y los documentos. Tomando como punto de inicio los factores latentes calculando la similitud entre los ítems o las palabras.(Hernando et al., 2013; Rus Marıa Mesas Javega, 2015)

LSI está ligado al correcto cálculo de similitud entre pares de vectores se debe probar que medidas de similitud presenta los mejores resultados.

2.3.2.2. Filtrado basado en contenido basado en modelo

Los sistemas de recomendación que implementan FBC basado en memoria utilizan técnicas probabilísticas del Machine Learning. A continuación se presentan algunas de las principales técnicas:

- **Probabilistic latent semantic indexing (en adelante PLSI)**

También conocido como Probabilistic Latent Semantic Analysis (PLSA), esta técnica a partir de este punto se denomina PLSI, es la evolución de la técnica no probabilística LSI a la cual se añade un modelo probabilístico. Dando como resultado la versión probabilística PLSI. Esta descompone la matriz de apariciones de palabras/ítems en dos matrices que van a tener un significado probabilístico. Los ítems y las palabras están caracterizadas por una distribución de probabilidad la cual indica el grado de pertenencia de un ítem hacia un tópico u otro. PLSI permite estudiar la similitud entre ítems, obteniendo la probabilidad que un documento pertenezca a determinado tópico.(González, 2013; Hernando et al., 2013)

PLSI basa su funcionamiento en Aspect model, el cual es un modelo de coocurrencia de datos que asocia una variable no observada a un tema, palabras o documento.

Utiliza las siguientes ecuaciones dependiendo la pertenencia del:

Conjunto de temas:

- Asociación de conjunto de temas. $z \epsilon Z = \{z_1, \dots, z_k\}$

Conjunto de palabras:

- Ecuación. Asociación conjunto de palabras. $w \epsilon W = \{w_1, \dots, w_n\}$

Conjunto de documentos:

- Asociación conjunto de documentos. $d \epsilon D = \{d_1, \dots, d_m\}$

La Figura 10 presenta el modelo grafico que sigue la técnica PLSI.

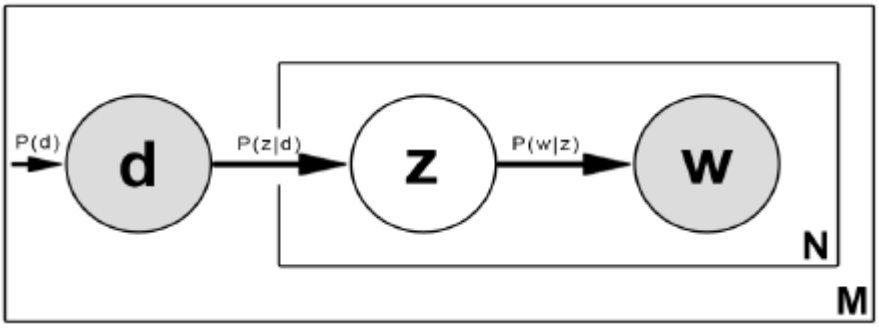


Figura 9. Modelo de PLSI.
Fuente:(Moya Garcia, 2015).

El modelo de PLSI, consta de los siguientes pasos:

- Seleccionar un documento $\boldsymbol{d}$ con probabilidad $\boldsymbol{P(d)}$
- Escoge una tópico latente $\boldsymbol{z}$ con probabilidad $\boldsymbol{P(z|d)}$
- Generar una palabra $\boldsymbol{w}$ con probabilidad $\boldsymbol{P(w|z)}$

Para calcular la probabilidad que se dé una coocurrencia de una palabra en un documento es necesaria la Ecuación 3.

Ecuación 3. Co-ocurrencia de una palabra dentro de un documento.

$$P(d|w) = P(d)P(w|d)$$

Fuente:(Ortega Requena, n.d.).

Donde la probabilidad de una palabra dentro de un documento se da por el desarrollo de la ecuación $P(w|d)$, en la Ecuación 4.

Ecuación 4. Probabilidad de término en un documento.

$$P(w|d) = \sum_{z \epsilon Z} P(w|z)P(z|d)$$

Fuente:(Moya Garcia, 2015).

El resultado de calcular la probabilidad de que en un determinado ítem los temas que lo conforman podría variar como se puede observar en la Figura 11 la probabilidad que en el ítem 1 se conforme de Política o Economía es 0 en ambos casos dando como resultado que en este determinado ítem el ítem dominante es Futbol. Pero se puede dar el caso que un ítem este conformado por más de un tema como es el caso del ítem 10 que está conformado por Futbol, Política y Economía en proporciones iguales del 0.33.

$P(z \mid d)$

	Fútbol	Política	Economía
I1	1.00	0	0
I2	1.00	0	0
I3	1.00	0	0
I4	0	1.00	0
I5	0	1.00	0
I6	0	1.00	0
I7	0	0	1.00
I8	0	0	1.00
I9	0	0.21	0.79
I10	0.33	0.33	0.33
I11	0.40	0.60	0

Figura 10. Resultado de calcular los temas con PLSI.
Fuente:(Moya Garcia, 2015).

- **Latent dirichlet allocation (en adelante LDA)**

El padre del modelado de tópicos es David Blei, que en (Blei, 2012), presenta de forma detallada las aplicaciones del Topic Models, donde define que un tópico es el conjunto de términos que representa una temática alojada en un corpus de documentos, sin pérdida de información estadística. Este enfoque persigue fielmente identifica relaciones latentes entre documentos pertenecientes al corpus, teniendo como meta brindar una descripción concisa de este sin presentar perdida de información desde la perspectiva estadística(Alvarado, 2015).

En (Alvarado, 2015)El aprendizaje no supervisado es empleado para descubrir tópicos e interacciones que se producen. Este tipo de aprendizaje, las etiquetas de clase son desconocidas, para lo cual se busca agrupar el conjunto de datos en base a similitudes existentes. El aprendizaje no supervisado se divide en dos:

- Probabilístico como Bayes, LDA, entre otros.
- No probabilísticos que mide distancias, entropías o métricas.

La distribución de categorías posee una distribución a priori de Dirichlet. Siendo esta una característica que ha llevado a un amplio campo de investigación e implementación de esta técnica en diferentes campos como en la biomédica en (Ruiz-Correa, 2010), en la Bioinformática en (Bisgin, Liu, Fang, & Xu, 2011), medir similitud entre libros bíblicos en (Hu, 2012), en el deporte como en (Pérez, 2012), entre otras diferentes campos de investigación e implementaciones(Alvarado, 2015).

En (Seiter, Amft, Rossi, & Tröster, 2014), determinan que la principal limitación de modelos de uní grama (uni-gram) es suponer que la colección de documentos están formados por

términos homogéneos, haciendo alusión que el corpus de documentos presenta un único tópico. Demostrando que LDA presenta menor sensibilidad al ruido de los datos(Alvarado, 2015).

LDA está dotado de la capacidad de agrupar y clasificar colecciones de documentos en función de características similares, sin la necesidad de un conocimiento a priori, dotándole del potencial para diversas aplicaciones(Alvarado, 2015).

La técnica de LDA es un modelo probabilístico del Machine Learning, que es enmarcado dentro de los modelos generativos, al tratar de describir como se crea un documento. Al igual que PLSI, calcula dos matrices de probabilidad, en el caso de LDA son $\boldsymbol{P(w|z)}$ y $\boldsymbol{P(w|\theta)}$. La diferencia entre las técnicas de LDA y PLSI, es como se calcula las matrices de probabilidad, en ambas técnicas, cada ítem este representado por un vector que sigue una distribución para el caso de PLSI es Categórica, y para LDA es Dirichlet. (González, 2013; Hernando et al., 2013)

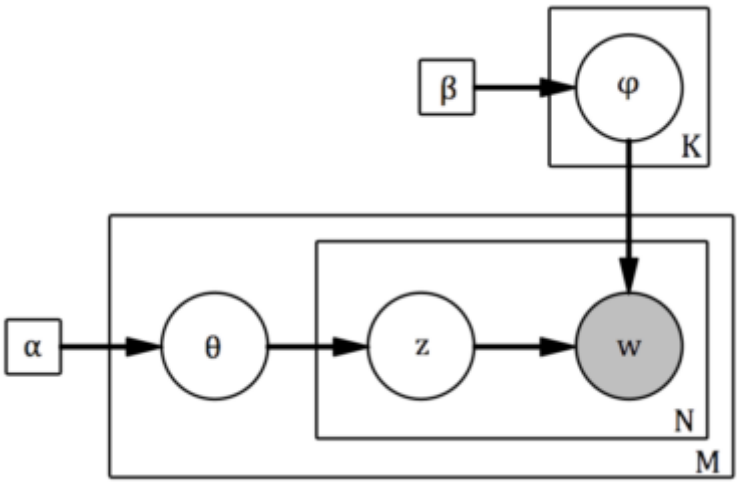


Figura 11. Modelo de LDA.
Fuente: (Moya Garcia, 2015).

(Tamaral, 2016)Los símbolos y letras que aparecen en la Figura 12, tienen el siguiente significado:

- M: número de documentos (ítems) con los que se está trabajando.
- K: número de tópicos o temas que conforman los documentos.
- N: número de palabras por cada documento presente.
- W: conjunto de todas las palabras de la colección de documentos. Éste es el único parámetro observable. Todos los demás son latentes, por lo que son inferidos a partir de las palabras. **"W_{ij}"**, representa a cada palabra i en el documento j.
- α: híper parámetro (debido a que es un parámetro de una distribución a priori) de Dirichlet de cada tópico en un documento.

- β: híper parámetro de Dirichlet de cada palabra en un tópico.
- θ_i: distribución de tópicos para cada documento i.
- φ_k: distribución de palabras para cada tópico k.
- Z_{ij}: indica el tópico de la palabra j en el documento i.

LDA es la versión Bayesiana de "Probabilistic Latent Semantic Analysis" (PLSA). Modelando las probabilidades priori/posterior con funciones Dirichlet de híper parámetro **"η"** y **"α"**. Para lo cual necesita conocer los documentos y el número de temas o tópicos (en adelante sea hace referencia a estos dos términos con el mismo significado) **"K"** a los que hacen referencia los textos(Contador Pachon, 2015).

Calculando la probabilidad de la colección de documentos dados los temas (priori este asigna los temas a los documentos de la colección). La probabilidad de los temas, empleando los documentos (posterior crea los temas de la colección)

$p(W_{1:D})$ Es la probabilidad marginal, requiere mucho tiempo de cómputo, lo cual no es factible. Para ello se han desarrollado varios métodos para mejorar el tiempo de computacional entre los cuales se mencionan en la Tabla 5.

Tabla 5. Métodos disponibles cálculo de híper parámetro con LDA.

Método	**Año**
Mean field variational methods	2001
Collapsed Gibbs sampling (CGS)	2002
Expectation propagation	2002
Collapse variational inference	2006
Online variational inference	2010

Fuente:(Contador Pachon, 2015).

Para la estimación de los parámetro de LDA, se puede elegir la inferencia variacional o el muestro de Gibbs. Las proporciones de temas aprendidos ***"θ",*** son elementos específicos, pero el conjunto de tópicos ***"φ",*** es compartido por todo los ítems.(Wu et al., 2016)

A continuación se muestra como es el proceso generativo de LDA, ilustrado en una serie de figuras:

La Figura 12 presenta como LDA, sigue una distribución Dirichlet para determinar los tópicos que conforman el corpus de documentos.

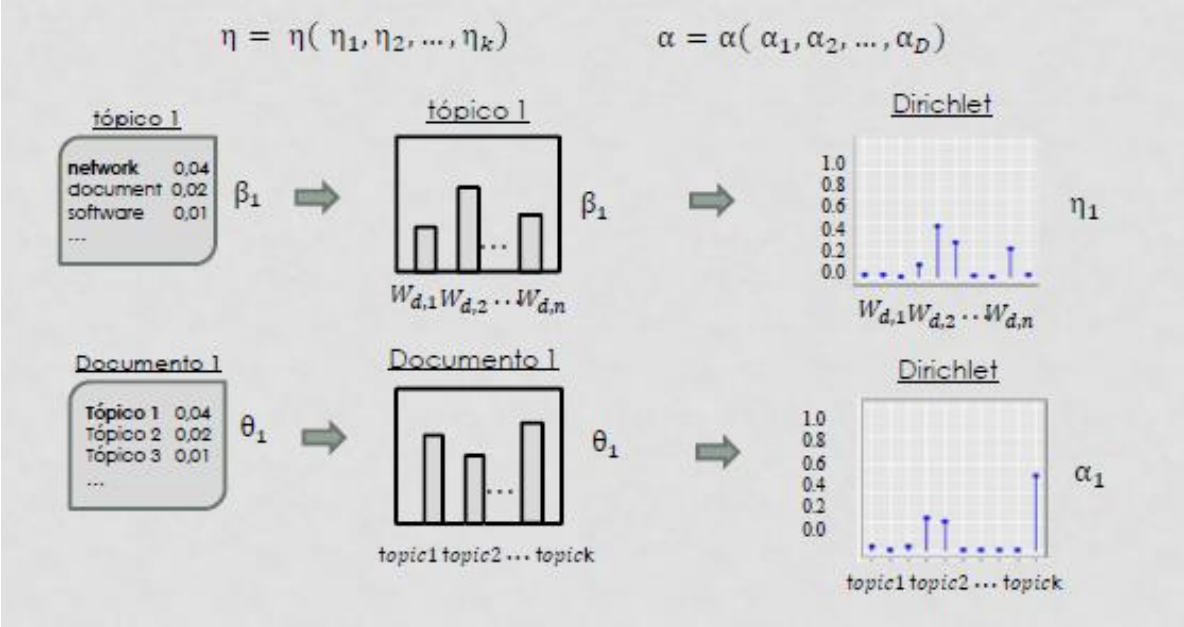


Figura 12. Distribución Dirichlet.
Fuente:(Contador Pachon, 2015).

LDA trabaja con colecciones de documentos o corpus, estos documentos son la entrada para seguir con el proceso generador; los documentos tienen, que seguir un pre-procesamiento dando granularidad a la estructura de los documentos para que el proceso sea eficaz. En la Figura 14 se aprecia cómo deben estar los documentos para ser trabajados con LDA. Dejando solo palabras que conforman cada uno de los documentos.

Figura 13. Granularidad en los documentos a procesar en técnica LDA.
Fuente:(Contador Pachon, 2015).

El proceso para crear tópicos o temas se realiza con ayuda de la selección de un sub-espacio con el muestre de Gibbs. La Figura 15, se ilustra cómo se limita el espacio donde se trabajó con los términos que describen los tópicos presentes en un documento.

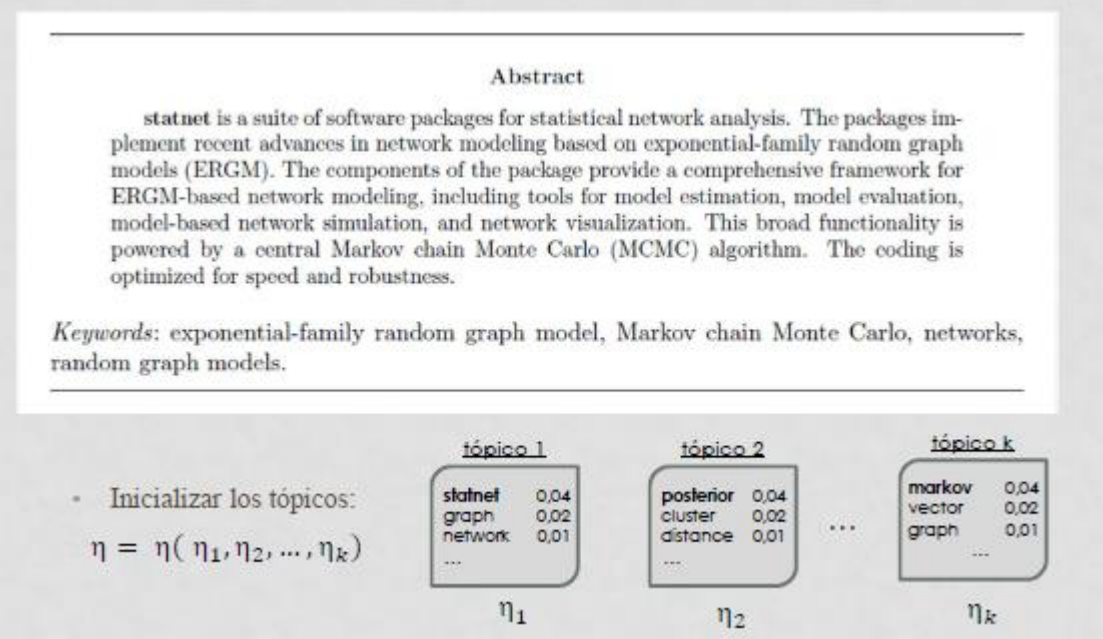
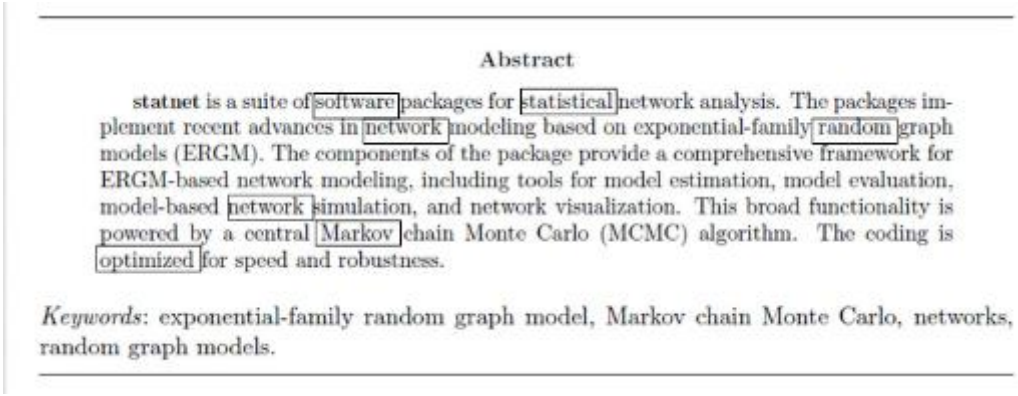


Figura 14. Crear topicos apartir de los terminos presentes en los documentos asignando estadistico.

Fuente: (Contador Pachon, 2015).

Selecciona un sub-espacio de búsqueda utilizando el método de MCMC (Markov Chain Monte Carlo). El muestreo de Gibbs, que es un algoritmo MCMC. Por tanto, genera cadenas de Markov de las muestras que selecciona, y de esta forma, se seleccionan las muestras más cercanas. De tal modo, se favorece que palabras próximas en el texto posean más probabilidad de pertenecer al mismo tópico. Este algoritmo es probabilista, por lo que hay que emplear una semilla en el programa, para que los resultados coincidan en diferentes ejecuciones con los mismos parámetros(Tamaral, 2016).



Abstract

statnet is a suite of software packages for statistical network analysis. The packages implement recent advances in network modeling based on exponential-family random graph models (ERGM). The components of the package provide a comprehensive framework for ERGM-based network modeling, including tools for model estimation, model evaluation, model-based network simulation, and network visualization. This broad functionality is powered by a central Markov chain Monte Carlo (MCMC) algorithm. The coding is optimized for speed and robustness.

Keywords: exponential-family random graph model, Markov chain Monte Carlo, networks, random graph models.

Figura 15. Selección de subespacio.
Fuente:(Contador Pachon, 2015).

Modificación de parámetro del híper parámetro η manteniéndolo normalizado.

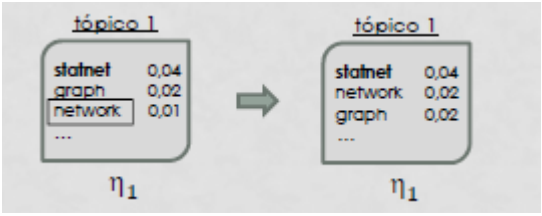


Figura 16. Modificación híper parámetro η.
Fuente: (Contador Pachon, 2015).

Calculando la probabilidad para que el modelo, explique los datos (posterior) Compara P (iteraciones = q) con la probabilidad de la iteración anterior, aplicando MLE. Repite el proceso hasta lograr el resultado óptimo.

Una vez creado los tópicos asigna los términos que describen el documento. En la Figura 17 se observa como son asignados los tópicos en un documento.

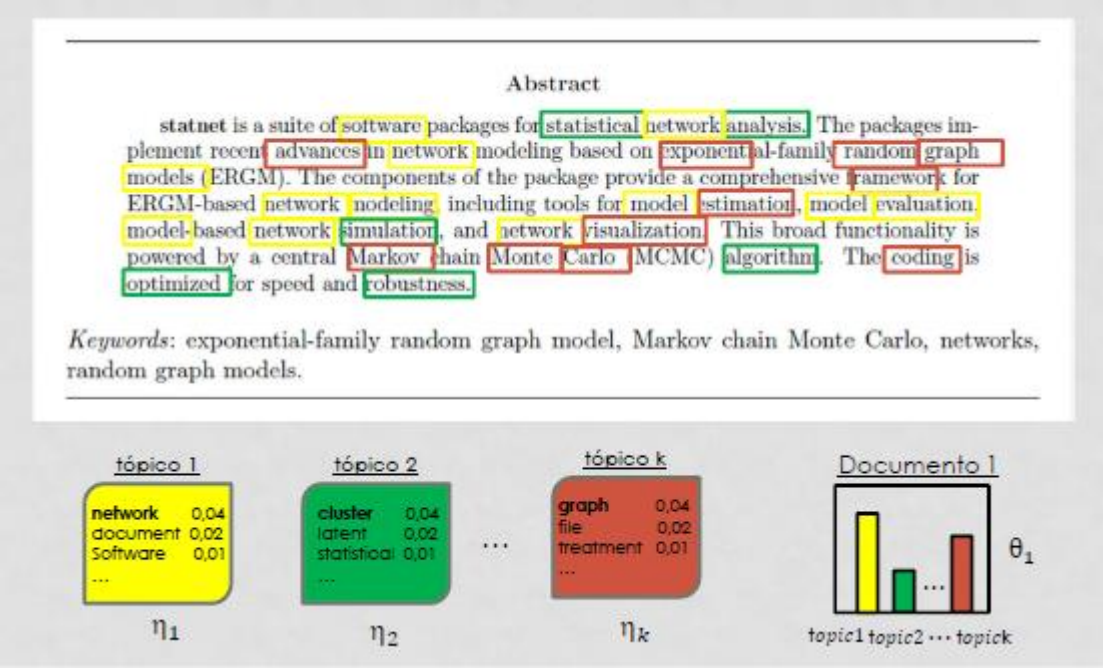


Figura 17. Asignacion de topicos a documentos.
Fuente:(Contador Pachon, 2015).

Cuando culmina de asignar los tópicos comenzara a clasificar los documentos.

El documento que se ha presentado a lo largo del proceso de ejecución del modelo como en el ejemplo de LDA llega a clasificar el documento con la probabilidad mayor que tiene el tópico NETWORK pero además de este también se encuentra conformado con una probabilidad menor por DOCUMENT y SOFTWARE.

Sin tomar la perspectiva estadística se puede decir que por cada uno de los documentos d se asocia aleatoriamente a cada término **"w"**, existente en el documento uno de los **"k"** tópicos. Con lo cual se tiene por cada documento d, por cada términos w en el documento d.

- Calcula la probabilidad de cada tema en cada documento con la $Probabilidad(tema\ t \mid documento\ d)$, encontraremos la proporción de palabras en **"d"**, que actualmente están asignadas a **"t"**.
- Calcula la probabilidad de cada palabra en cada tema $Probabilidad(palabra\ w \mid tema\ t)$, esta probabilidad representa la proporción que tiene asignada al tema **"t"**, en todos los documentos que proviene de las palabras **"w"**.

Se reasignara a la palabra w al nuevo tema t con probabilidad de cada tema en cada documento $P(tema\ t \mid documento\ d) * P(palabras\ w \mid tema\ t)$. Luego de varias interacciones determina un estado donde las asignaciones convergen en distribuciones estables.(Rus Marıa Mesas Javega, 2015)

Igual al PLSI, calcula dos matrices de probabilidad $P(w|z)$ y $P(z|\theta)$. La diferencia radica en la distribución Dirichlet que sigue LDA. Esta distribución es una generalización de la distribución Beta para el caso multinomial definida por:

- α, es el parámetro de la distribución de tópicos por documento (θ)
- β, es el parámetro de la distribución de palabras por tópicos (φ)

La estimación de los parámetros **"α"** y **"β"** es un problema irresoluble, razón por que se han desarrollado métodos aproximativos como los que se presenta en la Tabla 5.

2.3.3. Sistema de recomendación hibrido

Método de filtrado hibrido surge de combinar las fortalezas que presenta cada método de filtrado presentados en las secciones 2.3.1 y 2.3.2 respectivamente combinándolas en un solo método hibrido, reduciendo las debilidades que presenta cada método por separado(Parra, 2015d).

Filtrado colaborativo es el método más preciso entre los distintos métodos de filtrado, pero sufre de las siguientes debilidades:

- Dispersión (Sparsity).
- Arranque en frio (Cold start).
- Problema con un nuevo ítem (Problem new ítem).

Filtrado basado en contenido brinda la facilidad de trabajar con el contenido de los ítems permitiendo caracterizar su contenido, pero tiene las siguientes debilidades:

- Dispersión (Sparsity).
- Problema con un nuevo usuario (Problem new user).
- Sobre-especialización.
- Enfrentan problemas de semántica y sinonimia

Se puede decir de formar general que un sistema hibrido de recomendación teóricamente debe ser mejor, debido a que toman las ventajas tanto de FC y FBC, superando sus desventajas que presenta por separado(Peralta Costoya, 2013).

En (Peralta Costoya, 2013), denotan que un sistemas de recomendación hibrido usando Correlación de Pearson para FC y Support Vector Machines en FBC. Los resultados presentados denotan que la precisión del sistema es mejor que si emplea individualmente cada método de filtrado.

En el presente trabajo de titulación se investiga los distintos tipos de sistemas de recomendación presentados en secciones anteriores; como los algoritmos basados en

memoria y modelos que presenta cada tipo de filtrado, para encontrar el que mejor se adecue al objetivo del trabajo de titulación.

2.4. Evaluación de los SR

Un punto crítico y vital es la evaluación de los SR con métricas que permitan cuantificar la calidad de las recomendaciones. Es un tema ampliamente investigado y en constante evolución. La evaluación se puede clasificar en dos tipos: evaluación experimental de los resultados y evaluación centrada en el usuario. En este trabajo de titulación se revisan algunas medidas de evaluación experimental.

En otro trabajo de titulación se abarcan de manera más detalla este de tema de evaluación de las recomendaciones.

2.4.1. Evaluación experimental

En la evaluación experimental se utiliza métricas para examinar el rendimiento del sistema de recomendación comparando unos resultados experimentales con un conjunto de juicios de relevancia, sin interacción del usuario. Obteniendo una valoración cuantitativa de acuerdo a parámetros con alguna medida de evaluación. Podemos definir qué medida utilizar partiendo de la característica a medir dentro del sistema. Según la clase de parámetros empleados para medir el rendimiento, las medidas de evaluación de clasificaran en: Medidas de Exactitud, Precisión, Satisfacción, Diversidad y Novedad(González, 2013).

2.4.1.1. Medidas de exactitud

Un porcentaje grande de investigaciones se ha enfocado en el campo de evaluación experimental, se han centrado en la exactitud, comparando los rankings predichos por el sistema con los rankings reales de los usuarios hacia ítems del sistema. Estas medidas al tratar de reproducir la realidad lo más fielmente posible a través de recomendaciones acordes a la realidad de cada usuario. Los usuarios confiaran en el sistema. Entre esta clase de medidas de exactitud comúnmente empleadas tenemos: MAE y RMSE(González, 2013).

- **Mean absolute error (en adelante MAE por sus siglas en ingles)**

Esta medida de evaluación toma el error producido en las predicciones comparándolo con los resultados esperados para evaluar(González, 2013). Se resume en la Ecuación 5.

$$MAE = \frac{sumatoria\{|calificacion - prediccion|\}}{numero\ de\ casos}$$

Ecuación 5. Mean Absolute Error.
Fuente:(Fernando Ortega, 2013).

2.4.1.2. Medidas de precisión

Las medidas de precisión son concebidas para medir la capacidad del sistema al trabajar con un conjunto de ítems relevantes para un usuario y ordenarlos por importancia. Tenemos las siguientes medidas: Precisión y Recall, se debe tener presente que estas métricas técnicas no capturan si la usabilidad o calidad de las recomendaciones.

- **Precisión**

Mide el porcentaje de predicciones acertadas. La definición es la fracción de ítems relevantes del total del conjunto de ítems devueltos por el sistema. Lo cual se resume en la Ecuación 6.

$$Precisión = \frac{numero\ de\ recomendados\ y\ relevantes}{numero\ de\ recomendados}$$

Ecuación 6. Precisión.
Fuente: (Fernando Ortega, 2013).

- **Recall o cobertura**

Representa el total de predicciones correctas que pueden realizarse. Se define como la fracción de los ítems relevantes recomendados por el sistema sobre el total de ítems relevantes. La métrica de recall se puede resumir en la Ecuación 7.

$$Recall = \frac{numero\ de\ recomendados\ y\ relevantes}{numero\ de\ relevantes}$$

Ecuación 7. Recall.
Fuente:(Fernando Ortega, 2013).

2.4.1.3. Medidas de satisfacción

La satisfacción de los usuarios puede inclinarse a la usabilidad del sistema más que a su rendimiento. La usabilidad mide la capacidad de los SR para mantener un óptimo rendimiento con forme se añade datos; la finalidad es de explicar recomendaciones; o la cobertura, que es el porcentaje de contenidos recomendados.(González, 2013)

Esta medida capta criterios de satisfacción. Surgen del problema de utilizar únicamente medidas basadas en la precisión. Una alternativa son métricas que cuenten con cobertura. Atendiendo a las limitaciones de métricas de precisión atendiendo a usuarios con poca información(González, 2013).

- **Estabilidad**

Estabilidad refleja la consistencia de las recomendaciones ante la llegada de nuevas puntuaciones que concuerden con las estimaciones previas del sistema. Está estrechamente relacionada con la confianza y el grado de aceptación del sistema de parte de los usuarios. Pero no necesariamente tiene que estar relacionada la estabilidad con la exactitud pudiendo dar el caso de un sistema estable con un pobre rendimiento en términos de exactitud(González, 2013).

- **Robustez**

Una forma de mejor estabilidad en términos de exactitud seria la robustez o capacidad no tener influencia por valoraciones negativas que produzcan sesgos en los resultados del sistema(González, 2013).

2.4.1.4. Medidas de diversidad y novedad

La cobertura puede influir en la Diversidad de ítems recomendados que son recopilados y estos ítems son diferentes entre ellos. Han realizado estudios del impacto de la diversidad de ítems ofertados en SR de consumo comercial y también se ha tomado en cuenta la diversidad temporal, que mide la habilidad de un SR en evolucionar y adaptarse a los largo del tiempo. Como también evaluar la diversidad en listas de recomendaciones(González, 2013).

En (Peralta Costoya, 2013)Una característica deseada en los sistemas de recomendación, más difícil de evaluar, es la novedad de la recomendación, lo mismo con la métrica de serendipia, la cual es la capacidad del sistema de recomendar algo que el usuario no habría encontrado por su cuenta.

2.4.1.5. Medidas de similitud

La precisión de las recomendaciones depende de definir correctamente la similitud es el punto de partida y uno critico; persigue agrupar contenidos o usuarios de conjuntos de datos y sus representaciones. Para ello se debe definir una medida de similitud que establezca cuantitativamente lo cercanos o lejanos que están dos representaciones entre sí. Las medidas de similitud más utilizadas en SR son detalladas a continuación.(González, 2013; Hernando et al., 2013)

- **Función coseno**

Mide la similitud entre dos representaciones vectoriales; obteniendo como resultado el coseno del ángulo que forman las representaciones espaciales del par de vectores X e Y presentados en un espacio bidimensional formando un ángulo entre ellos, cuyo coseno representa la similitud entre ambos vectores; se puede generalizar esta función aplicando a los vectores n-dimensiones.

2.4.2. Explicación de las recomendaciones

Un punto a tomar en cuenta es justificar por qué se realiza la recomendación de un determinado ítem a un usuario u otro. Los usuarios tienden a desconfiar de los SR si estos fallan el usuario podría entenderlo. La finalidad es que el usuario de un voto de confía hacia el sistemas y las recomendaciones que percibe del mismo. Existen diversas formas de explicar una recomendación a los usuarios dentro de las medidas de evaluación de SR que cuantifican la calidad de las recomendaciones presentadas a los usuarios(González, 2013; Fernando Ortega, 2015).

Evaluar los sistemas de recomendación trae eficacia comercial, incrementando en click through, este es un indicador para medir la eficiencia de retorno de clientes, incremento de ventas lo cual aumenta los ingresos del negocio(Pablo Castells, Fernando Díez, 2011).

Evaluación de los sistemas de recomendación es un problema abierto, debido a que las métricas de error no necesariamente determinan la satisfacción del usuario(Pablo Castells, Fernando Díez, 2011).

- Los aciertos o errores en las valoraciones bajas son irrelevantes.
- Si los sistemas definen valoraciones sin predecir estas no se pueden evaluar.
- La efectividad en una recomendación se encuentra ligada a la valoración que percibe.

Las métricas empleadas para evaluar valoraciones no son fáciles de implementar. La suposición de la metodología de Cranfield, no llega a cumplir en la mayoría de experimentos de sistemas de recomendación. Existe una divergencia grande entre autores, lo cual produce que la comparación entre experimentos sea difícil(Pablo Castells, Fernando Díez, 2011).

En (Pablo Castells, Fernando Díez, 2011) se toma que el acierto no es el único factor de utilidad que se debe tomar en una recomendación y la efectividad del sistema. Para lo cual se debe analizar otras métricas como:

- Novedad
- Diversidad
- Cobertura
- Confianza

2.4.3. Perplexity

Esta métrica, se toma como valor en una distribución de probabilidades para predecir una muestra. Se usa comúnmente para comparar modelos probabilísticos. El valor de perplexity de una distribución de probabilidad discreta p viene definida por la Ecuación 9.

$$2^{H(p)} = 2 - \sum_{z} p(x) \log_2 p(x)$$

Ecuación 8. Perplexity.
Fuente: (Coronado Matutti, Cárdenas Acosta, Bello Medina, & Carrasco Rodríguez, 2015).

En (Coronado Matutti et al., 2015), donde H (p) es la entropía de la distribución y x son los eventos posibles. La perplexity de un modelo probabilístico desconocido p, puede ser propuesto basado en muestras de entrenamiento que fueron muestreados de p. Dado un modelo probabilístico q, uno podría evaluar q preguntándose qué tan bien el modelo predice muestras de test x_1, x_2,…, x_n muestreados de p. La perplexity del modelo q es definido en la ecuación 6.

$$\mathrm{b} - \frac{1}{N} \sum_{i=1}^{N} \log_b q(x_i)$$

Ecuación 9. Modelo perplexity.
Fuente: (Coronado Matutti et al., 2015).

Donde b es usualmente 2. Mejores modelos q de la distribución desconocida p, tenderán a asignar mayores probabilidades q(xi) a los eventos, por lo tanto si menor es el perplexity, mejor es el modelo(Coronado Matutti et al., 2015).

2.4.4. Distancia hellinger

Se usa para cuantificar la similitud entre dos distribuciones de probabilidad. En el presente trabajo, las distribuciones que comparamos son discretas, y la distancia Hellinger para las distribuciones $P = (p_1, \ldots, p_k)$y $Q = (q_1, \ldots, q_k)$, están definidas en la Ecuación 11.

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} \left(\sqrt{p_i} - \sqrt{q_i}\right)^2}$$

Ecuación 10. Distancia Hellinger.
Fuente: (Coronado Matutti et al., 2015).

El cual está relacionado directamente con la norma euclidiana de la diferencia de las raíces cuadradas de los vectores, la cual se presenta en la ecuación 8. (Coronado Matutti et al., 2015)

$$H(P,Q) = \frac{1}{\sqrt{2}} \left\| \sqrt{P} - \sqrt{Q} \right\|_2$$

Ecuación 11. Norma euclidiana.
Fuente: (Coronado Matutti et al., 2015).

En el presente trabajo de titulación se utiliza métricas técnicas como la Precisión y Recall para evaluar el sistema de recomendación a implementar.

Siendo deseable como trabajo futuro evaluar métricas más cualitativas para llegar a conclusiones sobre la calidad, usabilidad y novedad de las recomendaciones.

2.4.5. Validación cruzada (Cross Validation)

Existen diversas técnicas para validar métodos de regresión entre los que se encuentra la validación cruzada dentro de esta técnica se tiene métodos como hold-out y k-fold(Planells & Delegido, 2015).

2.4.5.1. Método hold-out

El hold-out separa el conjunto de datos en dos subconjuntos, el primero para entrenar el modelo y el segundo para pruebas de validación. De tal manera, se crea un modelo únicamente con un único conjunto de datos. Del modelo generado se emplea la salida que arroja el modelo comparándola con los datos reservados para la validación del modelo (estos no se han empleado para generar el modelo). Los resultados estadísticos obtenidos con los datos del subconjunto de validación son los que brindan validez del método empleado en términos de error(Planells & Delegido, 2015).

2.2.1.1. Método k-fold

El k-fold toma la base del método hold-out, dotándose de mayor utilidad al trabajar con conjunto de datos pequeños(Yang & Huang, 2014). Para trabajar con los datos se dividen en **"k"** subconjuntos, con la finalidad de emplear el método hold-out **"k"** veces, utilizando un subconjunto distinto para brindar validez al modelo entrenado con los distintos k-1 subconjuntos(Jung & Hu, 2015). El error medio obtenido a través de lo k análisis realizados brinda el error cometido por el método, permitiendo evaluar su validez(Planells & Delegido, 2015).

Comparando los dos métodos presentados, el método k-fold posee la ventaja que todo el conjunto de datos es empleado para entrenar y validar, brindando resultados más representativos a priori. El caso del método hold-out, implementa el proceso n veces de forma aleatoria, lo cual no garantiza que los datos tomados para entrenar y validar no se repitan durante el proceso aleatorio(Planells & Delegido, 2015).

	A	B	C	D	E
k-fold 1	PRUEBAS	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO
k-fold 2	ENTRENAMIENTO	PRUEBAS	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO
k-fold 3	ENTRENAMIENTO	ENTRENAMIENTO	PRUEBAS	ENTRENAMIENTO	ENTRENAMIENTO
k-fold 4	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO	PRUEBAS	ENTRENAMIENTO
k-fold 5	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO	ENTRENAMIENTO	PRUEBAS

Figura 18. Ejemplo de k-fold de validación cruzada con 5 particiones.
Fuente:(Hackeling, 2014).
Elaboración: el autor.

La Figura 18 presenta k-fold de validación cruzada con 5 subconjuntos de tamaños iguales de notando cada uno con etiquetas de la letra A hasta la E en sus columnas correspondientes. En cada uno de los subconjuntos la partición de pruebas varia mientras se entrena y prueba el modelo en todas las particiones.

2.5. Trabajos relacionados

2.5.1. Movie recommendation based on collaborative topic modeling

En este trabajo implementan Collaborative Topic Modeling (CTR) el cual es una aproximación hibrida entre el filtrado colaborativo y basa en contenido para recomendar películas tomando el contenido de los ítems como los rankings de la comunidad de usuarios. Los conjuntos de datos empleados son el de MovieLens de 10M. Y CMU Movie Summary Corpus el cual consta de 42,306 descripciones de películas. Ambos conjuntos de datos fueron separados en entrenamiento y pruebas respectivamente. Los resultados que obtienen con la implementación del modelo CTR mejoran los métodos de filtrado colaborativo tradicional con la adición de información de contenido con LDA (Bhowmick et al., 2014). En Collaborative topic modeling for recommending scientific articles presentan originalmente la combinación de las ventajas que presenta el filtrado colaborativo y Probabilistic Topic Model esta última es una técnica de aprendizaje automático que brinda una estructura latente interpretable para los usuarios sobre los ítems, para ofrecer recomendaciones de artículos científicos del sitio CiteUlike, se trabaja con el contenido de los artículos. Se presenta como el enfoque propuesto mejora las predicciones de ítems sin rankings por parte de los usuarios con respecto a los

métodos de Factorización de Matrices. Con la combinación del filtrado colaborativo con el método de factores LDA combatiendo el problema de nuevo usuario y ítem(Wang & Blei, 2011).

2.5.2. Incorporating group recommendations to recommender systems: alternatives and performance

En este trabajo abordan las recomendaciones a grupos de usuarios lo cual supone un nuevo reto en el área de los sistemas de recomendación por el alto nivel de complejidad que presenta. Se parte definiendo la similitud entre los grupos y el usuario para calcular las predicciones usando las votaciones de los K vecinos. La experimentación que se lleva contempla medidas de calidad clásicas tanto para las predicciones como de recomendaciones con una medida propuesta denominada UGSM. Los conjuntos de datos que se utilizan son MovieLens y Netflix. La hipótesis que plantea es demostrada, que la unificación temprana de los datos del grupo mejora el rendimiento y no afecta la calidad de las recomendaciones. Este trabajo deja varios retos por abordar como nuevas métricas de similitud orientada a la recomendación a grupos(F. Ortega, Bobadilla, Hernando, & Gutiérrez, 2013).

2.5.3. Hierarchical graph maps for visualization of collaborative recommender systems

En este trabajo se enfrenta el reto de recomendaciones a usuarios no registrados en el sistema debido a que el número de estos usuarios es mayor. Se busca ofrecer un modelo de inferencia sencillo de interpretar que permita a un usuario no registrado inferir sus recomendaciones. Utilizan árboles de recubrimiento mínimo (ARM) con el cual crean el modelo RS-IST (Recommender System Items Similarities Tree) Presentan un modelo de visualización para los ítems de un SR para estudiar la similitud. El modelo se prueba con el conjunto de datos de MovieLens 1 Millón. Evaluar la calidad de predicciones como de las recomendaciones es crítico y con modelo RS-IST es posible explicar las recomendaciones al estilo ítem. Los retos que se plantean necesitan de ardua investigación al existir escaso interés en los usuarios no registrados(Hernando et al., 2013).

2.5.4. Efficient features for movie recommendation system

En este trabajo presentan el desarrollo de un sistema de recomendación empleando la técnica LDA para recomendar películas basándose en la información contenida en las descripciones textuales, midiendo la similitud presentada por las distribuciones con Kullback-Leibler divergence . Evalúan la calidad de las recomendaciones con rankings que los usuarios emiten sobre las películas recomendadas por el sistema. Lo que se busca es explicar por qué se recomienda una película al usuario basando en las características que se obtiene con el modelo LDA.(Bhargav, 2014)

En el ANEXO 1 se encuentra una tabla con trabajos que describen técnicas de filtrado colaborativo como filtrado basado en contenido detallando las investigaciones de los últimos 5 años empleando técnicas como LDA y PMF en diferentes contextos.

2.6. Selección de técnicas

Las técnicas que utilizan el espacio latente para explicar los rankings de usuarios observados y palabras observadas. Mejoran la calidad de las recomendaciones sobre nuevos ítems en el sistema atacando el problema de arranque en frio.

Existen varios algoritmos basados en Filtrado Colaborativo que implementan modelos basados en Factorización matricial han demostrado lograr una precisión satisfactoria en la predicción de rankings a ítems sin calificación(Li, Li, Yao, Hwang, & Zhang, 2014).

En la aproximación del sistema de recomendación hibrido se busca tener recomendación novedosas, para lo cual no se puede concebir un sistemas de recomendación puramente basados en contenido, este presentan recomendación similares a ítems previamente consumidos por el usuario. Lo que se busca es entender porque se recomienda un ítems para esta tarea es necesario incluir el FBC pero incluyendo las ventajas del FC, para superar los problemas de sobre-especialización y el ingreso de nuevos usuarios e ítems al sistema.

Un enfoque que presenta ventaja sobre otros enfoques es Probabilistic Matrix Factorization. Este método escala linealmente con el número de observaciones y tiene un buen desempeño en conjunto de datos grandes, escasos y no balanceados; a este enfoque se le añade un tratamiento bayesiano denominando al nuevo enfoque como BPMF. Estos modelos pueden ser entrenados eficientemente usando métodos de Cadenas de Markov Monte Carlo, presentando mayor precisión en la predicción que PMF, el cual es entrenado con estimaciones MAP(Peralta Costoya, 2013).

Estos enfoque funcionan bien para realizar recomendación de ítems ya consumidos por los usuarios en el pasado; pero no se puede generalizar para ítems desconocidos. Para lo cual el enfoque de LDA, el cual es un modelo probabilístico generativo para conjunto de datos discretos como texto. Este es un modelo bayesiano jerárquico de tres niveles, dentro del cual cada ítem de la colección se modela como una mixtura finita sobre una colección de tópicos subyacentes. Cada tópico es modelado como una mixtura infinita sobre el conjunto de probabilidades de tópicos subyacentes. Las probabilidades de tópicos brindan una representación explicita de un documento que se le puede aplicar a la información textual de los ítems(Peralta Costoya, 2013).

Un modelo probabilístico como LDA es apropiado para lograr un aproximación hibrida de un sistemas de recomendación. En la literatura se encuentra un sistema hibrido que mezcla LDA

con técnicas de filtrado colaborativo, denominado CTR(Chong Wang, 2011). Este enfoque se implementa para recomendar artículos científicos, tomando ítems nuevos en el sistema, para recomendarlos a los usuarios. Ofrece representaciones interpretables de las recomendaciones presentadas tanto de los usuarios como de ítems. Esto se logra con la combinación de técnicas de FC basados en factores latentes y FBC basado en modelo probabilísticos de tópicos, donde las recomendaciones para un usuario en particular son parecidos a calcular una esperanza condicional de variables ocultas(Peralta Costoya, 2013).

El modelo CTR brinda mejores predicciones que los algoritmos tradicionales de Filtrado Colaborativo. Este modelo capta de manera más realista los gustos de los usuarios dando como resultado una variedad más amplia de recomendaciones(Bhowmick et al., 2014).

Hay diversas formas de clasificar representaciones textuales en el presente trabajo se presentan técnicas probabilísticas y no probabilísticas. En la Tabla 6 se detalla una comparación entre las diferentes técnicas presentadas en base a diferentes criterios. Se puede apreciar que las técnicas presentadas cumplen con no tener en cuenta el orden de las palabras. Se puede concluir que la técnica PLSI es sensible a la complejidad asociada al tamaño de colecciones de documentos. También tenemos que la técnica VSM es sensible al vocabulario, debido a que el resto de técnicas buscan tópicos latentes. LDA brinda un rendimiento superior en comparación de las demás técnicas presentadas. Asiendo de LDA un estándar en diversas áreas de investigación como recuperación de información, clasificación de textos, entre otros.

Tabla 6. Comparación de técnicas presentadas en base a criterios.

MODELO	**Sensibilidad al vocabulario**	**Tiene en cuenta orden de las palabras**	**Complejidad asociada al tamaño**
VSM	NO CUMPLE	CUMPLE	CUMPLE
LSI	CUMPLE	CUMPLE	CUMPLE
PLSI	CUMPLE	CUMPLE	NO CUMPLE
LDA	CUMPLE	CUMPLE	CUMPLE

Fuente: (González, 2013).
Elaboración: el autor.

2.6.1. Resultados esperados

Como resultado del presente capitulo se obtiene el estado del arte y marco teórico de técnicas de Machine Learning empleadas en el proceso de recomendación de los sistemas de recomendación, la clasificación de los distintos tipos de filtrado, disponibles para realizar

recomendaciones. Así como también métricas para evaluar la calidad de las recomendaciones.

Dentro del estado del arte se encuentra una sección de trabajos relacionados, que dan las pautas para la selección del modelo de filtrado basado en contenido.

El modelo seleccionado es LDA, este brinda la capacidad de caracterizar los ítems para emitir recomendaciones en base a la información de los ítems.

CAPITULO III: IMPLEMENTACIÓN

El presente capítulo presenta la implementación de un sistema de recomendación utilizando la técnica Topic Model con la variante LDA, utilizando el conjunto de datos de MovieLens dentro del lenguaje de programación R.

Este capítulo expone los principales pasos involucrados para la creación de recomendaciones en un sistema de recomendación basado en contenido.

Resumen del sistema:

- Crear un conjunto de datos (o dataset) seguido de una serie de pre-procesamientos sobre los datos mediante técnicas de procesamiento del lenguaje natural.
- El conjunto de datos contiene descripciones textuales de películas procedentes del sitio web de IMDB, contando con 10,329 películas.
- Generación de un modelo utilizando LDA el cual es entrenado con el conjunto de datos antes mencionado.
- Utilización de métricas de similitud, se compara los temas producidos por la técnica LDA tanto con el método Gibbs y VEM.

3.1. Pasos involucrados en la generación de un modelo con la técnica LDA

El primer paso a seguir para clasificar textos empleando técnicas de aprendizaje computacional o automático , es obtener atributos que caracterizan el texto a clasificar, así como aplicarle transformaciones para obtener una representación adecuada de los textos para su posterior procesamiento (Alvarado, 2015).

A continuación se presenta como se estructura los pasos involucrados en la creación de un modelo con la técnica LDA:

- Especificación del conjunto de datos.
- Herramientas utilizadas
- Carga de colección de documentos o corpus (conjunto de textos).
- Pre-procesamiento del corpus.
- Creación de Document Term Matriz (en adelante DTM).
- Determinar el número óptimo de tópicos que caractericen el conjunto de datos.
- Generación del modelo con técnica LDA
- Interpretación del modelo generado

3.1.1. Especificaciones del conjunto de datos

Siguiendo los objetivos propuesto en el presente trabajo de titulación la cual persigue la Implementación de un modelo de aprendizaje automático para la recomendación de ítems, se ha decidido trabajar con un conjunto de datos procedente de MovieLens debido a que es un servicio que capta un porcentaje considerable de usuarios en internet además que en el año 2007, Nexflix, propuso el reto de mejorar las recomendaciones que su sistema online de recomendaciones el mismo que ofrece a sus clientes recomendación de películas que los clientes pueden adquirir en el sitio esto con el fin de aumentar sus ventas.

El presente trabajo de titulación se trabaja el conjunto de datos MovieLens, el cual es una contribución del GroupLens este es un grupo de investigación del Departamento de Ciencias de la Computación e Ingeniería de la Universidad de Minnesota. Este grupo desde sus inicios ha manejado proyectos de investigación en variedad de campos de los cuales se menciona: sistemas de recomendación, comunidades en línea, bibliotecas digitales, entre otros.

Existen diferentes tipos de conjunto de datos disponibles por el GroupLens entre los cuales se trabaja con MovieLens Latest Datasets. Dentro de esta selección de datos dispone 2 versiones una **SMALL** y **FULL**, cada una de estas versiones tiene diferentes características como se detalla en la Tabla 7.

Tabla 7. Conjunto de datos MovieLens Latest Datasets.

Nombre Dataset	**rankings**	**Tag**	**Movies**	**Users**	**Last update**
Small	100,000	6,100	10,000	700	01/01/2016
Full	22,000,000	580,000	33,000	240,000	01/01/2016

Fuente: el autor.

El conjunto de datos con el que lleva el trabajo es la denominación **Small**, dentro de este se describen rankings con valor entre 1 al 5 con intervalo de 0,5. Conteniendo 105,339 rankings y 6,138 tag de 10,329 películas. Este conjunto de datos se ha creado con información proveniente de 668 usuarios de entre 3 de abril de 1996 y 9 de enero de 2016. El conjunto de datos fue generado el 11 de enero de 2016. Se trabaja con este conjunto de datos es por su número reducido de películas de las cuales está conformado al ser un trabajo investigativo la finalidad es meramente académica, además que el tiempo de procesamiento con un volumen mayor de datos es costoso computacionalmente. La Tabla 8 brinda especificación detallada del conjunto de datos MovieLens Latest Datasets Small.

Tabla 8. Conjunto de datos MovieLens Latest Datasets Small.

MovieLens Latest Datasets Small	
Característica	**Valor Numérico**

Usuarios	668
Películas	10329
rankings	105339
Tags	6138

Fuente: el autor.

El conjunto de datos detallado en la Tabla 8 contiene información de los usuarios como rankings hacia películas existentes en MovieLens, estos usuarios se han seleccionado al azar para la inclusión dentro del conjunto de datos. Todos los usuarios que forman parte del conjunto de datos habían emitido rankings o clasificado al menos 20 películas. No se incluye ninguna información demográfica de los usuarios como podría ser género, edad, país, entre otra. Cada usuario está identificado por un ID, y no se proporciona ningún otro tipo de información sobre el mismo.

Para la finalidad del presente trabajo de titulación, se desarrolla una herramienta capaz de obtener las descripciones de las películas, para el efecto de trabajar con las descripciones que acompañan el título de la película buscando fielmente seguir los objetivos del presente trabajo de titulación que es la implementación de un modelo de aprendizaje automático para la recomendación de ítems combinando técnicas de filtrado colaborativo y filtrado basado en contenido. En el filtrado colaborativo se utiliza los rankings de los usuarios hacia las películas, para el filtrado basado en contenido se necesitara las descripciones de las películas. Dentro del conjunto de datos se especifica que el archivo links.csv contiene identificadores de las películas en distintas fuentes como ***MovieLens, IMDB*** y ***TMDB***. El encabezado del archivo links.csv presenta información correspondiente a cada fuente donde se puede encontrar mayor detalle de cada película, en cada fila contiene los códigos que pueden utilizarse para obtener las descripciones de las películas siguiendo el patrón presentado en la Tabla 9.

Tabla 9. Estructura de links de películas de MovieLens.

MovieLens	**IMDB**	**TMDB**	**Título de Película**
https://movielens.org/movies/1	http://www.imdb.com/title/tt0114709/	https://www.themoviedb.org/movie/862	Toy Story (1995)
https://movielens.org/movies/2	http://www.imdb.com/title/tt0113497/	https://www.themoviedb.org/movie/8844	Jumanji (1995)
https://movielens.org/movies/3	http://www.imdb.com/title/tt0113228/	https://www.themoviedb.org/movie/15602	Grumpier Old Men (1995)

Fuente: el autor.

En la Tabla 9, se presenta una muestra de los enlaces con los códigos correspondientes tanto a ***MovieLens, IMDB y TMDB***, en el orden a como se encuentran en el archivo links.csv añadiendo el enlace correspondiente a cada sitio web para poder intuir como se crean los links para obtener las descripciones de las películas en la Figura 19 brinda una idea.

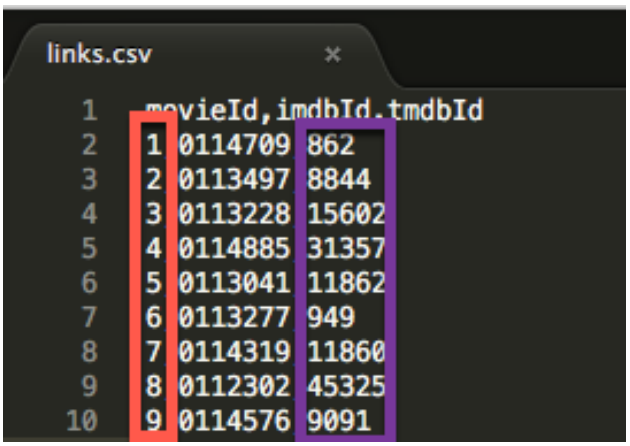


Figura 19. Estructura archivo links.csv.
Fuente: el autor.

La Figura 19 presenta una porción de cómo está estructurado el archivo links.csv en la primera columna denotada con un recuadro de color rojo se encuentran los códigos pertenecientes a MovieLens, en la segunda columna se encuentran los códigos pertenecientes a IMDB y por último en la tercera columna denotada por el recuadro de color morado se encuentran los códigos de TMDB. Cada una de las filas representa una película y su código disponible en tres diferentes fuentes donde recuperar mayor información de cada película.

Siguiendo el formato establecido en el archivo links.csv, después de la fila de encabezado, encontramos que la primera columna que contiene el identificador utilizado MovieLens con **movieId**, la siguiente columna el identificador utilizado por IMDB como **imdbId**, con el cual se puede acceder a la información de la película según este identificar con la concatenación del enlace de la página de **IMDB** con el identificador, en la última columna se encuentra el identificador de **tmdbId**. Cabe recalcar que las fuentes de donde se recupera la información son páginas de servicio de alquiler de películas, como también base de datos con información de películas.

Para obtener las descripciones de las películas se emplean librerías del lenguaje de programación Python como ***bs4*** y ***urllib2***, con las cual se procede a explorar la estructura del

sitio web de Internet Movie DataBase (por sus siglas en ingles IMDB) y localizar la descripción de cada una de las películas, el código empleado para realizar la obtención de las descripciones de cada una de las 10,329 películas del sitio ***IMDB*** se encuentra en el ANEXO 2.

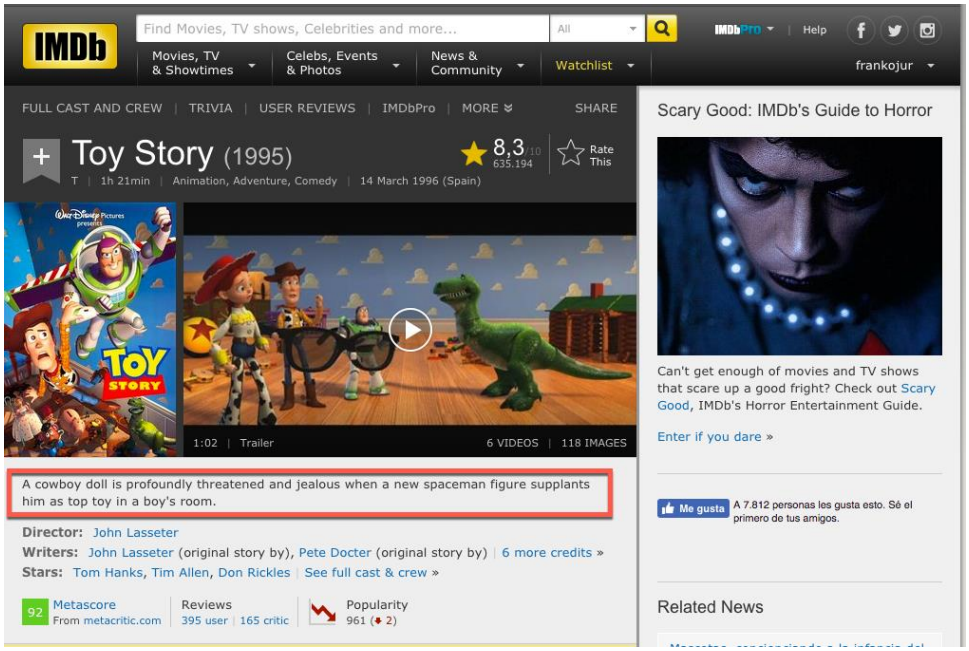


Figura 20. Estructura de la página web de Internet Movie DataBase (IMDB).
Fuente: el autor.

La Figura 20 presenta un enlace de apertura dentro de un navegador de la película “Toy Story” la misma que posee el código 1 en MovieLens; pero el código 0114709 en IMDB del cual se extrae la descripción disponible se denota mediante un recuadro de color rojo en la Figura 20 la descripción en formato textual la cual es extraída dentro de un archivo plano con ayuda de librerías del lenguaje de programación Python.

Una vez terminado el proceso de extracción de las descripciones de cada una de las películas del sitio IMDB, nos encuéntranos con el problema que algunas películas no disponen de una descripción en el sitio web de IMDB, el listado de estas películas se detalla a continuación en la Tabla 10.

Tabla 10. Películas sin descripción disponible en el sitio de web IMDB.

movieId	**imdbId**	**tmdbId**	**Título de película**
33090	0277909	30863	Mutant Aliens (2001)
96842	1510907	91911	Behind the Burly Q: The Story of

			Burlesque in America (2010)
95177	0354364	NA	Alcina (2000)
90090	1719498	113258	"Game of Death, The (Le Jeu de la Mort) (2010)"
77783	0064777	125271	Tora-san Our Lovable Tramp (Otoko wa tsurai yo) (1969)
68099	0180443	128857	Apollo 13: To the Edge and Back (1994)
56479	0141716	34424	Party Monster (1998)

Fuente: el autor.

Al no disponer de una descripción de estas películas en el sitio web de IMDB se procede a buscar en distintos sitios web como Amazon, TMDB entre otros. Donde se extrae una descripción de la película faltante, en algunos casos donde no existe una descripción de la película en un sitio que brinde el servicio de venta o alquiler de películas o sea una base de datos de películas se toma la revisión (review) de un usuario dentro del sitio IMDB el listado de estas películas se detalla a continuación en la Tabla 11.

Tabla 11. Observaciones de películas sin descripciones disponibles en el sitio web de IMDB.

Código IMDB mas enlace	**Sitio web donde se obtiene la descripción de la película**	**Observaciones acerca de las descripciones disponibles**
http://www.imdb.com/title/tt0277909	https://www.themoviedb.org/movie/30863-mutant-aliens	esta descripción fue traducida al encontrarse disponible en otro sitio web de películas
http://www.imdb.com/title/tt1510907/	https://www.rottentomatoes.com/m/behind_the_burly_q/	esta descripción se toma de otro sitio web de películas
http://www.imdb.com/title/tt0354364/	https://www.amazon.es/H%C3%A4ndel-Alcina-Staatsoper-Stuttgart-Alemania/dp/B00AA9QK4G	esta descripción se toma de otro sitio web de películas

http://www.imdb.com/title/tt1719498/	https://www.themoviedb.org/movie/113258-the-game-of-death	esta descripción fue traducida al encontrarse disponible en otro sitio web de películas
http://www.imdb.com/title/tt0064777/	http://www.imdb.com/title/tt0064777/	Se coloca la review de un usuario disponible en el mismo link al no encontrar disponibilidad en otro sitio web de películas
http://www.imdb.com/title/tt0180443/	http://www.imdb.com/title/tt0180443/	Se coloca la review de un usuario disponible en el mismo link al no encontrar disponibilidad en otro sitio web de películas
http://www.imdb.com/title/tt0141716/	http://www.imdb.com/title/tt0141716/	Se coloca la review de un usuario disponible en el mismo link al no encontrar disponibilidad en otro sitio web de películas

Fuente: el autor.

Se realiza una validación de todas las descripciones de películas extraídas en formato de texto plano para verificar que posean descripciones y que el conjunto de datos coincida con el total de películas contenidas en el conjunto de datos MovieLens todo este proceso se lo realiza de forma manual por parte del autor del presente trabajado de titulación.

Una vez finalizado el proceso de verificación de las 10,329 películas con sus respectivas descripciones que contienen información sobre las películas el conjunto de datos resultante coincide con los títulos disponibles en MovieLens.

3.1.2. Herramienta utilizada para desarrollo de técnica LDA

Para implementar la técnica LDA, se trabaja con el lenguaje de programación R por estar dotado de un entorno de Machine Learning que proporciona una gran variedad de herramientas y bibliotecas disponibles de forma libre.

R es un lenguaje de programación que es desarrollado por estadísticos para estadístico, aunque tiene un gran potencial y su aplicación se da en diversas áreas, entre las que se encuentra el Machine Learning que está constituido dentro del campo estadístico.

Para trabajar de manera eficiente con el lenguaje de programación R se emplea el entorno de desarrollo integrado (IDE) RStudio, el mismo que provee una interfaz amigable para el usuario dentro del lenguaje de programación R facilitando el trabajo con las bibliotecas y paquetes. Este IDE se encuentra disponible en código abierto, dispone de varios entornos para trabajar tanto web o localmente en el equipo(Racine, 2012).

A continuación se expone algunas características del IDE RStudio como:

- Resaltado de sintaxis, completado de código, identificación inteligente.
- Ejecución de código R de forma directa sobre la consola.
- Saltos rápidos a definiciones de funciones.
- Integración de ayuda y documentación.
- Administración fácil de varios directorios de trabajo.
- Navegador de espacios de trabajo y visualización de datos.

Existen dos paquetes disponibles para trabajar con Topic Model en el lenguaje de programación R como lo son:

- Paquete “lda”
- Paquete “topicmodels”

En los siguientes apartados se detalla el pre-procesamiento que se sufre cada una de los documentos que contiene la información de las películas del conjunto de datos de MovieLens.

3.1.3. Cambio de nombre a archivos planos

Cuando se generó los archivos planos de texto de las películas se guardó con el identificador correspondiente a MovieLens seguido del nombre con el cual se encontraba en el archivo movies.csv el mismo contiene el título de las películas con la cual se les proporciona dentro del sitio de MovieLens.

Se procede a cambiar el nombre de los 10,329 archivos planos que contienen las descripciones de las películas para poder trabajar de manera eficiente; debido a que algunos nombres de los archivos de texto plano son demasiado largos y poseen caracteres especiales, esto con la finalidad de evitar posibles inconvenientes en el tratamiento de los archivos, la Tabla 13 presenta en la columna Nombre Original se aprecia nombres de una muestra de algunas películas con nombres demasiado largos de los archivos de texto plano; además de estar acompañados de caracteres especiales. Los nombres de los archivos de texto plano tienen un patrón el cual es el siguiente primero el código del identificador de MovieLens

seguido de un guion medio con el título de la película. Con el nuevo nombre presenta el código MovieLens seguido de un guion medio con la palabras OK, la Tabla 13 presenta una pequeña muestra de cómo lucen los nombres de los archivos luego de modificarles el nombre original. Dejando solo el con el código del identificador MovieLens para hacer referencia a que descripción de película se realiza el análisis.

Tabla 12. Cambio de nombre a archivos.

Nombre Original	**Nombre Nuevo**
6722-Once Upon a Time in China II (Wong Fei-hung Ji Yi/ Naam yi dong ji keung) (1992).txt	6722-OK.txt
8456-Investigation of a Citizen Above Suspicion (Indagine su un cittadino al di sopra di ogni sospetto) (1970).txt	8456-OK.txt
26012-Samurai III/ Duel on Ganryu Island (a.k.a. Bushido) (Miyamoto Musashi kanketsuhen/ kettô Ganryûjima) (1956).txt	26012-OK.txt

Fuente: el autor.

3.1.4. Trabajo con el conjunto de datos dentro del IDE RStudio

Existen varias formas de cargar datos dentro del IDE de RStudio, como puede ser trabajando localmente o desde una base de datos, para el caso de trabajar desde un directorio local donde se encuentren alojados los datos a emplear. El conjunto de datos se trabaja localmente. Definiendo una ruta local en un directorio los archivos son necesarios para la creación de un Corpus el cual es una colección de los archivos que contienen las descripciones de las películas. Para conformar el corpus se necesita trabajar con la librería Text Mining (en adelante se emplea la siguiente nomenclatura “tm”) esta librería nos brinda funciones para realizar algunos tipos de pre-procesamiento de texto de forma que se pueda sacar el mayor provecho del mismo, este es un punto crucial para poder continuar con el desarrollo de la implementación de la técnica LDA, puesto que es el punto de partida para obtener el modelo que describa las recomendaciones de las películas.

Se puede obtener mayor información sobre la librería “tm” en el siguiente documento (Feinerer, Hornik, & Feinerer, 2015)

3.1.5. Preprocesamiento conjunto de datos

Dentro del pre-procesamiento se tiene técnicas de análisis de datos que permiten mejorar la calidad de un conjunto de datos, para que las técnicas de extracción de conocimiento obtengan mayor y mejor información(Alvarado, 2015)

Para procesar las descripciones de las películas del conjunto de datos MovieLens se emplea técnicas de procesamiento de lenguaje natural, para lograr sacar el mejor provecho del modelo a generar.

Dentro del procesamiento de lenguaje natural, un corpus es una colección de archivos textuales planos, utilizados para verificar hipótesis sobre el lenguaje, como extraer características de texto o encontrar patrones de uso de palabras. En el caso de las descripciones de las películas, se busca obtener características y patrones dentro del texto por lo cual se aplica pre-procesamiento(Bhargav, 2014).

.Después de importar el conjunto de archivos planos y convertir todos los documentos en un corpus, se procede a procesar el corpus con la librería “tm”, con una serie de transformaciones que se detallan a continuación:

- Normalización del texto en la colección de documentos dentro del corpus.
- Eliminación de palabras vacías.
- Remoción de números.
- Remoción de signos de puntuación.
- Normalización de términos
- Remoción de espacios vacíos.
- Remoción de signos especiales.

3.1.1.1 Normalización del texto en la colección de documentos.

Se lleva a cabo la normalización del texto contenido en los archivos planos de texto correspondientes al conjunto de datos MovieLens, esta normalización permite que se traten términos que son sintácticamente iguales como distintos. En el presente trabajo de titulación la normalización que se aplica es la transformación de todo el corpus a minúscula manteniéndolo normalizado (Alvarado, 2015).

3.1.1.2 Eliminación de palabras vacías.

La eliminación de Stopwords o palabras vacías es una lista de palabras cerradas, es decir términos gramáticas de alta frecuencia que se ignoran, al no proporcionar información útil, las cuales corresponden a pronombres, preposiciones, conjunciones, artículos, entre otros. Éstas aparecen con alta frecuencia en los documentos generando ruido y aumentando el tamaño de la representación. Las Stopwords disponibles en el paquete “tm” están en Inglés, para el efecto se las utiliza como parte del procesamiento del corpus como también un listado "SMART" que está disponible por la misma librería (Alvarado, 2015; Bhargav, 2014; Hofmann & Chisholm, 2016).

La eliminación de estas palabras permite trabajar con el conjunto de datos de forma eficiente y obtener el mayor provecho del modelo a generar con la técnica LDA.

3.1.1.3 Segmentación.

Dentro de la segmentación se abarca los siguientes puntos del pre-procesamiento como son:

- Remoción de números.
- Remoción de signos de puntuación.
- Remoción de espacios vacíos.
- Remoción de signos especiales.

Las segmentación consiste en separar el texto en unidades léxicas o también conocidos como token los cuales son un elemento básico de la minería de texto, que permite analizar y procesar texto a nivel de palabra(Bhargav, 2014). Una buena tokenización del texto tiene que pasar por la eliminación de signos de puntuación, números, espacios vacíos, signos especiales, paréntesis, guiones entre otros más. Se debe tener presente excepciones que pudiesen existir; puesto que provocarían un impacto negativo en la recuperación de información. Al corpus se lo segmenta quitando todos los signos de puntuación que posee la librería “tm” de R (Alvarado, 2015).

3.1.1.4 Lematización.

Este proceso de lematizacion hace alusión a remover los sufijos, reduciendo una palabra a su lema o raíz. Con frecuencia un término no aparece exactamente en un documento como se lo puede encontrar en distintos documentos, pero se lo encuentra con alguna de sus variantes gramaticales como plurales, gerundios, sufijos de tiempo verbal, entre otros. Este problema

se debe afrontar con la lematización de estas palabras a su raíz. En R se utiliza la librería “tm” para volver a su lema todas las palabras dentro de un corpus (Alvarado, 2015).

La Tabla 14 presenta algunos ejemplos con términos que son vueltos a su raíz con la función stemDocument de la librería ***“tm”***. Se puede apreciar como el término Killed es vuelto a raíz y se convierte a Kill, este proceso es realizado a todo el corpus que es creado con las descripciones de las películas de MovieLens.

Tabla 13. Proceso de lematizacion en los términos presentes en el corpus de películas de MovieLens.

	Sin procesar	**Procesada**
Termino	Killed	Kill
Termino	working	work
Termino	people	peopl

Fuente: el autor.

Para tener una idea de cómo se efectúa el pre-procesamiento dentro del corpus, se ha seguido el caso del documento 1003, correspondiente al identificador MovieLens de la película “Extreme Measures (1996)” luego de haber sufrido todos los procesamientos mencionados en los apartados 3.1.5.1 al 3.1.5.5. Se presenta en el ANEXO 3, una tabla donde se puede contemplar los cambios que ha sufrido los archivos durante todo el pre-procesamiento del corpus de archivos planos correspondiente a descripciones de películas de MovieLens.

El pre-procesamiento que se le aplica al corpus, cada documento de la clase seleccionada se convertirá a una representación compactada adecuada para el procesamiento con la técnica del algoritmo no supervisado LDA(Alvarado, 2015).

3.1.6. Matriz de documentos y términos (en adelante DTM)

La DTM es fundamental para poder generar un modelo con la técnica LDA, para poder sacar el máximo provecho de la técnica LDA se necesita que la DTM cuente con algunos controles con la finalidad de sacar el máximo provecho de los términos presentes en los documentos.

Con el corpus correspondiente a las descripciones de información de películas del conjunto de datos MovieLens se crea una DTM, esta es una matriz que contiene en sus filas los documentos en el caso del presente trabajo de titulación son las descripciones de películas, en las columnas los términos que caracterizan a las descripciones de las películas y en sus intersecciones se calculan la frecuencia de los términos en los documentos, con la función de ponderación del paquete ***“tm”***.

Se realiza un análisis de los términos con una alta frecuencia de aparición dentro de la DTM con la finalidad de identificar términos demasiado ocurrentes que puedan influir en la calidad de modelo a generar con la técnica LDA.

La Figura 22 presenta una gráfica de las frecuencias de términos con alta presencia la cual es superior a 200, donde se puede apreciar términos como family, father, find, finds, life, love, man, story, war, wife, woman, world y young, siendo estos términos los que aparecen con una frecuencia superior a 750 como se lo puede apreciar de mejor forma en el ANEXO 4, todos estos términos se encuentran relacionadas dentro del contexto de guiones de películas por tal razón no se procede a retirarlas de la DTM.

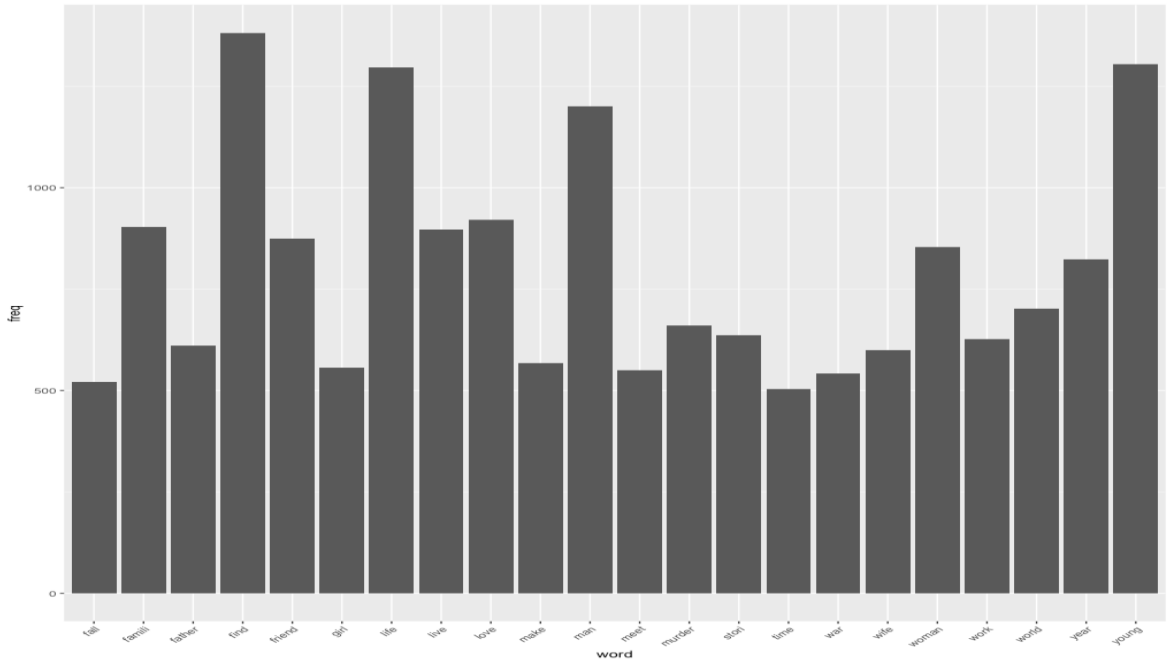


Figura 21. Términos con una frecuencia superior a 200 dentro de la DTM MovieLens.
Fuente: el autor.

Al trabajar con un volumen alto de información estos contendrán términos conformados por demasiadas letras producto del procesamiento que sufren o simplemente por redacción, como también se puede encontrar términos formados por un reducido número de letras los cuales no aportan información útil, para lo cual la Figura 23 presenta una gráfica de barras con el promedio de letras en los términos presentes en la DTM de películas del conjunto de datos MovieLens la cantidad de letras va desde 3 hasta 20, este rango de valores se encuentra ubicado en el eje X de la Figura 23, mientras que en el eje Y presenta la cantidad de términos. En la Figura 23 se puede observar que la media de letras se encuentra denotada con una línea de color verde ubicada en 7 pero se denota una concentración mayor de términos entre

el rango de 3 a 15 letras donde la cantidad de términos se agrupan con mayor frecuencia; razón por la cual se trabaja con este rango de letras para los términos presentes en la DTM

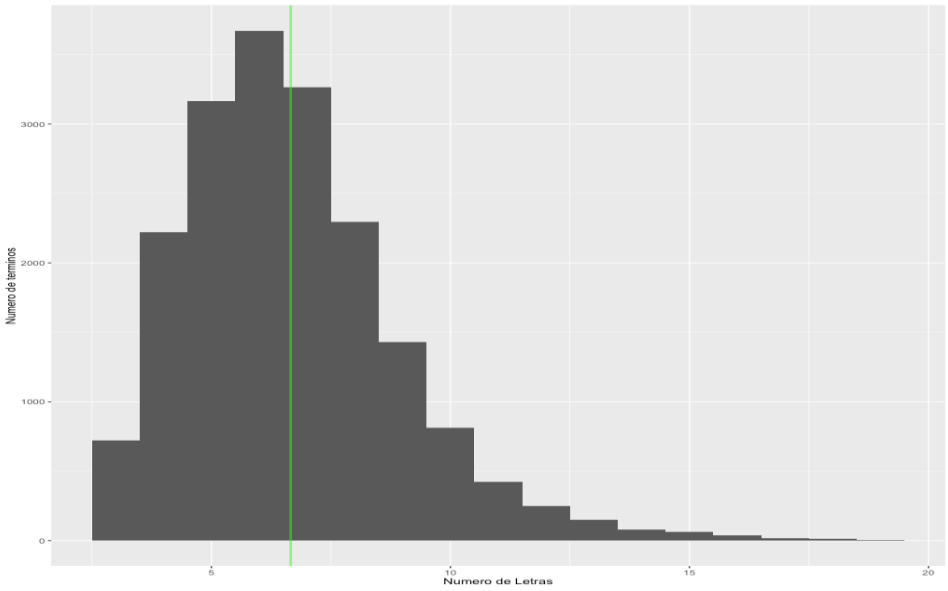


Figura 22. Promedio de letras en términos de la DTM de películas.
Fuente: el autor.

Determinando el control sobre la DTM que solo tome en consideración aquellos términos conformados por un rango de letras entre 3 y 15, tomando la gráfica de frecuencias que se presenta en la Figura 23. Notaremos que se reduce el número de términos presentes en la DTM este se lo puede constatar en la Figura 24 que presenta una salida de consola dentro del IDE RStudio, en la Figura 24 lado a presenta la DTM sin el control de rango de letras se puede apreciar que está conformada por 10329 documentos y 18636 términos; mientras que en la Figura 24 lado b está conformada por 10329 documentos y 18544 términos. La reducción de la DTM es significativo indicando que existían varios términos conformados por más de 15 letras. Además de esta forma no se encontraran términos que no aporten información útil y a su efecto llenen de ruido a la matriz DTM.

Lado a)

```
<<DocumentTermMatrix (documents: 10329, terms: 18636)>>
Non-/sparse entries: 174596/192316648
Sparsity           : 100%
Maximal term length: 36
Weighting          : term frequency (tf)
```

Lado b)

```
<<DocumentTermMatrix (documents: 10329, terms: 18544)>>
Non-/sparse entries: 174492/191366484
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
```

Figura 23. DTM de las descripciones del conjunto de datos MovieLens.
Fuente: el autor.

Sin embargo la cantidad de términos es relativamente grande en la DTM resultante del control de rango de letras y el nivel de escasez sigue siendo del 100%, lo cual es comprensible debido a que la DTM está compuesta principalmente por ceros en sus intersecciones. La Figura 25 presenta una muestra de la DTM, en la cual se aprecia que está siendo conformada por ceros dentro de las intersecciones que son el producto de calcular la ponderación de la ocurrencia de un término presente en las columnas y un documento presente en las filas de la DTM.

	Ángelä	Édith	Émilien	émigré	aaä	aaaah	aag	aang	aarń	aaron	aback	abagnal	abandon	abbé	abbä	abbandonä	abbey	abbí	abbot	abbott	abc	abctv	abdic	abduct	abductor
1-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100036-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100083-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100091-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100106-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100108-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100138-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100159-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100163-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100165-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100169-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100238-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100240-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100244-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1003-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100304-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100308-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100322-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100326-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100344-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100383-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 24. DTM del corpus de películas de MovieLens.
Fuente: el autor.

En (Crc et al., 2016) utilizan una forma para reducir la escasez. Partiendo de la idea donde en LDA las palabras que ocurren frecuentemente en unos documentos y en otros no, generan conflicto al momento de generar un modelo. Para las palabras que ocurren en muchos documentos y muchas veces en un solo documento reciben una menor ponderación usando el método term frequency inverse document frequency (TF-IDF) esta es una medida estadística de cuán importante es una palabra dentro de una colección de documentos para mayor detalle revisar (Hofmann & Chisholm, 2016). Permitiendo que aquellos términos con frecuencia baja se omitan, debido a que no aporta al modelo como también para aquellos términos que ocurren en varios documentos.

El cálculo de TF-IDF requiere de la librería Sparse Lightweight Arrays and Matrices ("slam"), esta proporciona estructuras de datos y algoritmos trabajar con matrices escasas. Para mayor detalle de la librería "slam" revisar (Kurt Hornik, David Meyer, & Christian Buchta, 2016)

Se realiza varias experimentos siguiendo el método TF-IDF, examinando los resultados, se nota una marcada reducción de términos en la DTM, pero también se elimina documentos por tal razón no se emplea este enfoque. Debido a que todas las películas tienen que estar presentes en la creación del modelo con la técnica LDA.

Se puede atribuir la pérdida de documentos a la reducida información que contienen algunos de estos documentos que conforman la DTM debido a que la descripción disponible es muy escasa de información.

La Figura 26 muestra información referente a la DTM aplicando el enfoque TF-IDF, se puede confirmar que existe una reducción de términos significativa seguida de perdida de documentos.

```
> library("slam")
> term_tfidf <-
+   tapply(myDtm$v/row_sums(myDtm)[myDtm$i], myDtm$j, mean)* log2(nDocs(myDtm)/col_sums(myDtm > 0))
> summary(term_tfidf)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.05032 0.37040 0.57640 0.63900 0.83340 3.81000
> myDtm <- myDtm[,term_tfidf >= 0.57640]
> myDtm <- myDtm[row_sums(myDtm) > 0,]
> summary(col_sums(myDtm))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   3.709   4.000  98.000
> myDtm
<<DocumentTermMatrix (documents: 9664, terms: 9271)>>
Non-/sparse entries: 30722/89564222
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
```

Figura 25. DTM con método term frequency inverse document frequency (TF-IDF)
Fuente: el autor.

Se dispone también del método de remoción de términos escasos dentro de la DTM con la función removeSparseTerms, pasándole como parámetros las DTM con el porcentaje de escasez que se requiere reducir. Este método resultara en la reducción de una gran cantidad de términos de la DTM.

Con los experimentos realizados variando el porcentaje de reducción de escasez de la DTM se trabaja con un porcentaje del 0.9999 de escasez de la DTM. Trabajando con 10329 documentos y 12345 términos.

La DTM resultante luego a haber sufrido una seria de procesamientos se presenta en la Figura 27.

```
> dtms <- removeSparseTerms(myDtmTraining, .9999)
> dtms
<<DocumentTermMatrix (documents: 10329, terms: 9313)>>
Non-/sparse entries: 165261/96028716
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
> dtms <- removeSparseTerms(myDtmTraining, .999)
> dtms
<<DocumentTermMatrix (documents: 10329, terms: 2679)>>
Non-/sparse entries: 138896/27532495
Sparsity           : 99%
Maximal term length: 15
Weighting          : term frequency (tf)
> dtms <- removeSparseTerms(myDtmTraining, .99)
> dtms
<<DocumentTermMatrix (documents: 10329, terms: 276)>>
Non-/sparse entries: 63905/2786899
Sparsity           : 98%
Maximal term length: 12
Weighting          : term frequency (tf)
> dtms <- removeSparseTerms(myDtmTraining, .9)
> dtms
<<DocumentTermMatrix (documents: 10329, terms: 4)>>
Non-/sparse entries: 4697/36619
Sparsity           : 89%
Maximal term length: 5
Weighting          : term frequency (tf)
> dtms <- removeSparseTerms(myDtmTraining, .99999)
> dtms
<<DocumentTermMatrix (documents: 10329, terms: 18544)>>
Non-/sparse entries: 174492/191366484
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
```

Figura 26. DTM resultante del procesamiento del conjunto de datos MovieLens
Fuente: el autor.

3.1.7. Nube de palabras

Una forma de presentar los términos contenidos en la DTM, es atreves de una nube de palabras en la cual se puede notar anomalías en los términos, que quizás se pasaron por alto y debería ser eliminados de la DTM. Para generar la nube de palabras se necesita el paquete “wordwloud” para mayor detalle del paquete revisar(Ian Fellows, 2014), se presentan las palabras con una mayor frecuencia con un tamaño de letra mayor y colores más cálidos mientras que aquellas que tienen menor ocurrencia se presentan con tamaño de letra menor y colores más fríos.

La Figura 28 muestra la nube de palabras de la DTM del corpus de descripciones de películas de MovieLens.

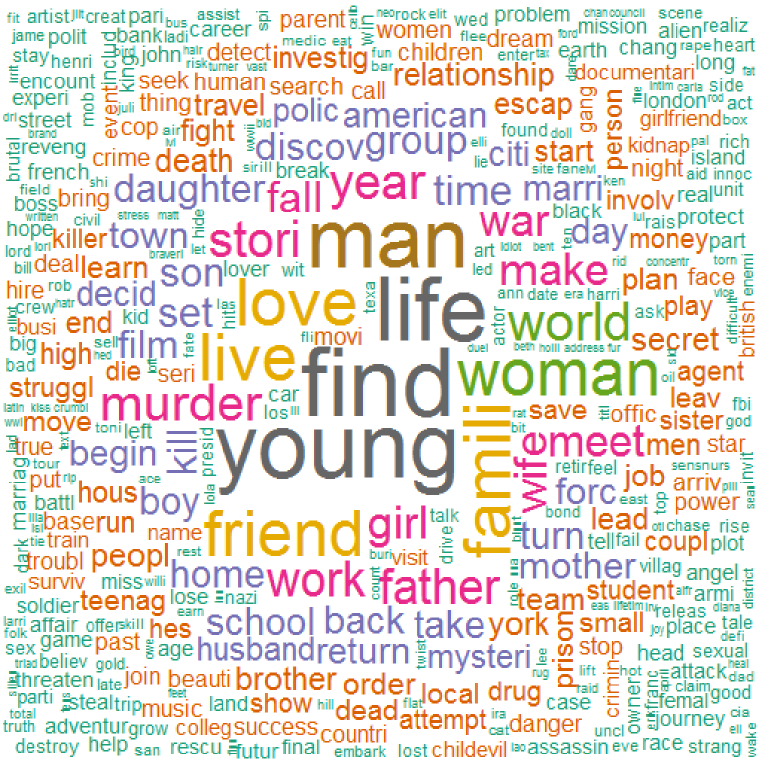


Figura 27. Nube de palabras del conjunto de datos MovieLens.
Fuente: el autor.

Dentro de la Figura 27 se denotan términos usados para narrar el guion de una película, encontramos términos como young, find, stori, entre otros. Además de servir como método de examinación de anomalías en los términos presentes en el conjunto de datos.

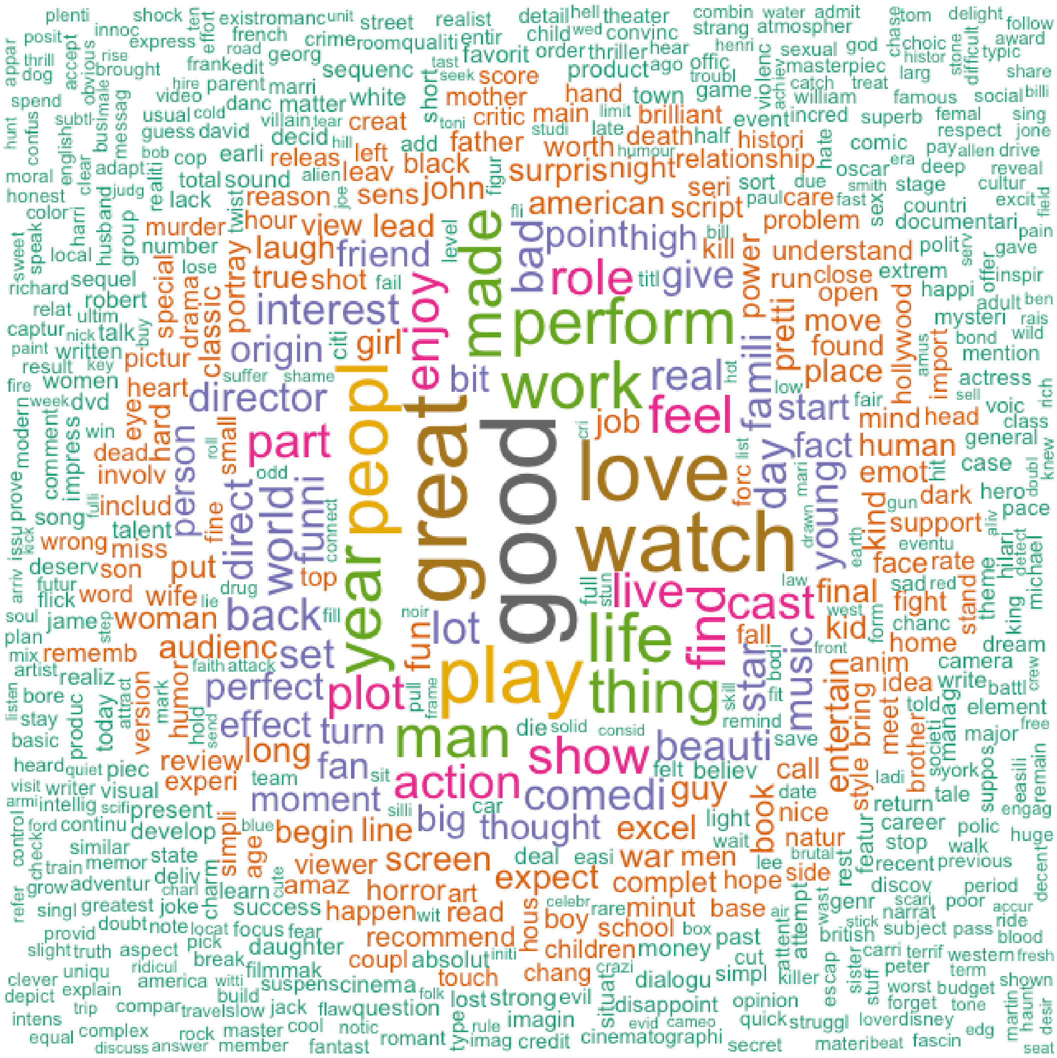


Figura 28. Nube de palabras review MovieLens.
Fuente: el autor.

3.2. Conjunto de datos CMU Movie Summary Corpus

El conjunto de datos está conformado por 42306 de las cuales de seleccionan 5000 descripciones de películas procedentes de Wikipedia para revisar mayor detalle sobre el conjunto de datos revisar(Bamman, O ’connor, & Smith, 2013). Al conjunto de datos se aplica el mismo pre-procesamiento que en el conjunto de datos de MovieLens.

Con la finalidad de comprar los resultados del conjunto de datos MovieLens contra del CMU Movie Summary Corpus, cabe destacar que este conjunto de datos no posee rankings.

La Figura 29 presenta los términos con una frecuencia superior a 2000 contenidos en el conjunto de datos de CMU. Se puede apreciar en la Figura 29 que existe una gran cantidad de términos que sobre pasan la frecuencia de 2000 de aparición dentro del corpus de documentos lo cual podría provocar anomalías en la generación de un modelo, se podría considerar eliminar términos que se utilizan frecuentemente al momento de realizar el guion que sigue una película como “film”, “find”, “kill”, “back”, “tell”, entre otros términos que no aporten significado y que aparecen frecuentemente al momento de realizar una descripción de alguna película.

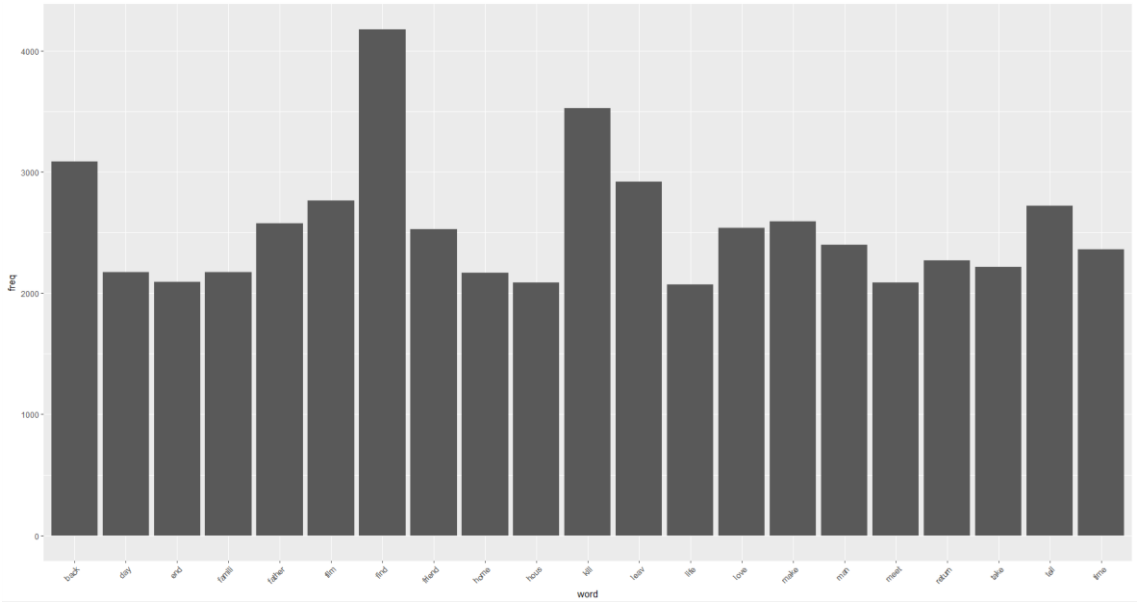


Figura 29. Frecuencia de términos en conjunto de datos CMU Movie Summary Corpus.
Fuente: el autor.

La Figura 30 presenta una gráfica donde se puede hacer un análisis de la cantidad de letras que se encuentran en los términos presentes en el conjunto de datos se puede deducir que términos con 2 letras no aportan significado, por tal razón la cantidad de letras parte de 3 hasta 30 letras ubicando la media en 7 letras. Partiendo de la gráfica presente en la Figura 30 se puede deducir que la cantidad de letras en los términos se puede configurar para que valla desde 3 hasta 15 letras al utilizar términos con más de este rango de letras no tendrían sentido además de no encontrar demasiados términos con una cantidad superior a 15 letras. Esto para limitar la cantidad de términos presentes en la DTM CMU Movie Summary Corpus.

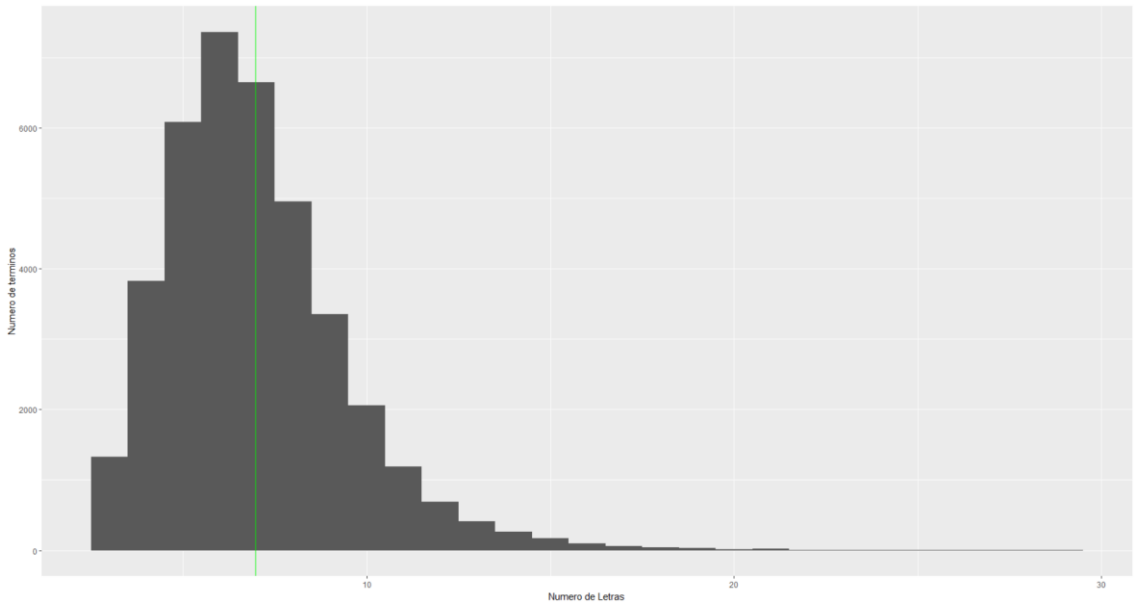


Figura 30. Cantidad de letras en términos en conjunto de datos CMU Movie Summary Corpus.
Fuente: el autor.

Utilizando el análisis de las Figuras 29 y 30 de la DTM del conjunto de datos CMU se concluye que el rango de letras de 3 a 15 para trabajar solo con aquellos términos que cumplan esta condición, los resultados de implementar el rango se presentan en la Figura 31 lado a presenta 5000 descripciones de películas de CMU sin la condición de rango de letras dando una DTM con 38,868 términos con una escasez del 100% teniendo términos con una cantidad máxima de letras de 131. La Figura 31 lado b presenta 5000 descripciones de películas de CMU con la condición de rango de letras de 3 a 15 obteniendo una DTM con 38,355 términos con una escasez del 100% teniendo términos con una cantidad máxima de letras de 15.

a) DTM CMU Movie Summary Corpus sin rango de letras.

```
> save(corpusTrainingPross, file = "resultsPreProcesamiento/myCorpusCMUDiciembre7.Rdata")
> myDtmTraining <- DocumentTermMatrix(
+   corpusTrainingPross)
> myDtmTraining
<<DocumentTermMatrix (documents: 5000, terms: 38868)>>
Non-/sparse entries: 492865/193847135
Sparsity           : 100%
Maximal term length: 131
Weighting          : term frequency (tf)
```

b) DTM CMU Movie Summary Corpus con rango de letras de 3 a 15 en términos.

```
> myDtmTraining <- DocumentTermMatrix(
+   corpusTrainingPross, control=list(
+     wordLengths=c(3,15)
+   )
+ )
> myDtmTraining
<<DocumentTermMatrix (documents: 5000, terms: 38355)>>
Non-/sparse entries: 492325/191282675
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
> save(myDtmTraining, file = "resultsPreProcesamiento/myDtmCMUD5000T38355Diciembre7.Rdata")
```

Figura 31. DTM del conjunto de datos CMU Movie Summary Corpus.
Fuente: el autor.

La Figura 31 presenta una nube de términos construida con la DTM resultante del conjunto de datos CMU donde se puede apreciar aquellos términos que tiene una alta frecuencia dentro de los datos. Se puede hacer una comparación con la nube de términos del conjunto de datos de MovieLens y determinar que existen términos que se encuentran en ambos conjunto de datos esto con la finalidad de observar términos relevantes e irrelevantes para la generación de un modelo con LDA.

Figura 32. Nube de términos de CMU Movie Summary Corpus.
Fuente: el autor.

3.3. Reviews del conjunto de datos MovieLens

Comparando las DTM del conjunto de datos MovieLens con la de CMU Movie Summary Corpus, se puede concluir que la diferencia de términos que maneja la muestra de 5000 películas de CMU es superior, en comparación a la de MovieLens. Por tal razón se busca nutrir el conjunto de datos MovieLens con otro tipo de información más robusta diferente a la descripción.

Analizando la estructura de la página web del sitio IMDB, se encuentran las review de los usuarios que son comentarios emitidos sobre el guion que sigue la película brindando información robusta sobre cada título de película que conforma el conjunto de datos MovieLens. El ANEXO 5 presenta una tabla la cual contiene la comparación con una muestra pequeña entre las descripciones y la review disponible en el sitio de web de IMDB del conjunto de datos MovieLens.

Todo el pre-procesamiento se realiza con el conjunto de datos de MovieLens con la nueva información de reviews de cada película se presenta en el ANEXO 7 se encuentra la Tabla 20. Cabe destacar que simplemente se realiza un análisis y no se detalla, puesto que los detalles son los ya mencionados en el apartado 3.1 del presente trabajo de titulación.

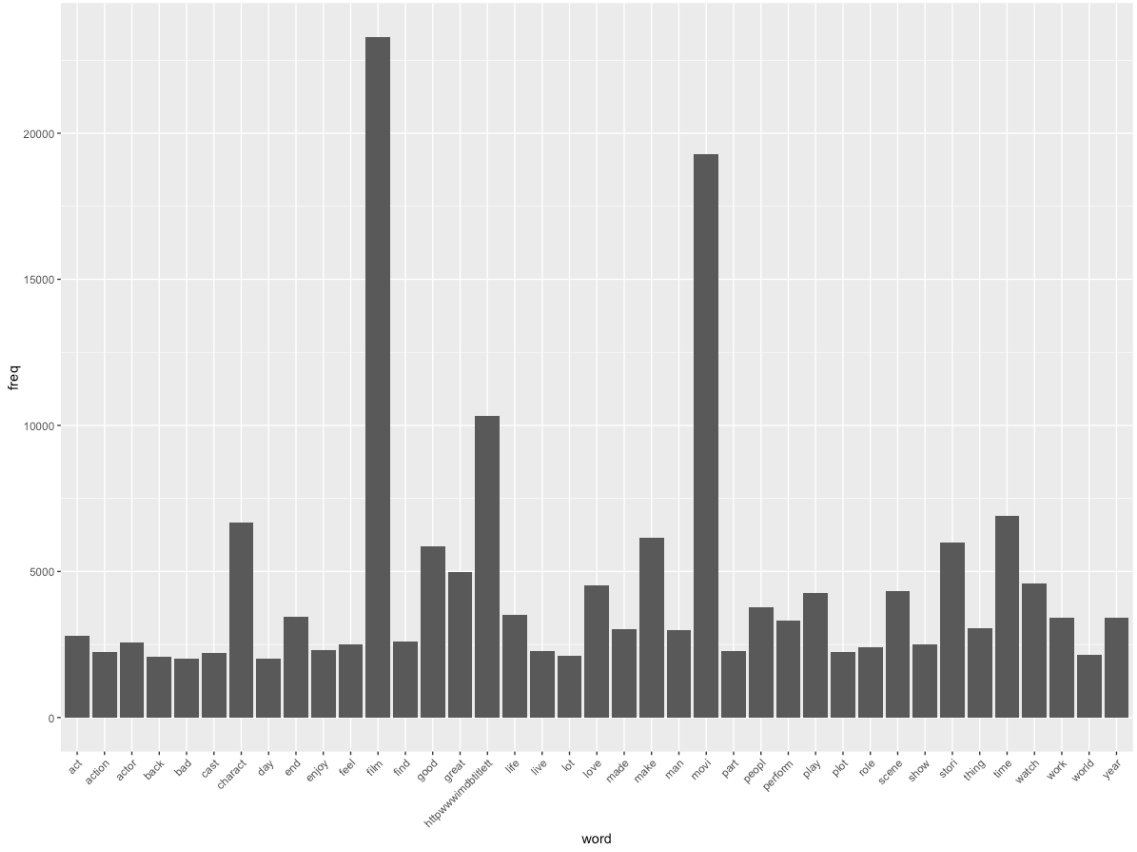


Figura 33. Frecuencia de palabras superior a 2000 de las review del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 33 presenta una gráfica de frecuencia de los términos presentes en el conjunto de datos MovieLens conformado por las reviews. Donde se puede observar que existen términos con alta presencia que no son informativos como: ***httpwwwimdbtitlett, film, stori, make, charact, movi, time, act, actor, end, scene***. Estas palabras se eliminan por no proporcionar información sobre el conjunto de datos MovieLens. El ANEXO 6 presenta la Figura 29 con una distribución de frecuencia de las palabras superior a 1500 de las reviews del conjunto de datos MovieLens.

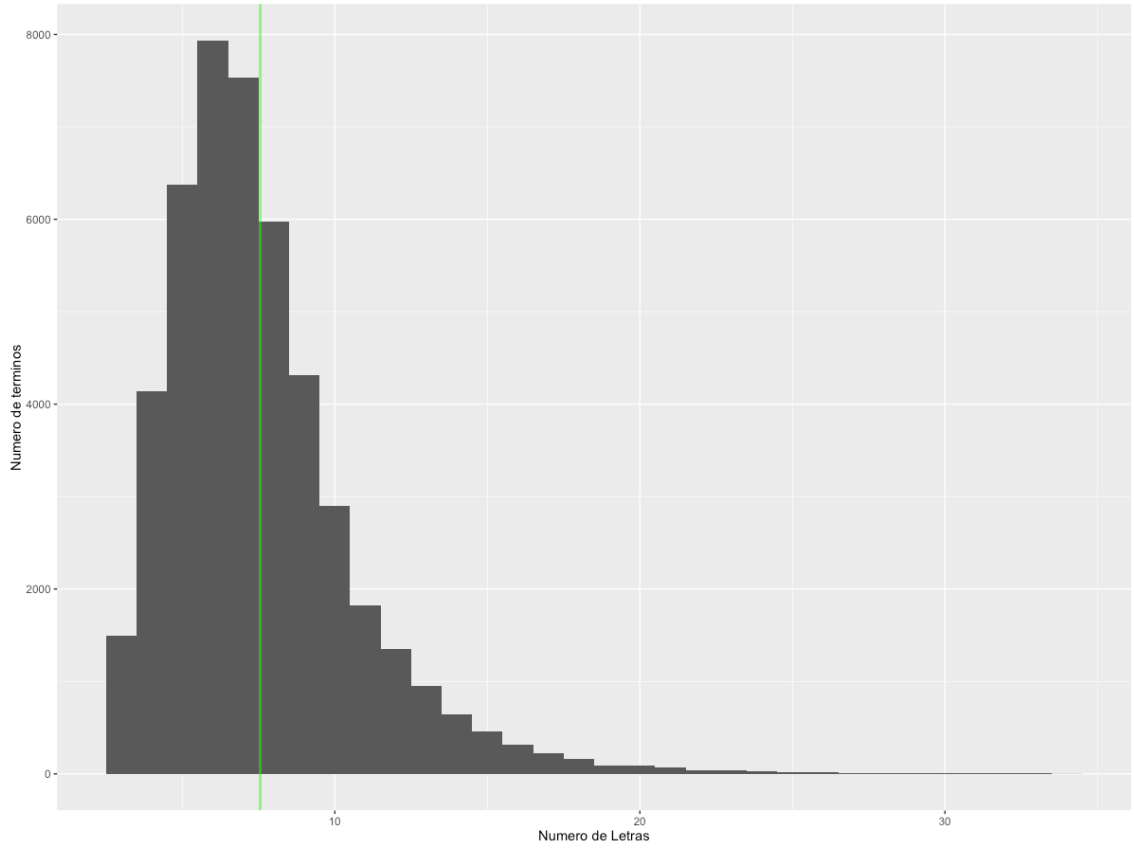


Figura 34. Media de palabras dentro de las Review del conjunto de datos MovieLens.
Fuente: el autor.

La media de palabras se ubica con la línea de color verde en la Figura 35 en 7 letras; pero se trabaja con el rango de entre 3 a 15 letras, siendo en este rango donde se tiene una concentración mayor de las palabras del conjunto de datos de MovieLens.

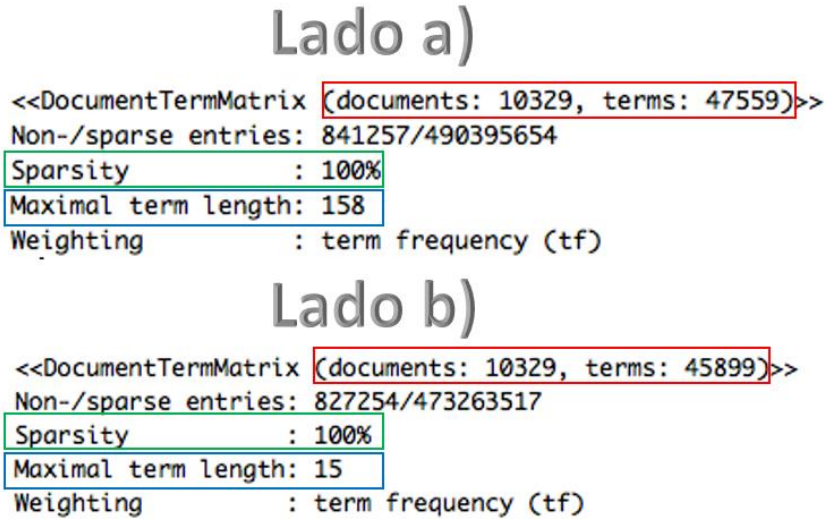
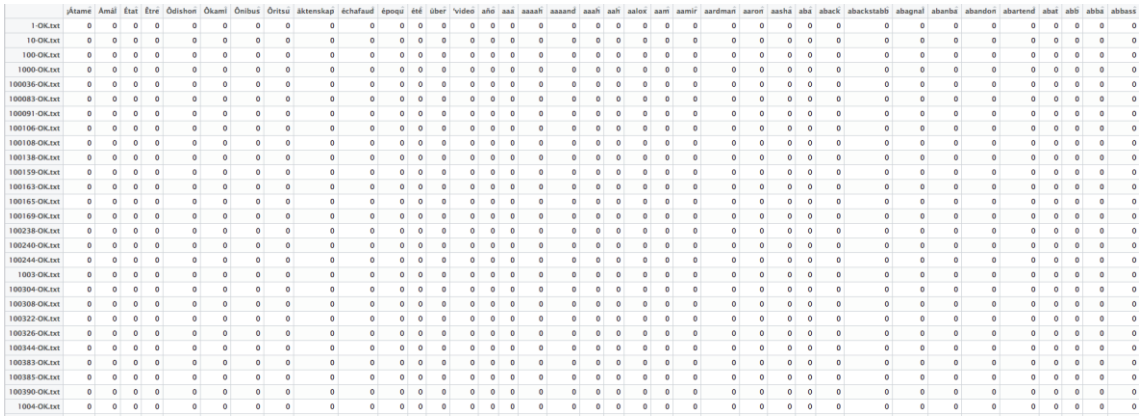


Figura 35. Análisis de DTM de las reviews del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 35 presenta detalles sobre los cambios que sufre la DTM con las reviews del conjunto de datos MovieLens. En la Figura 36 Lado a) presenta la DTM resultante del pre-procesamiento sobre los datos estando formada por 10329 documentos y 47559 términos esta información se encuentra resaltada por el cuadro de color rojo, donde se puede constatar que existen términos formados hasta por 158 letras esta información se encuentra resaltada por el cuadro de color verde, además de tener un 100% de escasez en la DTM esta información se encuentra resaltada por el cuadro de color azul. La Figura 31 Lado b) presenta la DTM aplicando el control de limitar a solo términos conformados por el rango de letras de 3 a 15, obteniendo 45899 términos esta información se encuentra resaltada por el cuadro de color rojo, formados hasta un máximo de 15 letras esta información se encuentra resaltada por el cuadro de color verde y la DTM con un 100% de escasez resaltada por cuadro de color azul.



	¡Átamé	Ámäl	État	Être	Ödishon	Ökaml	Önibus	Öritsü	äktenskap	échafaud	époqu	été	über	'video	año	aaa	aaaah	aaaand	aaah	aah	aalox	aam	aamir	aardman	aaron	aasha	aba	aback	abackstabb	abagnal	abanba	abandon	abartend	abat	abb	abba	abbass
1-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100036-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100083-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100091-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100106-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100108-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100138-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100159-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100163-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100165-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100169-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100238-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100240-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100244-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1003-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100304-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100308-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100322-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100326-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100344-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100383-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100385-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100390-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1004-OK.txt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 36. Muestra de DTM de reviews del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 36 presenta una muestra de la DTM de reviews del conjunto de datos MovieLens donde se puede constatar que está conformada por ceros en sus intersecciones que son el producto de calcular las veces que un término reside en un documento, teniendo una DTM con 100% de escasez.

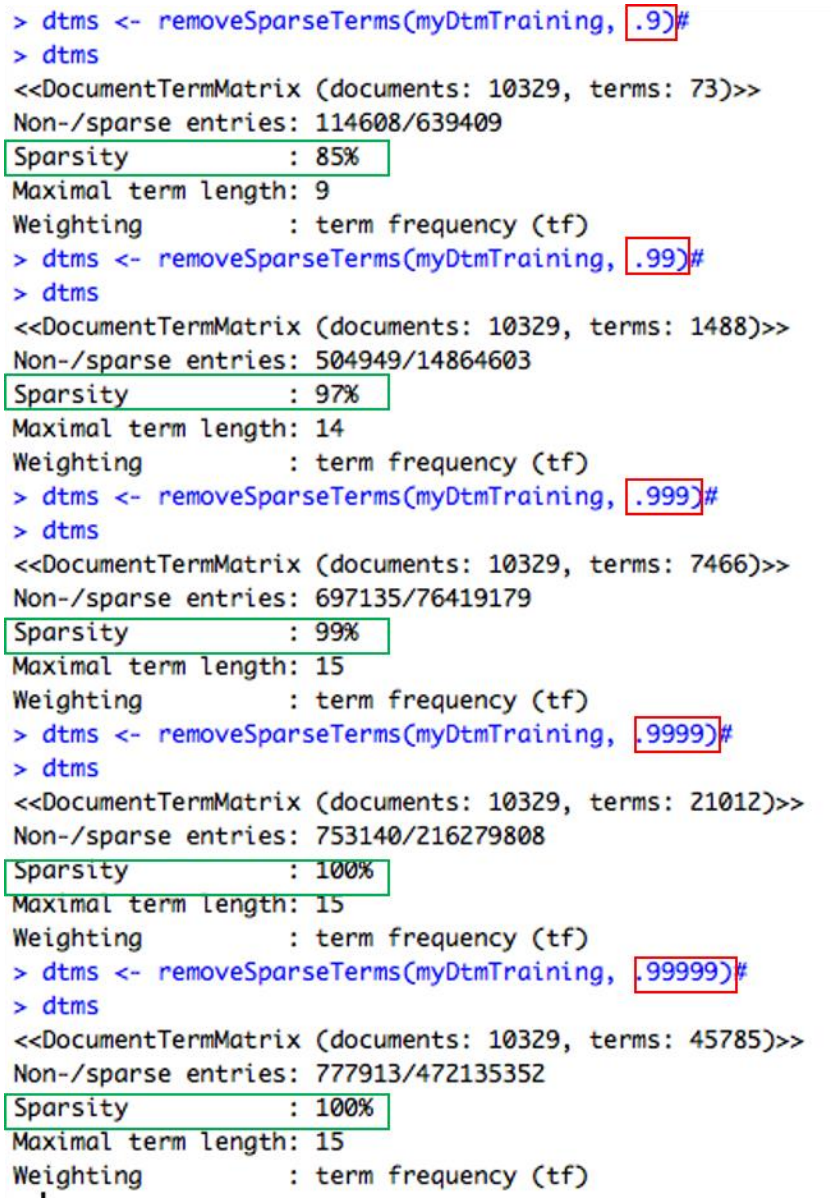


Figura 37. Remoción de escasez de términos en la DTM de reviews del conjunto de datos MovieLens.

Fuente: el autor.

La Figura 37 presenta la aplicación del método de remoción de escasez de términos en la DTM de reviews del conjunto de datos de MovieLens con diferentes porcentajes denotados por un recuadro de color rojo en la Figura 38 el porcentaje de escasez se marca por un cuadro de color verde en la Figura 38 la cantidad de términos varía de acuerdo al porcentaje de escasez que se fija en los porcentajes denotados por el recuadro de color rojo. Con este método de remoción de escasez no se pierden documentos simplemente se reduce aquellos términos poco frecuentes. Se trabaja con un 0.99999 de escasez en los datos.

```
> library("slam")
> term_tfidf <-
+   tapply(myDtm$v/row_sums(myDtm)[myDtm$i], myDtm$j, mean)* log2(nDocs(myDtm)/col_sums(myDtm > 0))
> summary(term_tfidf)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01947 0.08181 0.11910 0.15520 0.18600 2.85700
> myDtm <- myDtm[,term_tfidf >= 0.11910]
> myDtm <- myDtm[row_sums(myDtm) > 0,]
> summary(col_sums(myDtm))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   5.464   5.000 377.000
> myDtm
<<DocumentTermMatrix (documents: 10253, terms: 22810)>>
Non-/sparse entries: 85750/233785180
Sparsity           : 100%
Maximal term length: 15
Weighting          : term frequency (tf)
```

Figura 38. Term frequency inverse document frequency (TF-IDF) en la reviews del conjunto de datos MovieLens.

Fuente: el autor.

La figura 38 presenta los resultados de aplicar el método de TF-IDF en la DTM de reviews del conjunto de datos MovieLens, produciendo perdida de documentos, pero reduciendo la cantidad de términos presentes en la DTM no se emplea este método por la pérdida de documentos.

3.4. Técnica latent dirichelet allocation (LDA)

Existen diferentes algoritmos para implementar Topic Model (en adelante TM), en la Tabla 15 se listan algunas técnicas con su nombre y año de creación:

Tabla 14. Algoritmos disponibles para Topic Model.

Algoritmo	**Año**
Probabilistic Lantent Semantic Analysis (PLSA)	1999
Latent Dirichlet Allocation (LDA)	2003
Dinamic TopicModel	2006
Bigram TopicalMdel	2006
Correlated TopicModel	2007
Bayesian Non-Parametric TopicModel	2007
Supervised TopicModel	2007
Topical N-Gram Model (TNG)	2007
Label Topic	2007
TurboTopics (LDAPD)	2009

Relational TopicModel	2009
Ideal TopicModel	2010
Phrase discovering TopicModel (PDLDA)	2012
TopMine	2015

Fuente: (Contador Pachon, 2015).

En la implementación se selecciona LDA, debido a su gran aplicación en diferentes investigaciones para mayor especificación revisar el ANEXO 1 donde se presenta una tabla con una serie de trabajos e investigaciones desarrolladas con LDA; además de ser el TM más simple, LDA necesita conocer los documentos en forma de una matriz DTM y el número de tópicos "***K"*** a los que hacen referencia los documentos(Contador Pachon, 2015).

El paquete a llevar la experimentación originalmente está escrita en el lenguaje de programación **"C"** por el padre del Topic Model quien es David Blei; pero existe disponible diferentes implementaciones en distintos lenguajes de programación. La Tabla 16 presenta estas variantes de Topic Model en diferentes lenguajes de programación.

Tabla 15. Implementaciones disponibles en distintos lenguajes de programación de Topic Model

Nombre	**Modelo/Algoritmo**	**Lenguaje**	**Autor**	**Notas**
lda-c	Latent Dirichlet allocation	C	D. Blei	This implements variational inference for LDA.
class-slda	Supervised topic models for classifiation	C++	C. Wang	Implements supervised topic models with a categorical response.
lda	R package for Gibbs sampling in many models	R	J. Chang	Implements many models and is fast. Supports LDA, RTMs (for networked documents), MMSB (for network data), and sLDA (with a continuous response).
online lda	Online inference for LDA	Python	M.Hoffman	Fits topic models to massive data. The demo downloads random Wikipedia articles and fits a topic model to them.
online hdp	Online inference for the HDP	Python	C. Wang	Fits hierarchical Dirichlet process topic models to massive data. The algorithm determines the number of topics.

tmve (online)	Topic Model Visualization Engine	Python	A. Chaney	A package for creating corpus browsers. See, for example, Wikipedia.
ctr	Collaborative modeling for recommendation	C++	C. Wang	Implements variational inference for a collaborative topic models. These models recommend items to users based on item content and other users' rankings.
dtm	Dynamic topic models and the influence model	C++	S. Gerrish	This implements topics that change over time and a model of how individual documents predict that change.
hdp	Hierarchical Dirichlet processes	C++	C. Wang	Topic models where the data determine the number of topics. This implements Gibbs sampling.
ctm-c	Correlated topic models	C	D. Blei	This implements variational inference for the CTM.
diln	Discrete infinite logistic normal	C	J. Paisley	This implements the discrete infinite logistic normal, a Bayesian nonparametric topic model that finds correlated topics.
hlda	Hierarchical latent Dirichlet allocation	C	D. Blei	This implements a topic model that finds a hierarchy of topics. The structure of the hierarchy is determined by the data.
turbotopics	Turbo topics	Python	D. Blei	Turbo topics find significant multiword phrases in topics.

Fuente:(D. Blei, C. Wang, J. Chang, n.d.)

Las implementaciones disponibles para el lenguaje de programación R son:

- Paquete “lda”
- Paquete “topicmodels”

A continuación se detalla brevemente cada paquete

En el aparatado 3.4.1 se detalla brevemente el paquete “topicmodels”

3.4.1. Paquete “topicmodel”

El paquete “topicmodels” permite trabajar con diferentes métodos de muestreo ya sea con el método de VEM, que por defecto es el método principal del paquete, como también con el muestreo de Gibbs Sampling este deberá ser especificado para su implementación.

Parámetros para trabajar con LDA, en la creación del modelo se acogen algunas funciones del paquete “topicmodels” con la finalidad de no entrar en demasiados detalles del paquetes solo se describen algunos parámetros los que se utilizan en la creación del modelo con la técnica LDA. Para mayor detalla se recomienda revisar la documentación oficial del paquete(Bettina & Kurt, 2016).

- ESTIMATE.ALPHA, ALPHA, ESTIMATE.BETA: los valores por defecto para estimar el valor de alpha (ESTIMATE.ALPHA = TRUE) y (ESTIMATE.BETA=TRUE). Estos parámetros pueden ser modificados dependiendo de la finalidad del experimento.
- VERBOSE: tiene el valor de 0 por defecto, para no mostrar todas las interacciones del algoritmo si el valor por defecto en alterado se mostrara, cada resultado de las interacciones.
- SAVE y PREFIX: el valor por defecto es 0, para no guardar resultados intermedios.
- KEEP: cuando el valor es un número entero positivo, los valores log-verosimilitud se mantendrán para iteración del algoritmo.
- SEED: permite la reproductibilidad
- Delta: parámetro de las distribuciones previas, para la distribución de los tópicos.
- ITER, BURNIN, THIN: determinan el número aleatorio de muestras para el corpus durante el muestreo de Gibbs. El número de interacciones BURNIN que se establece, indica el número que debe ser descartado antes de cada interacción THIN, luego de este paso se almacenan las interacciones de ITER.
- BEST: el valor por defecto es TRUE, lo cual limita a devolver la mejor probabilidad posterior.

3.4.2. Paquete “lda”

El paquete “lda” toma como entrada el número de documentos, número de términos en el vocabulario y el número de tokens en el conjunto de datos. A diferencia del paquete “topicmodels” que parte de la DTM para más detalles del paquete revisar (Chang, 2015)

CAPITULO IV: ANALISIS DE RESULTADOS

El presente capitulo presenta el análisis de resultados de la implementación del modelo LDA con los parámetros que caracterizan el modelo.

4.1. Determinar número óptimo de k tópicos

Uno de los principales retos que se presenta en la construcción de un modelo con la técnica LDA es la búsqueda del número óptimo de tópicos o temas (en adelante se hace referencia a un término y otro sin modificar su significado), el cual consiste en determinar el valor óptimo de los ***K*** tópicos que caracterizan al conjunto de datos. Este valor es fijado priori para generar un modelo es donde radica el principal reto de LDA. La perplexity es utilizada como un método de evaluación del valor óptimo de tópicos en diferentes investigaciones como(Contador Pachon, 2015; Coronado Matutti et al., 2015; Hofmann & Chisholm, 2016). El análisis del valor obtenido con la perplejidad caracterizara la calidad del modelo ha generado por la técnica LDA.

Los métodos de evaluación del número óptimo de tópicos ***k*** actúan como una guía aproximada para ayudar a reducir el área de enfoque donde se encuentra el valor óptimo de los ***k*** tópicos que caracterizaran al modelo y no significa que se vaya a obtener el valor exacto de los ***k*** tópicos, aquí es donde se requiere el conocimiento de un experto en el dominio de los datos con que se está trabajando(Hofmann & Chisholm, 2016).

El conjunto de datos de MovieLens posee una cantidad de géneros en los que están clasificadas las películas siendo un total de 17 géneros a los que pertenecen las 10,329 películas, se utiliza este valor como referencia para limitar el enfoque donde se encuentra el valor óptimo de k tópicos siguiendo el enfoque empleado en (Bhowmick et al., 2014).

Cross Validation (o validación cruzada en español), se aplica al conjunto de datos con el fin de entender la actuación del modelo siguiendo la método presentado en (Hofmann & Chisholm, 2016; Hornik & Grün, 2011). Haciendo 10 folds Cross validation se utiliza junto a la perplejidad para obtener una idea de cómo se comporta el modelo con diferentes valores de ***K***. La perplejidad consta como un método dentro del paquete de “topicmodels” para más detalles del método de la perplexity revisar(Bettina & Kurt, 2016). En diferentes enfoques implementados para determinar el número óptimo de k tópicos se utiliza 10-Folds-Cross-Validation y 5-Folds-Cross-Validation, donde cada pliegue o fold contiene datos para realizar entrenamiento y pruebas con diferentes partes del conjunto de datos sin repetirse para más detalle de Cross Validation revisar el apartado 2.2.22. Cada fold se compone de un 90% para entrenamiento y 10% para pruebas. El resultado de las interacciones de Cross Validation se promedia (Hofmann & Chisholm, 2016).

Se despliega una serie de pruebas para validar el número de ***k*** tópicos a trabajar se forman pruebas con 5-fold y 10-fold de Cross Validation con valores de ***k*** que va desde 25, 50, 75, 100, 125, 150, 175, 200. Estos valores de ***k*** se utilizan con la finalidad de determinar el número de k tópicos en el conjunto de datos MovieLens observando cómo se comportan los datos con diferentes números de tópicos ***K***.

Se emplea el método es el muestreo de Gibbs Sampling con 10-Fold-Cross-Validation junto con los controles BURNIN = 1000, con lo que el modelo se ejecutara 1000 veces antes de comenzar a guardar cualquier resultado. ITER = 1000, el modelo se ejecutara durante 1000 interacciones y guarda los resultados de cada THIN = 100; el control de BEST = FALSE, con la finalidad de obtener todos los valores de log-likelihoods y no solo aquellos con un valor inferior. De esta manera cada interacción puede ser examinada con los datos pruebas con la métrica de perplejidad o perplexity.

Todos los controles se los utiliza con los pliegues o folds de entrenamiento y pruebas del 10-Folds-Cross-Validation. Sin embargo la excepción es la adición del control de ESTIMATE.BETA en los datos de pruebas. Este se establece en FALSE (de forma predeterminada es TRUE), de tal manera que las distribuciones que se generan en los modelos de entrenamiento se pueden emplear para utilizarlas en los datos de pruebas y calcular la perplejidad.

Los resultados obtenido del log-likelidoods para los pliegues de pruebas del cross-validation. Partiendo de este análisis se puede observar que valor de ***K*** tópicos produce un valor inferior en la perplejidad del modelo. Un valor en la perplejidad mientras menor sea es mejor, caso contrario al tener un valor superior en la perplejidad es malo, debido a que los datos se están dispersando demasiado uno de los otros.

La Figura 31 presente los resultados de 10-Folds-Cross-Validation junto a la perplexity del modelo con diferentes valores de k que van desde 25, 50, 75, 100, 125, 150, 175 y 200. Los resultados presentes en el Figura 31 muestran que valores menores en la perplexity se encuentran entre 25 y 50 valores superiores a este intervalo en la perplexity se disparan y en el análisis de la perplexity solo aquellos valores menores son representativos el análisis se lo realiza con el rango de valores que sea inferior a 25 para analizar los resultados y determinar el valor de ***k*** óptimo.

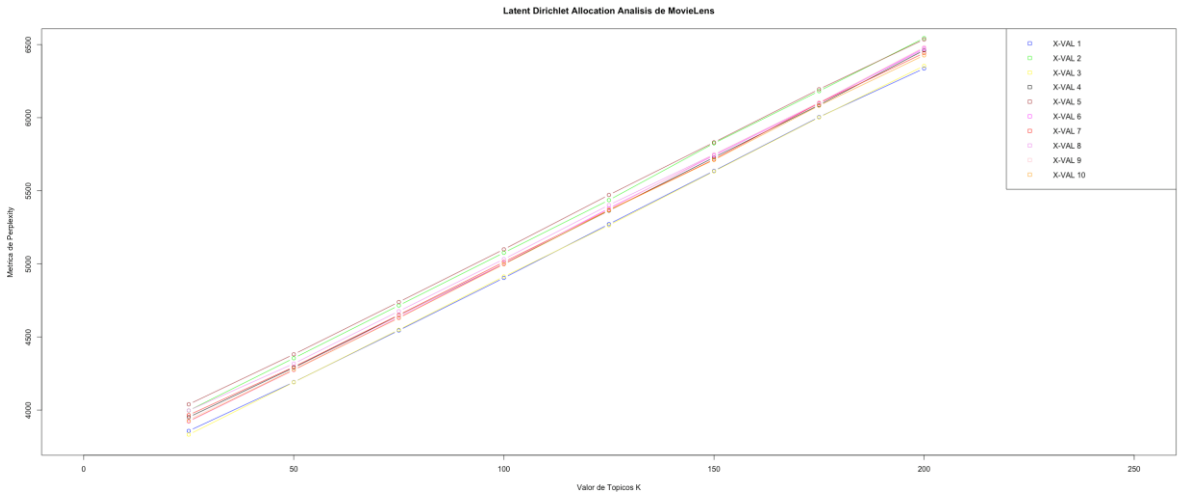


Figura 39. 10-Folds-Cross-Validation con Gibbs junto la Perplexity del conjunto de datos MovieLens con K de 25, 50, 75, 100, 125, 150, 175 y 200.
Fuente: el autor.

La Figura 39 presenta los valores de cada uno de los ***k*** utilizados 25, 50, 75, 100, 125, 150, 175 y 200. Donde se puede cuantificar los valores presentes en el grafica de la Figura 33. El valor menor de la perplexity se encuentra en ***k*** de entre 25 y 50. Utilizando el enfoque presentado en (Bhowmick et al., 2014), donde reducen el foco del número de tópicos a generar utilizando la cantidad de géneros que posee el conjunto de datos en el caso del trabajo de titulación los datos pertenecen a 17 géneros y como los valores presentados en el análisis de perplexity están por encima de este valor, se reduce el área de enfoque a probar con valores comprendidos en el rango de 2 a 24, el valor de ***k*** 1 no se lo analiza debido a que un solo tópico no puede abarcar 18 géneros de películas a los que pertenecen las 10,329 películas.

```
      25       50       75      100      125      150      175      200
3942.624 4285.145 4643.549 5004.369 5370.040 5731.676 6095.465 6452.523
```

Figura 40. Valor de la perplexity en conjunto de datos MovieLens con k de 25, 50, 75, 100, 125, 150, 175, 200, 250.
Fuente: el autor.

El siguiente experimento para determinar el número óptimo de ***k*** se trabaja con valores comprendidos entre 2 a 24 para realizar un análisis sobre los resultados presentados.

La Figura 42 presenta 10-Folds-Cross-Validation con Gibbs junto a la Perplexity del conjunto de datos MovieLens con ***K*** de 2 a 24. Esta gráfica comienza a tener pequeño crecimiento conforme sube el valor de ***k*** por los diferentes Folds pero no una manera brusca como se presenta en el Figura 41 sino suavizada aunque el valor de la perplexity más pequeño lo

podemos ubicar entre 3 y 6. Se puede generar modelos con estos valores de ***K*** y ayudarnos del análisis de un experto en los datos.

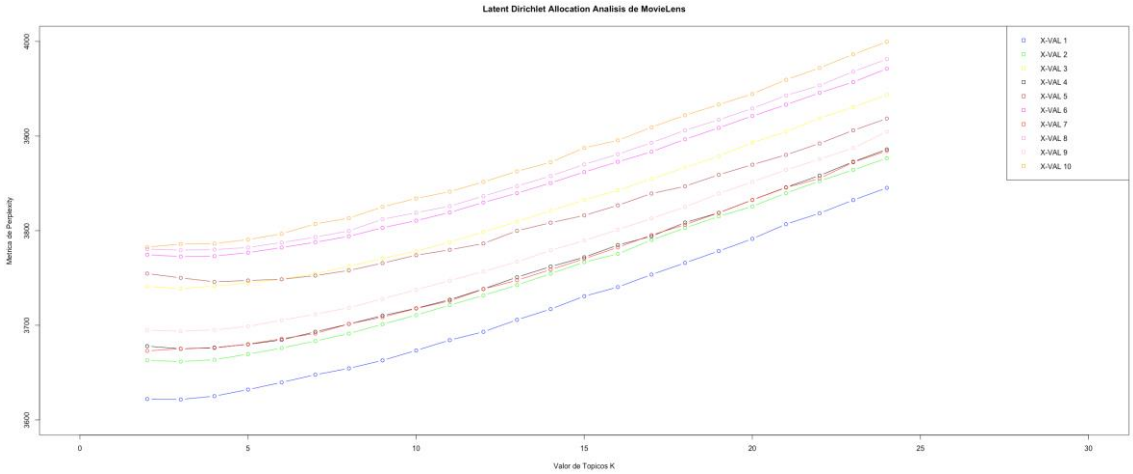


Figura 41. 10-Folds-Cross-Validation con Gibbs junto a la Perplexity del conjunto de datos MovieLens con K 2 a 24.

Fuente: el autor.

La Figura 43 presenta los valores que cuantifican la gráfica presente en la Figura 42. Se puede decir que el valor de ***k*** optimo se encuentra está dentro del rango 3 a 6 tópicos con los que se generar modelos con este número de tópicos. Además de contrastar con algún otro método de validación del número de k tópicos. Se recuerda que este método de evaluación del modelo de perplexity no es exacto si no sirve como una guía aproximada para determinar cuál es el número de k tópicos que generen un modelo con la técnica LDA se requiere de la presencia de un experto en los datos.

K	Perplexity	K	Perplexity
2	3.716.430	13	3.787.238
3	3.715.412	14	3.798.160
4	3.716.330	15	3.809.709
5	3.720.188	16	3.820.234
6	3.725.494	17	3.832.533
7	3.732.219	18	3.844.689
8	3.739.445	19	3.856.720
9	3.748.730	20	3.869.067
10	3.757.325	21	3.882.179
11	3.765.984	22	3.894.162
12	3.776.088	23	3.907.695
K	Perplexity	24	3.921.035

Figura 42. Valor de la perplexity en conjunto de datos MovieLens con k de 3, 5, 10, 15, 20, 25, 30, 50.
Fuente: el autor.

Se realiza experimentos con 5 Folds de Cross Validation con el método variational expectation-maximization (VEM) para analizar la perplexity del modelo de igual forma se emplea la misma cantidad de valores de ***K*** que en el experimento de 10 Folds. Contrastando los resultados de ambos métodos disponibles dentro del paquete “topicmodels”

La Figura 44 presenta una gráfica con el resultado de crear 5-Folds-Cross-Validation con VEM con valores k tópicos de 25, 50, 75, 100, 125, 150, 175 y 200.

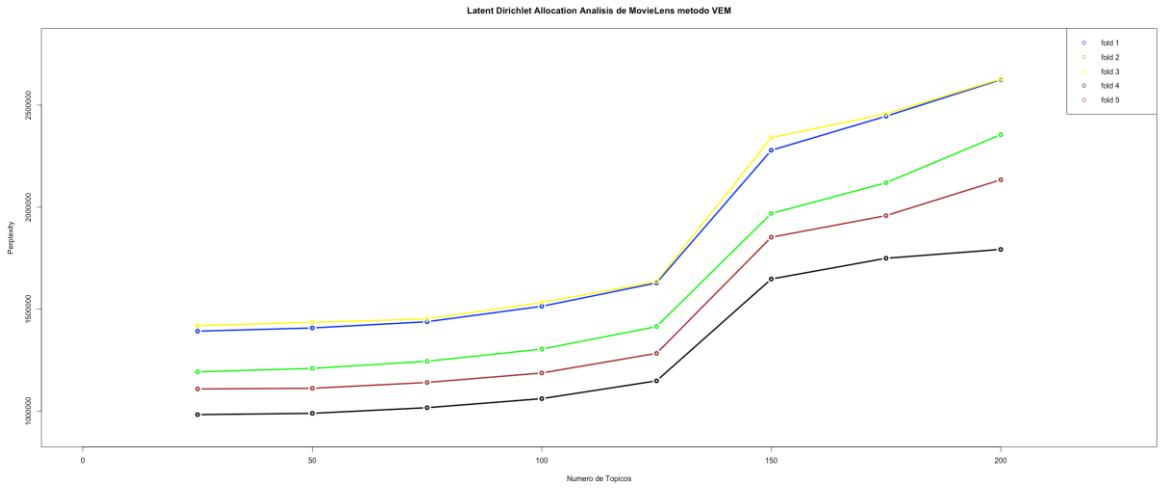


Figura 43. 5-Folds-Cross-Validation con método VEM junto K de 25, 50, 75, 100, 125, 150, 175 y 200.
Fuente: el autor.

```
     25      50      75     100     125     150     175     200
1218488 1230191 1257911 1319184 1421333 2017110 2145448 2306350
```

Figura 44. Valor de la perplexity 5 Folds VEM con K de 25, 50, 75, 100, 125, 150, 175 y 200.
Fuente: el autor.

El valor de la perplexity en los valores que van desde 24 a 200 el que presenta un valor es 25 siendo este el valor óptimo de **k** de tópicos que mejor resultan presenta.

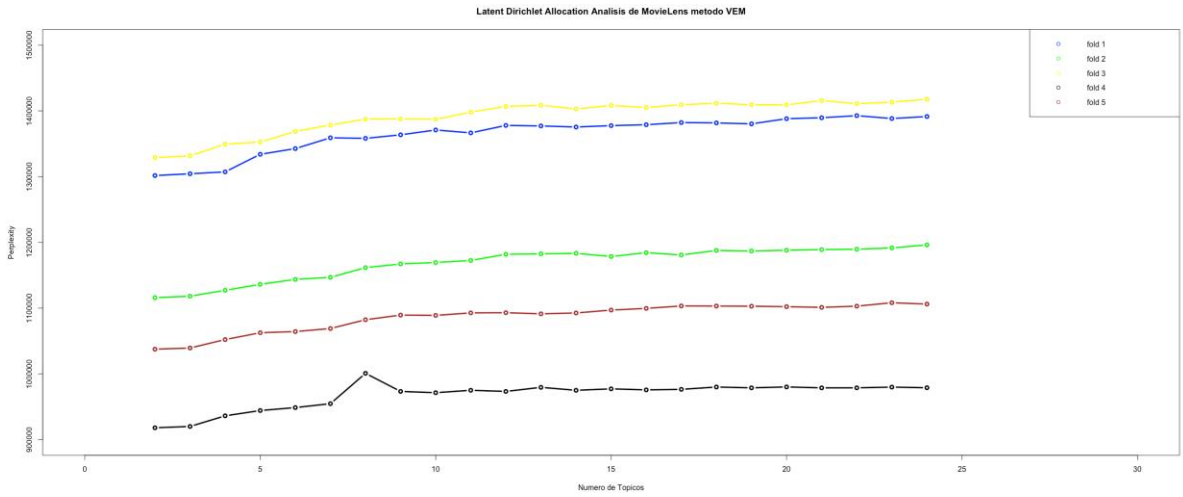


Figura 45. 5-Folds-Cross-Validation VEM con K de 2 a 24.
Fuente: el autor.

La Figura 45 presenta los diferentes valores que presenta la perplexity con un rango de valores de 2 a 24 ubicando el valor menor entre 2 y 3 tópicos.

```
      2       3       4       5       6       7       8       9      10
1140304 1142599 1154316 1165818 1173528 1181518 1197970 1196128 1197479
     11      12      13      14      15      16      17      18      19
1200900 1206498 1207705 1205774 1207634 1208644 1210351 1212750 1211543
     20      21      22      23      24
1213442 1214742 1214940 1216172 1217992
```

Figura 46. Valor de la perplexity con K de 2 a 24.
Fuente: el autor.

La Figura 46 presenta un valor de la perplexity ubicado en 2 siendo este el valor menor entre los tópicos de 2 a 24.

4.2. Marginal likelihoods

Para determinar el número óptimo de ***k*** tópicos existe varios métodos expuestos en (Ponweiser, 2012) entre los que se expone el método Harmonic Mean para mayor detalle de este método de evaluación revisar(Ponweiser, 2012), el método de Harmonic Mean se trata de una prueba estadística que identifica un valor para k tópicos en base a estadísticos generados por los diferentes resultados con ***N*** valores *K* dispuesto constatando cuál de los ***N*** posibles valores de ***k*** es el que contempla todos los documentos que conforman el corpus. Este criterio se ha implementado previamente en diferentes investigaciones de LDA como en (Eliana & Juan Sebastián, 2015; Kovanović, Joksimović, & Gašević, 2015; Ponweiser, 2012); con el resultado obtenido del método se concluirá en un valor de k tópicos este valor será aquel que posea la mayor probabilidad de abarcar toda la información contenida en el corpus. Siguiendo lo expuesto se debe observar que valor ***K*** produce la probabilidad más alta este es el valor de K con que se debe trabajar (Eliana & Juan Sebastián, 2015).

A continuación se expone el brevemente como se descubren los tópicos con LDA.

El algoritmo LDA descubre los tópicos siguiendo las siguientes 3 fases:

- Fase 1. Se debe colocar el número de tópicos ***K*** que se va descubrir. En esta fase se puede utilizar diversos métodos para estimar el valor de ***K*** tópicos, dentro de la literatura, encontraremos 3 métodos de evaluación como se escriben en (Ponweiser, 2012) estos son Performance Measurement on Data, Secondary Tasks y by Human Judgement. El método de evaluación más aceptado Performance Measurement on Data dentro de este nos topamos con los más conocidos en la literatura, como es la perplexity y Harmonic Mean (Eliana & Juan Sebastián, 2015).
- Fase 2. LDA asigna cada uno de los términos a un tópico temporal, el cual se actualiza iterativamente en la fase 3. Esta asignación sigue un criterio de pertenencia a una distribución de Dirichlet, esta distribución permite que un término que aparece más de dos veces en diferentes documentos se asigne a diversos tópicos simultáneamente(Eliana & Juan Sebastián, 2015).
- Fase 3. Iterativa de LDA, revisa y actualiza la asignación de los tópicos recorriendo uno a uno todos los términos presentes en cada documento del corpus. Para cada término, la asignación a un tópico, se realiza con base a dos criterios(Eliana & Juan Sebastián, 2015):
    - ✓ ¿Qué tan predominante es un término a través de los tópicos?
    - ✓ ¿Qué tan predominante son los tópicos en un documento?

Se realiza experimentos con el método de Harmonic Mean con diferentes valores de *k* que van desde 25, 50, 75, 100, 125, 150, 175 a 200.

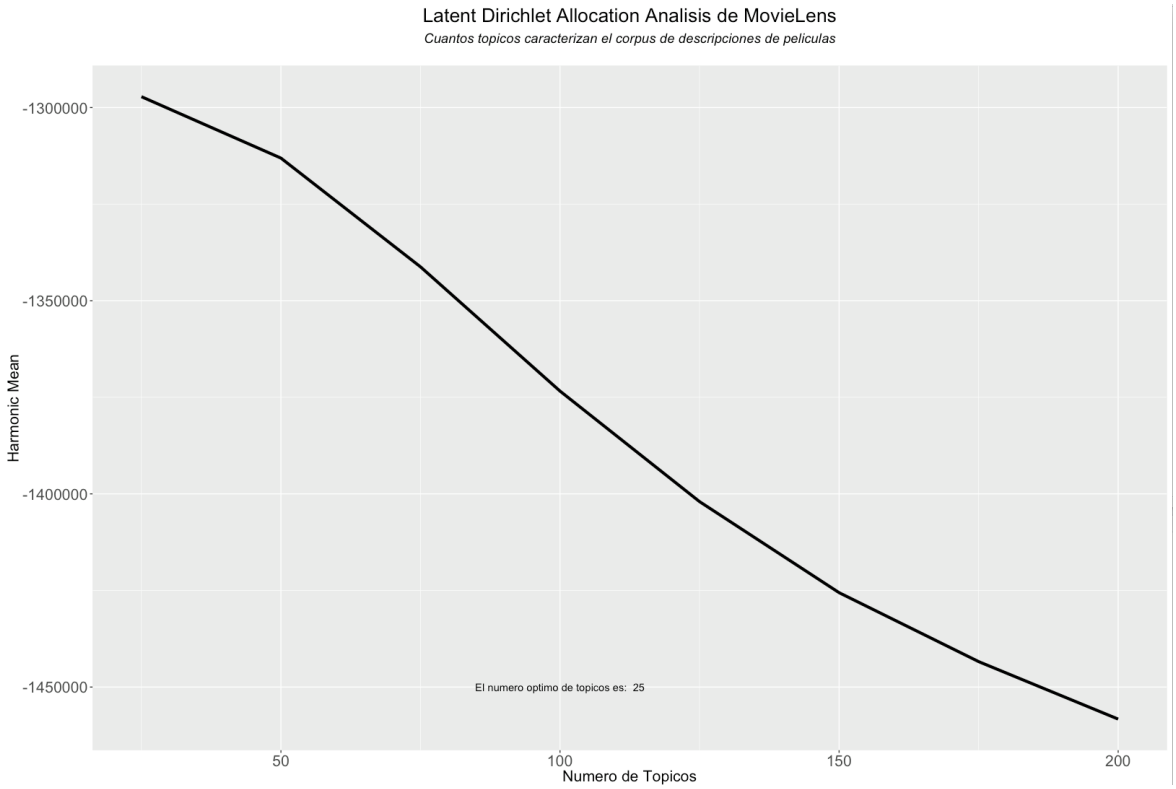


Figura 47. Harmonic Mean descripciones del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 48 muestra los resultados generados con 25, 50, 75, 100, 125, 150, 175 y 200 valores en ***K*** donde se determina que el valor optimo es 25; pero se realiza otro experimento con valores de k de 2 a 24 buscando contrastar los resultados presentados por la perplexity en el apartado 4.1.

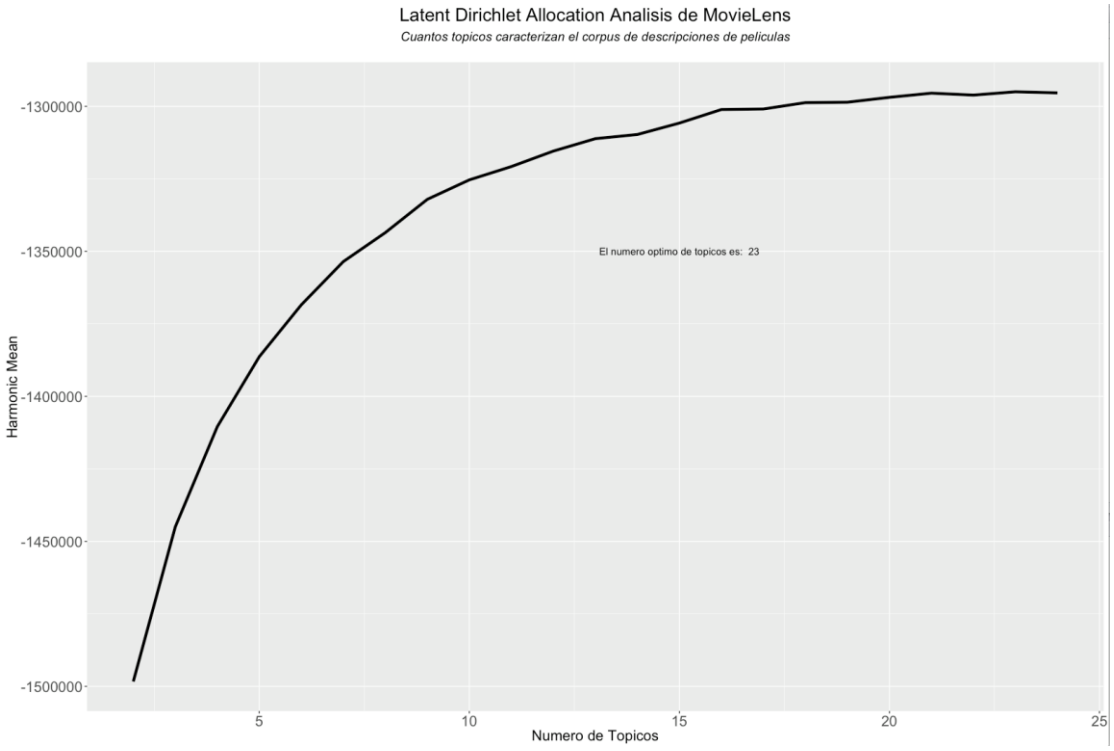


Figura 48. Harmonic Mean rango de 2 a 24 descripciones del conjunto de datos MovieLens. Fuente: el autor.

La Figura 48 presenta los resultados de la Harmonic Mean en el rango de 2 a 24 donde se puede apreciar que el valor de ***K*** óptimo es 23. Con este valor de K se trabaja para la creación del modelo con LDA y analizar los resultados presentados con los diferentes métodos que ofrece el paquete "topicmodels" como VEM y Gibbs.

4.3. Diferentes métodos de evaluación k tópicos enfocados en maximizar y minimizar

La determinación del número óptimo de ***k*** tópicos, es el principal reto que se afronta en la construcción del modelo LDA, este determina la calidad del modelo. Existen diferentes métodos que se han probado e implementado en la literatura. Existiendo diversas formas de evaluar la calidad del modelo como se menciona en el apartado 4.2, también se puede tomar métodos que maximizan y minimizan entre los que se encuentran el valor de ***K*** como:

Minimización:

- Arun2010 descrita en (Arun, Suresh, & Madhavan, 2010).
- CaoJuan2009 descrita en (Cao, Xia, Li, Zhang, & Tang, 2009).

Maximización:

- Deveaud2014 descrita en (Deveaud, SanJuan, & Bellot, 2014).
- Griffiths2004 descrita en (Griffiths & Steyvers, 2004; Ponweiser, 2012)

Este conjunto de métodos de evaluación del valor óptimo de ***k*** se encuentran contempladas dentro del lenguaje de programación R en el paquete “ldatuning”. Este paquete recibe como parámetros de entrada una DTM, una secuencia de valores para determinar el número de ***k*** tópicos con cada método de evaluación disponible en el paquete, también se puede trabajar con los métodos “Gibbs” o “VEM”, además de definir controles sobre el modelo como SEED, BURNIN, ITER, VERBOSE. Su punto fuerte es la posibilidad de trabajar con todos los procesadores disponibles en los computadores que se ejecute con tan solo definir el número de procesadores con los que cuente el equipo con la finalidad de mejorar el tiempo de generación del modelo con paralelismo computacional. Para mayor detalle del paquete revisar (Murzintcev Nikita, 2016).

Los parámetros de configuración con los que se genera el modelo se presentan a continuación:

- SEED = 20080809
- BURNIN = 1000
- ITER = 1000
- K = 25, 50, 75, 100, 125, 150, 175 y 200.
- Método de Gibbs

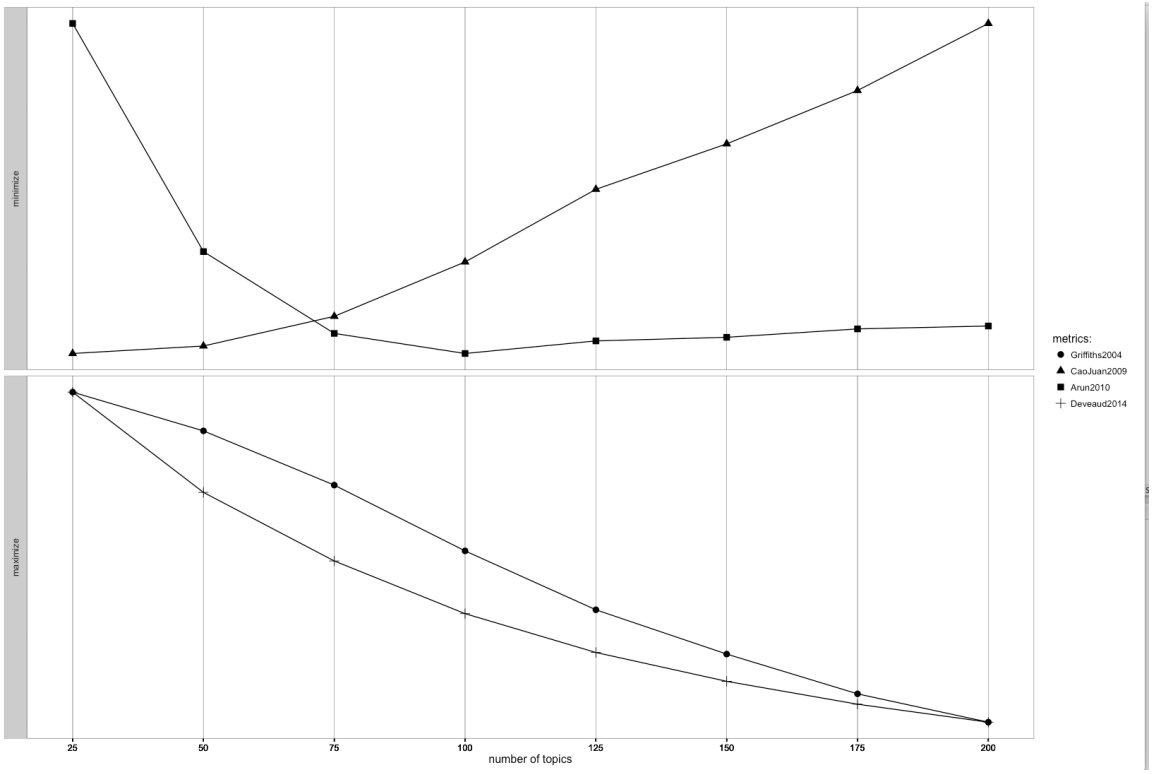


Figura 49. Valor de k tópicos que maximizan y minimizan la calidad del modelo LDA.
Fuente: el autor.

La Figura 50 muestra un grafica dividida en dos partes la primera ubicada en la parte superior de la Figura 50 presenta el valor de ***k*** tópicos que minimizan el error atreves de los métodos de evaluación CaoJuan2009 identificado por la forma de triángulo y Arun201 identificado por la forma de cuadrado, estos dos métodos buscan minimizar el error en la generación del modelo; pero los resultados presentados por el CaoJuan2009 no aporta información al comenzar minimizando el error para comenzar a crecer el error conforme aumenta el valor de ***K***. con el método Arun 2010 es más informativo al comenzar con un error elevado y con una tendencia al minimizar el error conforme aumenta el valor *k* ubicando un valor de 100 tópicos que son los que minimizan el error.

Dentro de la Figura 50 se ubican también los métodos de evaluación que buscan el valor ***k*** que maximiza la calidad del modelo son Griffiths2004 identificado por la forma de circulo y Deveaud2014, ambos métodos de evaluación no aportan información sobre qué valor de ***k*** es el óptimo, debido a que comienzan maximizados ambos métodos y conforme los valores de k crecen hasta llegar a 200 estos métodos decrecen de forma progresiva.

De análisis que se ejerce sobre la Grafica presente en la Figura 50 se puede concluir que el valor de ***K*** se encuentra en 100; pero contrastando los resultados presentados de método Harmonic Mean se observa que el valor de **K** se encuentra en 23 esta valor es aceptado por

estar relativamente ceca a los 17 géneros de películas porque está conformado el conjunto de datos MovieLens.

Se realiza otro experimento con una secuencia de valores de ***k*** que va desde 2 a 24 buscando el valor K óptimo. Con la finalidad de dar validez al valor de k 23, que presenta el apartado 4.2.

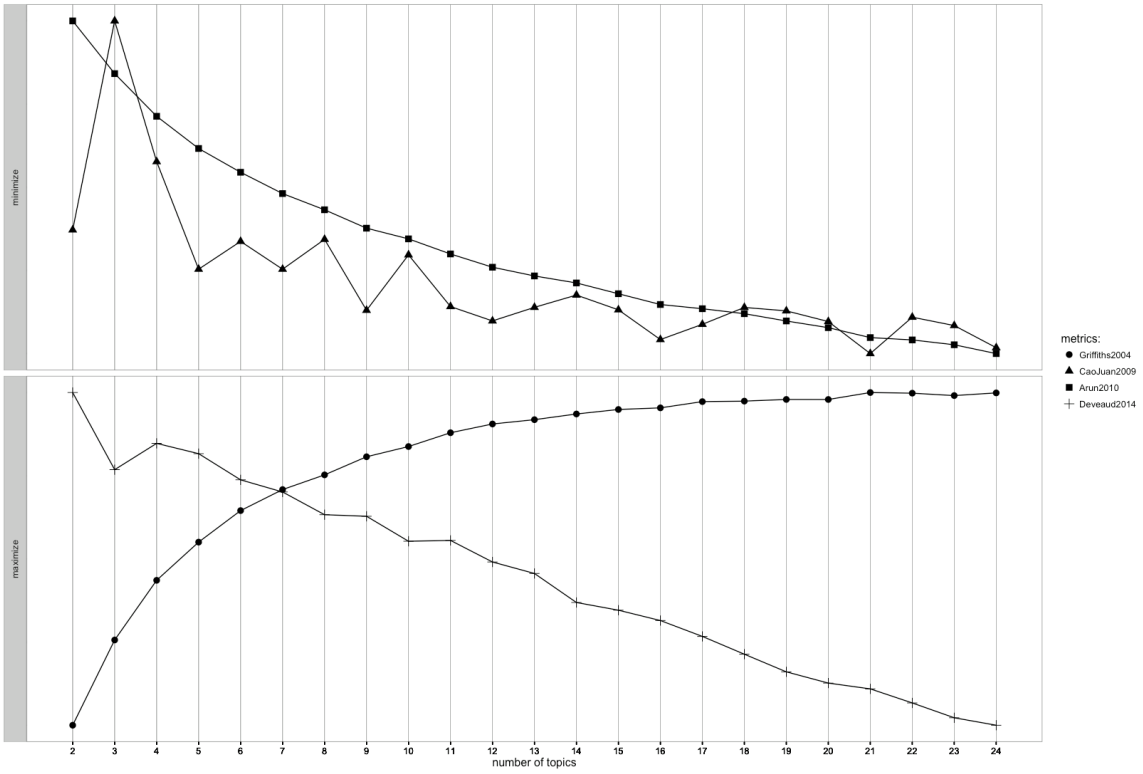


Figura 50. Valor de k tópicos que maximizan y minimizan la calidad del modelo LDA con la secuencia de 2 a 24.
Fuente: el autor.

La Figura 51 presenta los resultados de evaluar el modelo LDA con valores de ***K*** en secuencia de 2 a 24 de las descripciones del conjunto de datos MovieLens. Ubicando el valor de ***k*** en 21. Con este valor de 21 y 23 en ***k*** se genera modelos para analizar los resultados presentes en las matrices Alpha y Betha de las descripciones del conjunto de datos MovieLens.

4.4. Evaluación de resultados con LDA

Con los valores de ***K*** obtenidos con ayuda de los diferentes métodos de evaluación descritos en los apartados del 4.1 al 4.3, se procede a generar modelos con los valores ***k*** de 21 y 23 con la DTM correspondiente a las descripciones del conjunto de datos MovieLens.

4.4.1. Generación de modelo con LDA

Se genera modelo con valores en ***k*** de 21,22 y 23 para analizar la calidad de los tópicos obtenidos, con la finalidad de observar si abarcan toda la información presente en las descripciones del conjunto de datos MovieLens.

Se busca identificar que los tópicos producidos por LDA tengan sentido. Un experto en el dominio del conjunto de datos puede ser necesario para interpretar los resultados. En algunas ocasiones se pueden encontrar palabras en los tópicos extrañas que si la persona que analiza estos términos no posee conocimiento acerca del dominio del conjunto de datos, no comprenderá los resultados por tanto no podrá emitir un juicio sobre estos resultados. Esto presenta un reto para aquellas personas que analizan los resultados si notan que discrepan sobre los resultados tendrá que optar por consultar a un experto en el conjunto de datos, para llevar a cabo una depuración de los términos sobre aquellos que está discrepando discutiéndolo con el experto. Para de esta manera eliminarlos (Hofmann & Chisholm, 2016).

Si existen términos que se deban eliminar del análisis por consiguiente de la matriz DTM, para conseguir una mayor claridad en los términos presentes dentro de los tópicos generado por el modelo LDA (Hofmann & Chisholm, 2016).

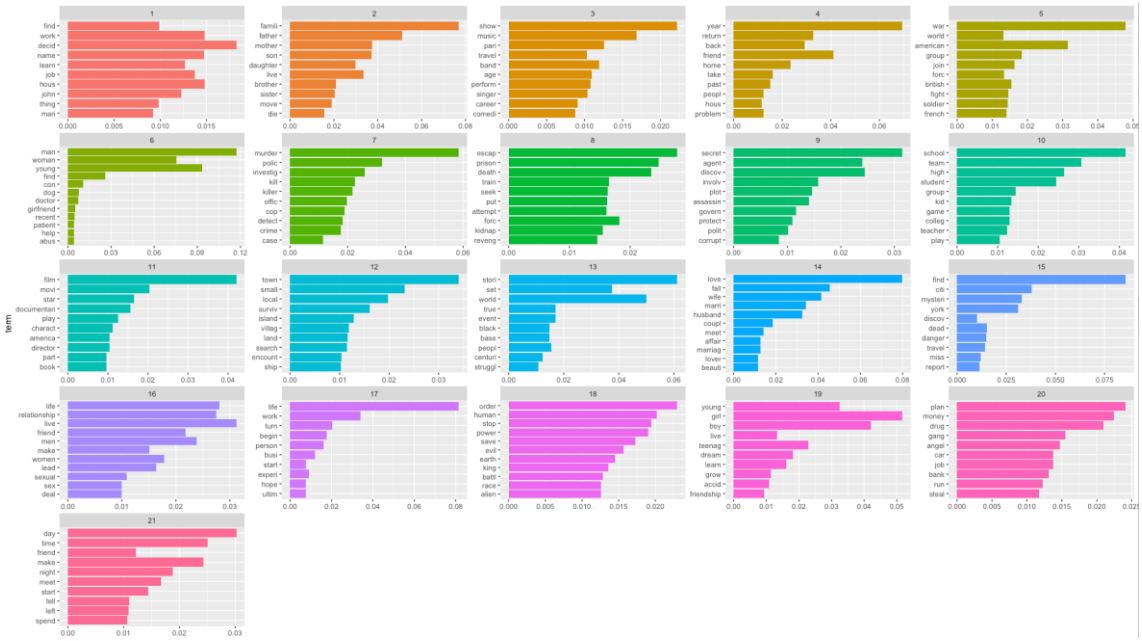


Figura 51. Principales términos de 21 tópicos de las descripciones de películas del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 52 presenta los términos dominantes dentro de las descripciones de películas del conjunto de datos MovieLens. Las probabilidades presentadas atreves de las frecuencias son el valor de Beta de cada termino a partir de cada tópico generado con LDA.

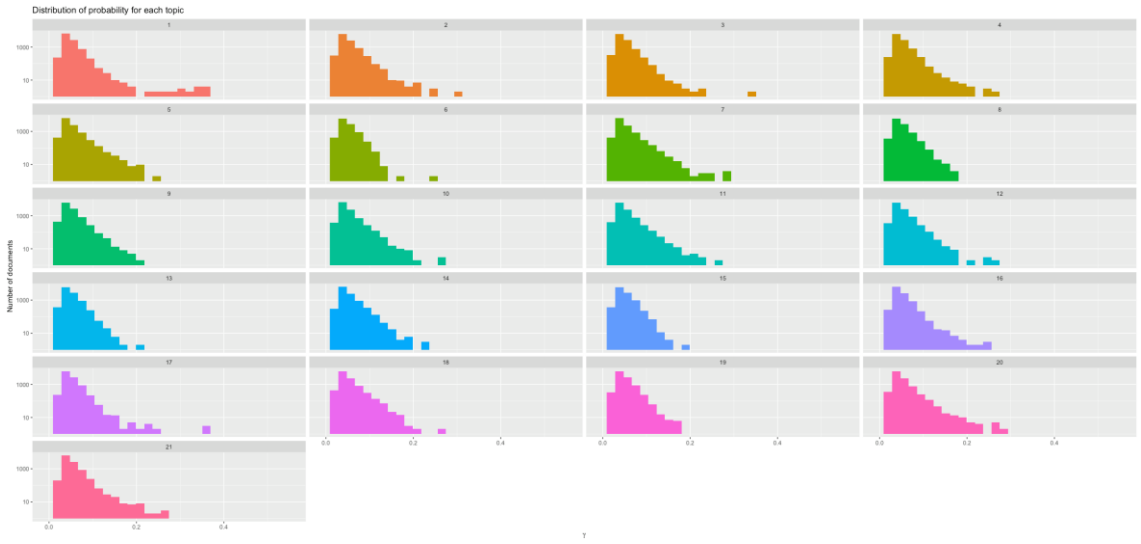


Figura 52. Distribución de probabilidades de 21 tópicos de las descripciones de películas del conjunto de datos MovieLens.
Fuente: el autor.

Se puede presentar el caso en donde los tópicos generados por el modelo se encuentren combinados, lo cual refleja que los términos son similares entre tópicos (o podría ser el caso

que los términos presentes en un tópico abarquen más de un tópico), debido a esto vale la pena crear un modelo con un valor mayor en ***K*** que produzca un mayor cantidad de tópicos como también con valor menor dando resultado a un mejor modelo producido por LDA. En este punto la presencia de una persona experta en el dominio del conjunto de datos es necesaria. En este punto se necesita un poco de ensayo y error en conjunto con la experiencia del experto en el conjunto de datos(Hofmann & Chisholm, 2016).

Una forma de determinar cuando el modelo generado produce tópicos con mayor coherencia es en el listado de términos que acompañan los tópicos, estos términos tiene que estar relacionados semánticamente (esto es más evidente cuando un término esta seguido por otro termino que por lo general se usan ambos para producir una idea global del mismo tópico esto queda más claro con un ejemplo: si se generase un tópico con los términos padre, madre, hijo, hija, entre otros. estos términos por lo general se usan cuando habla de los miembros que conforman una familia siendo este el tópico que contiene estos términos). Dando tópicos con un nivel de identificación claro siempre y cuando los datos sean de un dominio general(Hofmann & Chisholm, 2016).

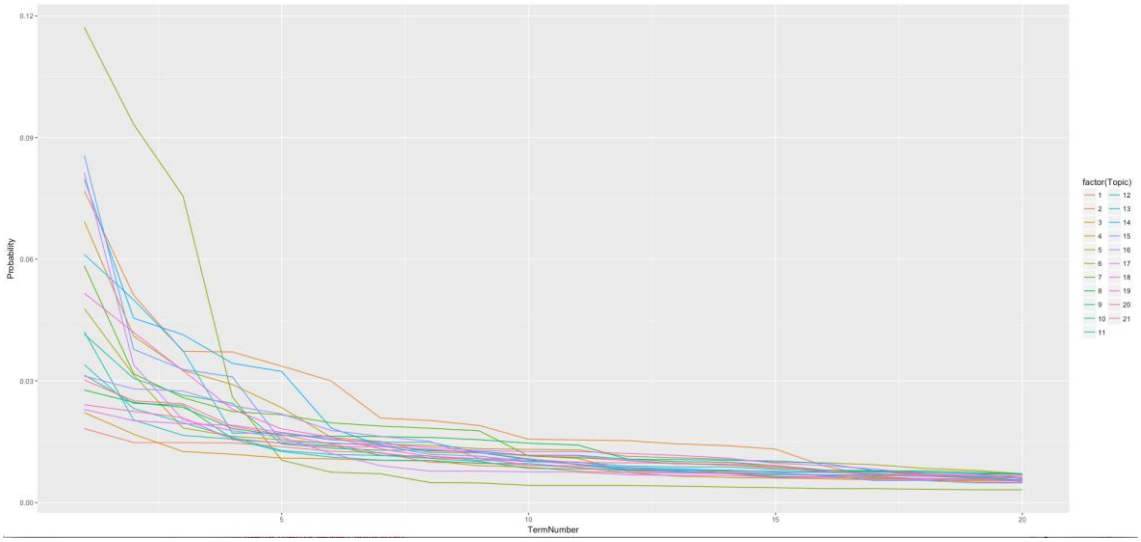


Figura 53. Distribución de Términos en los Tópicos de las descripciones de películas del conjunto de datos MovieLens.
Fuente: el autor.

Las distribuciones de los términos que presenta el modelo LDA con un K de 21 en la Figura 54 donde encontramos en el eje X la cantidad de términos que se producen en los tópicos una probabilidad elevada, se interpreta como si una cantidad menor de términos describen los tópicos generados por el modelo y por tal razón si un documento encaja en un tópico o no sobre la base del vocabulario de los documentos.(Se debe tener presente que cada tópico

asigna una probabilidad a cada termino en el corpus de documentos)(Hofmann & Chisholm, 2016).

Las distribuciones de tópicos se pueden examinar y utilizarse como un control físico en contra de los documentos asignados a cada tópico.

4.5. Visualización de Tópicos

Una forma de evaluar la calidad de los tópicos en de manera visual con ayuda del paquete "LDAvis", se necesita las matriz Alpha y Beta del modelo, para revisar detalles del paquete revisar (Sievert & Shirley, 2015).

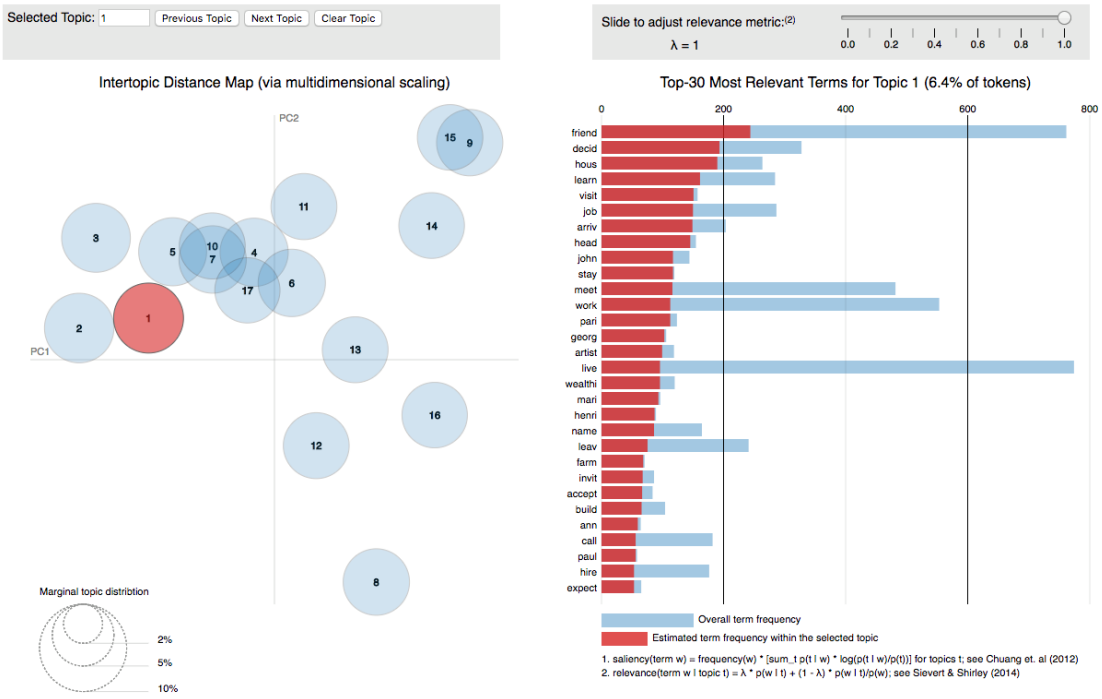


Figura 54. Visualización del modelo generado para evaluar la calidad de los tópicos con el conjunto de datos MovieLens.
Fuente: el autor.

El paquete "LDAvis" calcula las frecuencias de los tópicos, las distancias inter-tópicos y proyecta los tópicos en un plano bidimensional para representar la similitud entre tópicos. Presentando un loops para ajustar lambda entre 0≤λ≤1, que controla como se clasifican los términos en cada tópico, donde los términos se listan siguiendo la disminución de relevancia, la relevancia del termino ***w*** al tópico ***t*** está dado por λ×p (w|t)+ (1−λ) ×p (w|t)/p (w). Los valores cercanos a 1 en lambda (λ) ofrecen altos rankings de relevancia a los términos frecuentes

dentro un tópico determinado, mientras que los valores en lambda cercanos a 0 brindan rankings de alta relevancia a términos exclusivos dentro de un determinado tópico.

4.6. Implementación de un sistema de recomendación de películas con LDA.

Como un resultado de la investigación se desarrolla un sistema de recomendación orientado a recomendar películas en base a las características que estas posee las cuales son obtenidas con LDA, que son las matrices Alpha y Beta, se mide la distancia con la correlación de Pearson entre la similitud de la película que el usuario ha consumido en el pasado y en base a esta se realiza la recomendación de un listado de 10 película similares en un 70%, con cada nueva película que el usuario valla consumiendo la recomendación va mejorando.

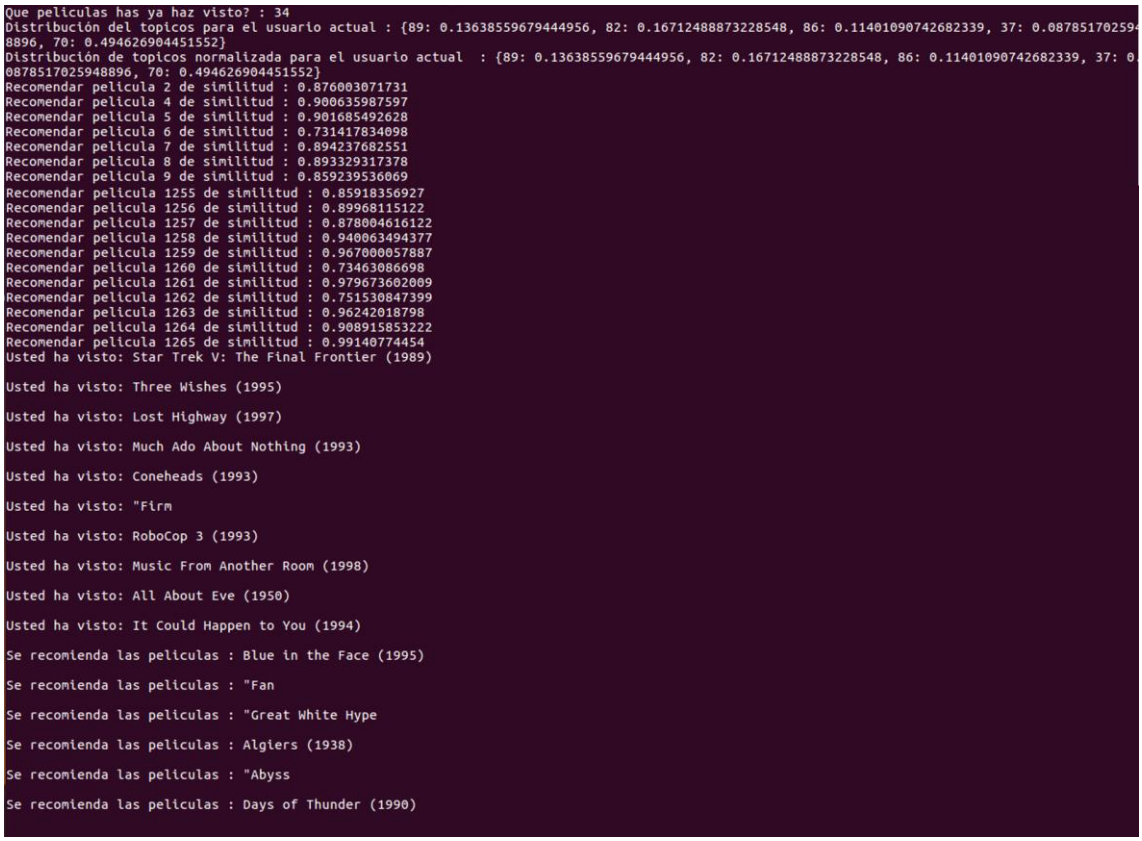


Figura 55. Recomendación de películas en base a las review de las películas del conjunto de datos MovieLens.
Fuente: el autor.

La Figura 55 presenta las recomendaciones de un usuario que ha visto películas como Star Trek: The Final Frontier (1989) y un listado más que se detallan en la Figura 56, emitiendo recomendaciones como Blue in the Face.

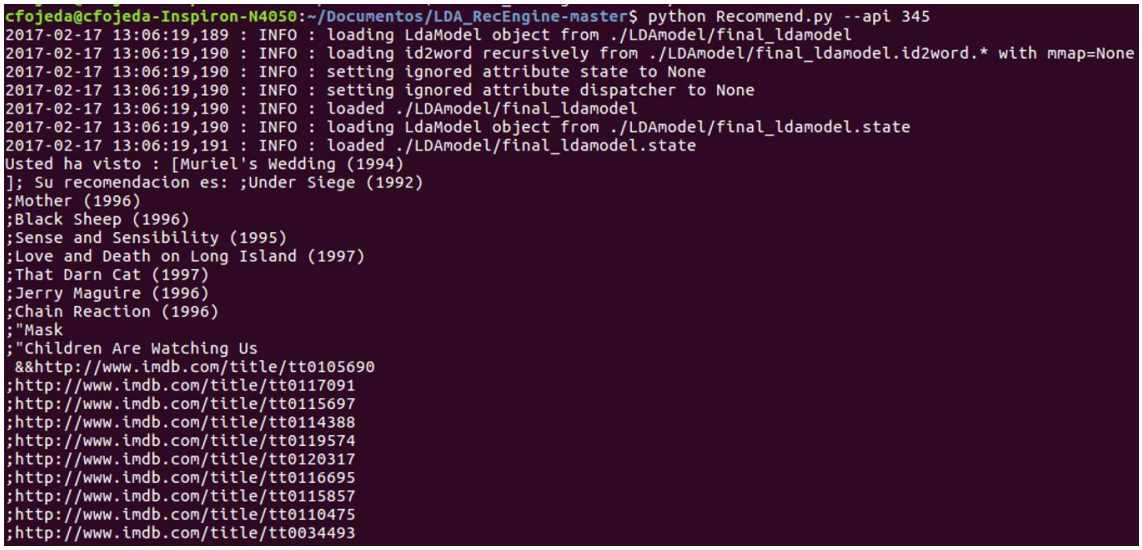


Figura 56. Recomendación de películas con links en sitio web de IMDB sobre el conjunto de datos MovieLens.

Fuente: el autor.

Las recomendaciones que se presentan en la Figura 56, presentan los nombres de las películas junto a los links en el sitio web de IMDB, sobre las películas que forman el conjunto de datos MovieLens.

CONCLUSIONES

Al finalizar el presente trabajo de titulación se concluye lo siguiente:

- En el presente trabajo de titulación se desarrolla un prototipo de extracción de características de películas o temas (tópicos). Se entrena el modelo LDA con un conjunto de descripciones de películas del conjunto de datos MovieLens.
- Los resultados de las descripciones de las películas son satisfactorios; pero necesitan información descriptiva de las películas y mejores métodos que capten la trama que sigue la historia dentro de la película.
- Los temas de películas son características eficientes para los sistemas de recomendación de película, al permitir representar patrones semánticos detrás de las películas. Con las descripciones disponibles de las películas como datos, los temas de películas captan aspectos esenciales de la película, como el género y el estado de ánimo.
- Se concluye que para las descripciones de películas del conjunto de datos MovieLens el valor de K que abarca todas las características de los datos.
- Los tópicos generados con LDA producen términos coherentes manteniendo similitud con el dominio de los datos.

RECOMENDACIONES

Al finalizar el presente trabajo de titulación se recomienda lo siguiente:

- Los temas como explicación en el proceso de recomendación de películas son de mucha utilidad, pero necesita ser ajustado con la capacidad de evaluar temas individuales. Los temas de películas clasificadas por los usuarios se pueden utilizar como retroalimentación al sistema mejorando la calidad de las recomendaciones.
- En el presente trabajo de titulación, se considera las descripciones de películas disponibles en el sitio IMDB para extraer características. Tal método se puede extender o combinar con otros metadatos de las películas como:
  - ✓ Trama
  - ✓ Genero
  - ✓ Palabras clave.
- Con el reciente avance en aprendizaje profundo, sería interesante estudiar el efecto de la combinación de LDA como algunos pre-procesamientos de análisis profundo en las descripciones de películas.
- El pre-procesamiento de las descripciones de películas debido a que LDA no toma en cuenta el orden de las palabras, como es fácilmente visible, este orden en las palabras en algunos casos tiene peso, especialmente para palabras clave de películas conformado por bigramas como “comedia oscura” o “horror nórdico”.
- Extraer y emplear la construcción de las descripciones de películas tiene el potencial de captar aún mejor la semántica de la película.
- Construcción de Topic Models con diversas variantes disponibles la técnica LDA puede considerarse como un modelo base para construir modelos más complejos sobre esta base de LDA, atendiendo complejas necesidades de los datos disponibles. CTM Y DTM son modelos complejos que se construyen partiendo de LDA. En el caso de DTM se puede emplear para observar los patrones cambiantes de las películas con el pasar del tiempo. Concretamente se puede utilizar series de programas de televisión que poseen varias temporadas emitidas don DTM se podría destacar el aumento y el declive de los protagonistas en el transcurso de las temporadas.
- Topic Model se puede ampliar incluyendo información adicional, como los metadatos. Por ejemplo modelos de autor-tema adjuntando proporciones de los temas a los autores, haciendo posible calcular la semejanza del autor en base a proporciones del tema.

- Los modelos jerárquicos de LDA son otra dirección a explorar mientras que extender centenares de temas a millares podría representar una amplia gama de géneros de películas.
- La recomendación basadas en Topic Model que gusten a los usuarios y los temas de calificación en si son algunas formas de mejorar los temas extraídos y construir un sistema basado en Topic Model.
- Con las diversas formas de contenido disponible, el reto es extraer de forma eficiente las características de todas las formas de metadatos, recomendar contenido relevante al usuario final y mantener la serendipia en las recomendaciones.

BIBLIOGRAFÍA

A Probabilistic Approach for Recommendation Looking at Collaborative Topic Regression as introduced. (n.d.). Retrieved from https://www.kma.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_KMA/Probabilistic_Recommendation.pdf

Alvarado, V. A. (2015). Clasificación de artículos científicos. Retrieved from http://repositorio.udec.cl/handle/11594/1952

Arun, R., Suresh, V., & Madhavan, C. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. *Pacific-Asia Conference*. Retrieved from http://link.springer.com/10.1007%252F978-3-642-13657-3_43

Bamman, D., O 'connor, B., & Smith, N. A. (2013). Learning Latent Personas of Film Characters, 352–361.

Bettina, G., & Kurt, H. (2016). Package "topicmodels." Retrieved from https://cran.r-project.org/web/packages/topicmodels/topicmodels.pdf

Bhargav, S. (2014). *Efficient Features for Movie Recommendation Systems*. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:759691

Bhowmick, A., Prasad, U., & Kottur, S. (2014). *Movie Recommendation based on Collaborative Topic Modeling*. Retrieved from https://satwikkottur.github.io/reports/F14-ML-Report.pdf

Bisgin, H., Liu, Z., Fang, H., & Xu, X. (2011). Mining FDA drug labels using an unsupervised learning technique-topic modeling. *BMC*. Retrieved from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-S10-S11

Blei, D. M. (2012). *Introduction to Probabilistic Topic Modeling*. *Communications of the ACM* (Vol. 55). http://doi.org/10.1145/2133806.2133826

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. http://doi.org/10.1016/j.knosys.2013.03.012

Bokde, D., Girase, S., & Mukhopadhyay, D. (2014). Role of matrix factoriation model in collaborative filtering algorithm. *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, *1*(6).

Cao, J., Xia, T., Li, J., Zhang, Y., & Tang, S. (2009). A density-based method for adaptive LDA model selection. *Neurocomputing*. Retrieved from http://www.sciencedirect.com/science/article/pii/S092523120800372X

Chang, J. (2015). Package "lda" Title Collapsed Gibbs Sampling Methods for Topic Models. Retrieved from https://cran.r-project.org/web/packages/lda/lda.pdf

Chong Wang, D. M. B. (2011). Collaborative Topic Modeling for Recommending Scientific Articles. *Kdd'11*, 448–456. http://doi.org/10.1145/2020408.2020480

Contador Pachon, S. (2015). Clasificación De Textos Científicos Con R. In *VII JORNADAS DE USUARIOS DE R*. Madrid. Retrieved from http://r-es.org/7jornadasR/ponencias/sergio_contador_pachon.pdf

Coronado Matutti, M. A., Cárdenas Acosta, R., Bello Medina, K., & Carrasco Rodríguez, J. L. (2015). "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE DE INFERENCIA DE PROGRAMAS DE ESPECIALIZACIÓN EN INGENIERÍA USANDO TOPIC MODELING. LIMA – PERÚ.

D. Blei, C. Wang, J. Chang, M. H. (n.d.). Topic Modeling Software. Retrieved February 7, 2017, from http://www.cs.columbia.edu/~blei/topicmodeling_software.html

Deveaud, R., SanJuan, E., & Bellot, P. (2014). Accurate and effective latent concept

modeling for ad hoc information retrieval. *Document Numérique*. Retrieved from http://www.cairn.info/revue-document-numerique-2014-1-page-61.html

Eliana, S., & Juan Sebastián, O. M. (2015). Application of topic modeling for Trauma Studies: The case of Chevron in Ecuador. *Investigación Y Desarrollo*. Retrieved from http://www.scielo.org.co/scielo.php?pid=S0121-32612015000200001&script=sci_arttext&tlng=en

Feinerer, I., Hornik, K., & Feinerer, M. (2015). Package "tm." *Corpus*. Retrieved from https://safesteps.com/wp-content/uploads/2014/04/tm.pdf

Feng Tan, Li Li, Zeyu Zhang, and Y. G. (2014). A Multi-attribute Probabilistic Matrix Factorization Model for Personalized Recommendation, 535–539. http://doi.org/10.1007/978-3-319-21042-1

González, Á. C. (2013). *Recomendación de Contenidos Digitales basada en divergencias del lenguaje Diseño , Experimentación y*. UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA. Retrieved from http://e-spacio.uned.es/fez/eserv/bibliuned:master-ETSIInformatica-LSI-Acastellanos/Castellanos_Angel_TFM.pdf

Griffiths, T., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National*. Retrieved from http://www.pnas.org/content/101/suppl_1/5228.short

Hackeling, G. (2014). Mastering Machine Learning with scikit-learn. Retrieved from https://books.google.com/books?hl=es&lr=&id=fZQeBQAAQBAJ&oi=fnd&pg=PT7&dq=%2522Mastering+Machine+Learning+with+scikit-learn%2522&ots=wtq27JoVLU&sig=gNn5O77ylW2s3ca6_LYTa_9xFMw

Hernando, A., Moya, R., Ortega, F., & Bobadilla, J. (2013). Hierarchical graph maps for visualization of collaborative recommender systems. *Journal of Information Science*, *40*(1), 97–106. http://doi.org/10.1177/0165551513507407

Hofmann, M., & Chisholm, A. (2016). *Text Mining and Visualization: Case Studies Using Open-Source Tools*. (M. Hofmann & A. Chisholm, Eds.). http://doi.org/13: 978-1-4822-3758-0

Hornik, K., & Grün, B. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*. Retrieved from http://epub.wu.ac.at/3987/

Hu, W. (2012). Unsupervised Learning of Two Bible Books: Proverbs and Psalms. *Sociology Mind*. Retrieved from http://file.scirp.org/Html/20943.html

Ian Fellows. (2014). Word Clouds. Retrieved from https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf

Jung, Y., & Hu, J. (2015). AK-fold averaging cross-validation procedure. *Journal of Nonparametric Statistics*. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/10485252.2015.1010532

Kovanović, V., Joksimović, S., & Gašević, D. (2015). Content Analytics: the definition, scope, and an overview of published research. In. *Handbook of*. Retrieved from http://vitomir.kovanovic.info/public/papers/Kovanovic et al. - 2015 - Content Analytics (preprint).pdf

Kurt Hornik, David Meyer, & Christian Buchta. (2016). Sparse Lightweight Arrays and Matrices.

Lee, H., & Kwon, J. (2015). Improvement of Matrix Factorization-based Recommender Systems Using Similar User Index. *International Journal of Software Engineering and Its Applications*, *9*(3), 71–78. http://doi.org/http://dx.doi.org/10.14257/ijseia.2015.9.3.08

Li, F., Li, G., Yao, B., Hwang, S., & Zhang, Z. (2014). Learning to Recommend with User Generated Content, *1*, 221–232. http://doi.org/10.1007/978-3-319-21042-1

Moya Garcia, R. (2015). Hierarchical Graph Maps for Visualization of Collaborative

Recommerder Systems.
Murzintcev Nikita. (2016). Tuning of the Latent Dirichlet Allocation Models Parameters. Retrieved from https://github.com/nikita-moor/ldatuning
Orii, N. (2012). Collaborative Topic Modeling for Recommending GitHub Repositories.
Ortega, F. (2013). Sistemas de Recomendación. Retrieved from http://www.lpsi.eui.upm.es/~fortega/docs/recsys_nov_2013.pdf
Ortega, F. (2015). *Incorporating group recommendations to recommender systems: Alternatives and performance*. *Information Processing and Management*. Universidad Politécnica de Madrid Escuela. http://doi.org/10.1016/j.ipm.2013.02.003
Ortega, F., Bobadilla, J., Hernando, A., & Gutiérrez, A. (2013). Incorporating group recommendations to recommender systems: Alternatives and performance. *Information Processing and Management*, *49*(4), 895–901. http://doi.org/10.1016/j.ipm.2013.02.003
Ortega, F., Hernando, A., Bobadilla, J., & Kang, J. H. (2016). Recommending items to group of users using Matrix Factorization based Collaborative Filtering. *Information Sciences*, *345*, 313–324. http://doi.org/10.1016/j.ins.2016.01.083
Ortega Requena, F. (n.d.). Incorporating Group Recommendations to Recommender Systems : Alternatives Índice. Retrieved from http://www.lpsi.eui.upm.es/~fortega/docs/fortega_phd_thesis_presentation.pdf
Ortega Requena, F. (2015). Incorporating Group Recommendations to Recommender Systems: Alternatives and Performance. Madrid. Retrieved from http://www.lpsi.eui.upm.es/~fortega/docs/fortega_phd_thesis_presentation.pdf
Pablo Castells, Fernando Díez, E. P. (2011). Recuperacion y almacenamiento de informacion en la web 4. Sistemas de recomendación.
Parra, D. (2015a). Factorización Matricial en Sistemas Recomendadores Clase de Introducción. Retrieved from http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/clase10_MF.pptx.pdf
Parra, D. (2015b). Filtrado Basado en Contenido IIC 3633 -Sistemas Recomendadores. Retrieved from http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/clase6_contentBased_1.pdf
Parra, D. (2015c). Item-Based CF Item-Based CF IIC 3633 -Sistemas Recomendadores. Retrieved from http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/clase2_IBCF.pdf
Parra, D. (2015d). Sistemas Recomendadores Híbridos IIC 3633 -Sistemas Recomendadores. Retrieved from http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/clase8_hybrid.pdf
Peralta Costoya, M. (2013). *Impacto de metadata basada en curriculum en la eficiencia de sistemas de recomendación en educación*. PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE.
Pérez, S. R. (2012). Estudio de técnicas no supervisadas para descubrir tópicos en videos deportivos. Retrieved from http://repositori.uji.es/xmlui/handle/10234/95470
Planells, L. P., & Delegido, J. (2015). Análisis de métodos de validación cruzada para la obtención robusta de parámetros biofísicos. *2015, Vol. 44, P. 55-65*. Retrieved from http://roderic.uv.es/handle/10550/49965
Ponweiser, M. (2012). Latent Dirichlet allocation in R. Retrieved from http://epub.wu.ac.at/3558/
Racine, J. (2012). RStudio: A Platform-Independent IDE for R and Sweave. *Journal of Applied Econometrics*. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/jae.1278/full
Ruiz-Correa, S. (2010). Clasificación de malformaciones craneales causadas por

craneosinostosis primaria utilizando kernels no lineales. *Revista Mexicana de*. Retrieved from http://new.medigraphic.com/cgi-bin/resumen.cgi?IDARTICULO=25521

Rus Marıa Mesas Javega. (2015). *ANALISIS DE TENDENCIAS Y MARCAS DEPORTIVAS A TRAVES DE TWITTER*. Retrieved from https://repositorio.uam.es/bitstream/handle/10486/669057/Mesas_Javega_RusMaria_tfg.pdf?sequence=1

Seiter, J., Amft, O., Rossi, M., & Tröster, G. (2014). Discovery of activity composites using topic models: An analysis of unsupervised methods. *Pervasive and Mobile Computing*. Retrieved from http://www.sciencedirect.com/science/article/pii/S1574119214000832

Sievert, C., & Shirley, K. (2015). LDAvis: Interactive Visualization of Topic Models. Retrieved from https://cran.r-project.org/web/packages/LDAvis/LDAvis.pdf

Tamaral, A. A. (2016). *Manejo de herramientas Big Data para realizar topic modeling en discursos universitarios y clusterización de los resultados*. UNIVERSIDAD AUTONOMA DE MADRID.

Wang, C., & Blei, D. M. (2011). Collaborative Topic Modeling for Recommending Scientific Articles. Retrieved from https://www.cse.cuhk.edu.hk/irwin.king/_media/presentations/presentation.pdf

Wu, H., Yue, K., Pei, Y., Li, B., Zhao, Y., & Dong, F. (2016). Collaborative Topic Regression with social trust ensemble for recommendation in social media systems. *Knowledge-Based Systems*, *97*, 111–122. http://doi.org/10.1016/j.knosys.2016.01.011

Yang, Y., & Huang, S. (2014). Suitability of five cross validation methods for performance evaluation of nonlinear mixed-effects forest models–a case study. *Forestry*. Retrieved from http://forestry.oxfordjournals.org/content/87/5/654.short

Zhang, Z., & Liu, H. (2014). Application and research of improved probability matrix factorization techniques in collaborative filtering. *International Journal of Control and Automation*, *7*(8), 79–92. http://doi.org/10.14257/ijca.2014.7.8.08

ANEXOS

ANEXO 1: Contiene una tabla de una colección de artículos científico, que sirven de base teórica, para el desarrollo y selección de modelo a implementar en el presente trabajo de titulación.

Tabla 16. Trabajos relacionados con técnicas de filtrado colaborativo y basado en contenido en los últimos 5 años.

Artículos científicos de Matrix Factorization o en español Factorización Matricial (en adelante MF)					
Artículo científico	Descripción	Trabajos futuros	Problemas	Hallazgos	Fuente bibliográfica
Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey	Estudia el rol de los diversos modelos de MF para hacer frente a los retos de Filtrado de Colaborativo (en adelante CF).		EL CF hoy en día enfrenta el problema de grandes conjuntos de datos y poca densidad en la matriz de calificación. Este problema se presenta, debido a que no todos los usuarios califican o asignan rankings a los ítems disponibles en los sistemas. Los rankings se realizan en una escalan de 1 a 5.	• La Descomposición en Valores Singulares (SVD) es capaz de manejar grandes conjuntos de datos, escasez en la MF y el problema de escalabilidad del CF eficientemente. • El Análisis de Componentes Principales (PCA) es capaz de encontrar una proyección lineal de los datos de alta dimensión a un sub-espacio de inferior dimensión tales como la varianza; reteniendo lo máximo y minimizando el error de reconstrucción de mínimos cuadrados.	(Bokde, Girase, & Mukhopadhyay, 2014)
Improvement of Matrix Factorization-based Recommender Systems Using Similar User Index	Presenta un nuevo enfoque basado en la MF del índice de usuario similar para los sistemas de recomendación a gran escala.		El problema que se presenta es complejidad de tiempo en la composición de recomendaciones, los enfoques basados en MF son ineficaces en el tratamiento de una gran cantidad de datos históricos.	• El índice de usuario similar se utiliza de manera eficiente para reducir los datos inútiles en el cálculo para la composición de recomendaciones; sin ser afectado por la escasez de datos componiendo recomendaciones de manera más rápida sin tener en cuenta el aumento de tamaño de los datos. Esto mejora el rendimiento de los sistemas de	(Lee & Kwon, 2015)

				recomendación basados en MF.	
Application and Research of Improved Probability Matrix Factorization Techniques in Collaborative Filtering	Proponen una a la técnica MF, mejorando la Probabilidad de la MF llamada MPMF.	El modelo propuesto debe superar algunos retos como: • Desplegarse e Implementar en un sistema. • Se propone método de CF que integre valoraciones de los usuarios y la información del usuario social con el fin de mejorar aún más la precisión recomendación. • Determinar adaptativamente los parámetros óptimos todavía sigue siendo un trabajo de mayor exploración e investigación.	El modelo presenta escasez de datos y escalabilidad, que es omnipresente en los algoritmos de Filtrado Colaborativo en recomendaciones tradicionales y por lo tanto se presenta el MPMF.	• MPMF necesita menos tiempo de ejecución que la de Probabilístico Matriz de Factorización (PMF), pero se puede conseguir una mejor precisión de la predicción. • MPMF, hace frente a un problema de poca densidad de datos con características de programación más sencilla y menor complejidad de tiempo. • La Precisión de la predicción del algoritmo MPMF es bastante alto con una buena escalabilidad en conjuntos de datos reales.	(Zhang & Liu, 2014)
Collaborative Topic Modeling for Recommending Scientific Articles	Este enfoque combina ventajas de Filtrado Colaborativo tradicional y el Probabilistic Topic Models; este último proporciona una estructura latente interpretable para los usuarios y los elementos, que puede formar recomendaciones sobre ambos artículos existentes y recién publicados.	Algoritmo proporciona perfiles de usuario interpretables. Tales perfiles pueden ser útiles en los sistemas de recomendación del mundo real. Las evaluación del modelo se realiza con datos de prueba este enfoque debe ser probado en un ambiente real.	La solución a la MF clásica en las recomendaciones es la combinación con la técnica LDA la cual es Topic Models más simple: pero eficaz para descubrir cómo se crea un documento y clasificarlos en tópicos o temas. De esta manera se caracteriza el contenido de los ítems. Permitiendo realizar recomendaciones sobre los datos de retro alimentación del usuario y el contenido de los ítems.	• Propuesto un algoritmo para recomendar artículos científicos a los usuarios según su contenido y rankings de otros usuarios. • Se demuestra que este enfoque funciona bien con respecto a los métodos tradicionales de MF y hace buenas predicciones sobre los artículos completamente sin calificación. • El enfoque tradicional de recomendación CF, y métodos de recomendación más exitosos son modelos de factores latentes, que proporcionan una mejor recomendación que los métodos de la vecindad clásicos del CF.	(Chong Wang, 2011)

				• Topic Models se han utilizado para tareas como la exploración corpus, la clasificación de documentos, y la recuperación de información. el más simple es LDA. • A diferencia de un modelo de clústeres, donde se asigna a cada ítem a un grupo, LDA permite que los ítems presentan múltiples temas. • La combinación de CF y LDA se ajusta a un modelo que utiliza el espacio latente para explicar tanto las puntuaciones observadas y las palabras observados.	
Recommender systems survey	Este artículo proporciona una visión general de los sistemas de recomendación, así como los métodos y algoritmos de filtrado colaborativo; También explica su evolución, proporciona una clasificación original para estos sistemas, identifica áreas de aplicación en el futuro y desarrolla ciertas áreas seleccionadas para el pasado, presente o futura importancia de los sistemas de recomendación.	• combinación adecuada de métodos de recomendación existentes que utilizan diferentes tipos de información disponible. • Obtener el máximo aprovecho del potencial individual de varios sensores y los dispositivos de la Internet de las cosas. • Adquisición e integración de tendencias relacionadas con hábitos, consumo y gustos de los usuarios individuales en el proceso de recomendación. • Validación de seguridad y privacidad de los procesos de los sistemas de recomendación.		• La primera generación de sistemas de recomendación utilizan datos basados en el contenido de los productos adquiridos o utilizados, datos demográficos recogidos en los registros de los usuarios, y los datos basados en la memoria recogidos de preferencias de elementos de los usuarios. • La segunda generación de sistemas de recomendación, recopilación información social. • La tercera generación de sistemas de recomendación va a utilizar información proporcionada por los	(Bobadilla, Ortega, Hernando, & Gutiérrez, 2013)

		• Nuevas medidas de evaluación y el desarrollo de un estándar para las medidas de evaluación no normalizadas de los sistemas de recomendación. • Diseño de marcos flexibles para análisis automatizado de datos heterogéneos.		dispositivos integrados en Internet. • La primera etapa se centró en mejorar la precisión de recomendación a través del filtrado. La mayoría de los métodos y algoritmos basados en la memoria se han desarrollado y optimizado. En esta etapa, los enfoques híbridos (principalmente de colaboración-demográfica y colaborativo de contenido de filtrado) mejoraron la calidad de las recomendaciones. • En la segunda etapa, los algoritmos que incluían información social con enfoques híbridos fueron adaptadas y desarrolladas. • En la actualidad, los algoritmos de conjunto híbrido incorporan información de ubicación en algoritmos de recomendación existentes.	
Recommending items to group of users using Matrix Factorization based Collaborative Filtering	Cómo llevar a cabo recomendaciones para grupos usando MF basado en CF. Proponen tres enfoques principales para asignar el grupo de usuarios al espacio de factor latente y comparar los métodos propuestos en tres escenarios diferentes: cuando el tamaño del grupo es pequeño, mediano y grande.	• Para realizar recomendaciones a grupos de usuarios (GRS) se debe probar cuál de los enfoques propuestos se adapta a las características del GRS y elegir uno de ellos. • Definir diferentes pesos no solo combinando el número de valoraciones que ha recibido el ítem y la desviación estándar de dichos valores, diferentes definiciones de peso deben ser formuladas.		• Si el conjunto de datos es pequeño y los grupos son pequeños el mejor enfoque Después de Factorización (AF) recomendación es debido a su sencillez y buena precisión. • Si los grupos son de tamaño medio o el conjunto de datos es grande, debemos utilizar el enfoque Ponderado antes de factorizar (FMB). • Si el conjunto de datos es grande o los grupos son muy grandes, Antes de	(Fernando Ortega, Hernando, Bobadilla, & Kang, 2016)

		• Incluir un sistema de entrenamiento para saber cómo los usuarios del grupo representan la estructura social de los grupos reales de los usuarios, en los que algunas personas tienen más influencia en el grupo que otros. • probar si los enfoques propuestos funcionan correctamente con diferentes algoritmos de factorización de la que se usan.		factorización (BF) es el mejor método de recomendación. • MF utilizada en el CF ha presentado mejor calidad de recomendación que la técnica de K-NN basado CF cuando se calcularon las recomendaciones de un único usuario, como también a un grupo de usuarios.	
Latent Dirichlet Allocation (en adelante LDA)					
Latent Dirichlet Allocation for Tag Recommendation	Presentan un enfoque basado en Latent Dirichlet Allocation (LDA) para recomendar etiquetas de los recursos con el fin de mejorar la búsqueda.	• Observar si es beneficioso y efectivo combinar las reglas de asociación y LDA. • Combinación de modelos de lenguaje derivados de las etiquetas reales anotados a un recurso con los modelos de temas latentes. • Experimentar con el uso de la probabilidad de etiquetas derivadas de los modelos de tema para la visualización de las recomendaciones de la etiqueta en forma de nubes de etiquetas.		• En comparación con las reglas de asociación, LDA logra una mayor precisión, y, en particular recomienda etiquetas con mayor relevancia para la búsqueda. • La principal contribución de los modelos de temas latentes es reducir la escasez del espacio etiqueta.	Krestel, R., Fankhauser, P., & Nejdl, W. (2009, October). Latent dirichlet allocation for tag recommendation. In Proceedings of the third ACM conference on Recommender systems (pp. 61-68). ACM.
Managing Uncertainty in Group Recommending Processes	Se estudia el problema de la recomendación a grupos, en el que el objetivo es obtener una recomendación para un grupo de personas. Centrándose en el manejo de la incertidumbre en el proceso de decisión para	Se puede extender fácilmente a aquellas disciplinas donde la agregación de información representa un importante componente, y esas disciplinas incluyen estadística, teoría de la decisión, economía,	Uno de los principales problemas en casos de recomendación a grupos es la búsqueda de mecanismos de agregación que permitan obtener las recomendaciones para el grupo.	• Teniendo en cuenta la incertidumbre a la hora de realizar la agregación, obtenemos mejores resultados para los grupos. • Factores que afectan al rendimiento del sistema, como la forma en que se crea el grupo, el número	De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2009). Managing uncertainty in group recommending processes. User Modeling and User-Adapted Interaction, 19(3), 207-242.

	grupos, estableciendo dos fuentes distintas: la incertidumbre alzar de preferencias del usuario y la incertidumbre que es inherente al proceso de predecir.	ciencias políticas, psicología, etc.		de individuos en el grupo, la función de agregación usada, etc. • El modelo propuesto es bastante genérico, de tal forma que puede ser aplicado en diferentes tareas de recomendación.	
Combining Content-based and Collaborative Recommendations: a Hybrid approach based on Bayesian networks	Se realiza el diseño de un nuevo modelo de recomendación basado en redes bayesianas para intentar realizar predicciones más eficientes y correctas.		Una razón por que la predicción de la calificación no es una ciencia exacta es debido a la incertidumbre intrínseca asociada a las diferentes tareas que componen este campo de investigación.	• Se probó empíricamente que la combinación de la información colaborativa y de contenido ayudan a mejorar la precisión del modelo. • Se presenta las directrices para estimar los valores de probabilidad de un conjunto de datos y se diseñó un algoritmo de propagación eficaz, basado en modelos canónicos. • Se puede trabajar de forma exclusiva aplicando el filtrado colaborativo o basado en contenido.	De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning, 51(7), 785-799.
Using second hand information in Collaborative Recommender Systems	Estudian si la información de segunda mano origina Mejores recomendaciones para ello implementan en dos SR: Uno en redes bayesianas y otro basado en vecindario.		El problema surge cuando, al evaluar un usuario sobre un ítem concreto, las personas con gustos similares no poseen información sobre el ítem. En estos casos el sistema ofrecería una predicción que no sería lo suficientemente buena.	• Se probó que obtener nueva información de segunda mano mejora las predicciones de los sistemas. • El uso de información de segunda mano no contribuye a la predicción del voto: cuando la nueva información no se puede obtener de la base de datos de votos. En esta situación, el uso de información de contenido (si está disponible) puede ser una buena solución para realizar las predicciones, como una aproximación hibrida.	De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Using second-hand information in collaborative recommender systems. Soft Computing, 14(8), 785-798.

				• El uso de información de segunda mano tampoco ayudaría si ya existe suficientes votos de 'primera mano'. • Es provechoso en aquellos casos en los cuales el ítem actual no es raro ni muy frecuente (que debería ser lo usual en la mayoría de sistemas). En estos casos, debería ser interesante la búsqueda de nueva información en la base de datos de votos. • Esta propuesta podría ser útil en tiendas online (como Amazon o una aplicación basada en películas) donde, con bastante frecuencia, aparecen productos nuevos. En estas tiendas, los usuarios empiezan a votar después de incluir el nuevo ítem.	
A Multi-attribute Probabilistic Matrix Factorization Model for Personalized Recommendation	Explotan LDA y Probabilistic Matrix Factorization (en adelante PMF) para incorporar la información del contenido de los artículos y las relaciones sociales de los usuarios, denominado modelo SCT-PMF, para mejorar la exactitud de predicción de recomendación.	Las relaciones sociales pueden incluir la confianza o desconfianza en la información, que son dignos de estudio futuro.	Las relaciones sociales de los usuarios son considerados, pero no hay más acciones en relación con la difusión de información o de propagación entre los usuarios se cuentan.	• El enfoque (SCT-PMF) supera a otros métodos (PMF, LDA y CTR) no importa cómo el parámetro H sintonizado está cambiando. Se ilustra que el modelo propuesto es escalable. • Se observa, el efecto de incluir parámetros sociales dando como resultado, las relaciones sociales tienen una gran influencia en la recomendación de artículos a los usuarios de los experimentos realizados en el mismo. • El contenido de elementos influye fuertemente en el	(Feng Tan, Li Li, Zeyu Zhang, 2014)

				rendimiento de la recomendación en un aspecto de los experimentos realizados.	
Collaborative Topic Regression with social trust ensemble for recommendation in social media systems	Proponen enfoques correspondientes a conocer los factores latentes tanto de los usuarios y los ítems, así como otros parámetros a estimar.	Examinar medidas más avanzadas que califiquen la confianza social entre los usuarios, en el modelo CTR-STE para investigar la eficacia de las recomendaciones. Desarrollar una metodología para encontrar automáticamente el ajuste óptimo de D. Proporcionar al CTR-STE la capacidad de diversificación para mejorar el centro de la diversidad de las recomendaciones.	Los rankings iguales a cero actualmente no se utilizan en los modelos para formar la recomendación, ya que la matriz calificación siempre contiene un gran número de entradas distintas de cero y conduce a altos costos generales de cálculo en la formación de los modelos.	• CTRSTE (extendiendo CTR que incorpora naturalmente la información de contenido a través de LDA) produce consistentemente mejores resultados que la recomendación del modelo CTR. • En comparación con CTRSMF y LACTR, el modelo CTRSTE es simple un principio algorítmico, pero más robusto al trabajar con diferentes conjuntos de datos proporcionando explicaciones más intuitivas para la predicción de características. • La explotación de información de la red social en el modelo CTR puede mejorar significativamente la precisión recomendación, especialmente el indicador de recuperación. • Si los usuarios son más consistentes con sus amigos de confianza en términos de intereses, CTRSTE muestra mejores recomendaciones. • El ensamble de confianza social es más flexible y robusto que otros enfoques para la integración de la red social con el modelo CTR.	

				• Los gustos personales de los usuarios tienen más influencia en sus decisiones de adopción de los ítems que sus círculos sociales en los sistemas de medios sociales.	
Learning to Recommend with User Generated Content	Se propone una forma unificada de utilizar diferentes tipos de UGC para mejorar la precisión de la predicción de las recomendaciones en este trabajo, se centran principalmente en las críticas y las etiquetas. Se estudia la distribución de los intereses de un usuario bajo diferentes temas basados en la agrupación para comprender mejor su preferencia.	CTR tiene dos problemas: primero descripciones de artículos son estáticas y por lo general no distinguen dos productos que están en la misma categoría (Por ejemplo, las descripciones de los ordenadores portátiles que son producidos por Samsung y Lenovo son similares, ya que ambos consisten de palabras similares que se utilizan para describir las características del producto, como la memoria, la CPU y el precio.) Segundo es difícil para inferir la preferencia de un usuario a través de descripciones de los artículos, ya que son independientes para los usuarios Intención de emplear los tweets para deducir intereses de los usuarios en los sitios de redes sociales.	El interés del usuario por lo general se relaciona con algunos temas. Por ejemplo, si Juan es fan de Harry Potter, se le recomendaría no sólo películas relacionadas, sino también libros relacionados, aparatos y ropa. Además, también puede marcar etiquetas y escribir comentarios sobre los temas relacionados. Aunque estos productos están en diferentes categorías, que tienen características comunes que pueden inferirse por UGC. Sin embargo, la mayor parte de las obras existentes justifica con el CF basado en evaluaciones de los usuarios en cada producto por separado.	Entre todos los algoritmos basados en CF, los modelos basados en MF han sido verificados para lograr una precisión satisfactoria en la predicción de calificación, y por lo tanto ampliamente estudiados y desarrollados. Los basados en contenido siguen desempeñando un papel importante en los sistemas de recomendación (etiqueta información social, la opinión del usuario, la respuesta la pregunta, blog, tweet, etc.). En varios sitios de comercio electrónico dominantes, como Amazon, EBay y Jingdong, no hay relaciones sociales para que estos sitios difícilmente pueden beneficiarse de las técnicas de recomendación sociales. Dado que existen opiniones de los usuarios en casi todos los sitios de comercio electrónico y sitios de redes sociales, indica que nuestros modelos tienen una amplia aplicabilidad.	(Li et al., 2014)

Fuente: el autor.

ANEXO 2: código empleado en la extracción de las descripciones desde el sitio web de IMDB.

El código se implementa en el lenguaje de programación Python en conjunto con las librerías bs4 y Lib2.

ANEXO 3: Pre-procesamiento que sufren todos los 10,329 documentos concernientes a descripciones de películas del conjunto de datos de MovieLens.

Tabla 17. Transformación en diferentes etapas del pre-procesamiento del corpus de películas de MovieLens.

Título de la película	**Descripción original**	**Normalización del texto**	**Eliminación de palaras vacías**	**Segmentación**	**Lematizacion**
"Extreme Measures (1996)"	Thriller about Guy Luthan (Hugh Grant), a British doctor working at a hospital in New York who starts making unwanted enquiries when the body of a man who died in his emergency room disappears. The trail leads Luthan to the door of the eminent surgeon Dr Lawrence Myrick (Gene Hackman), but Luthan soon finds himself in danger from people who want the hospital's secret to remain undiscovered.	thriller about guy luthan (hugh grant), a british doctor working at a hospital in new york who starts making unwanted enquiries when the body of a man who died in his emergency room disappears. the trail leads luthan to the door of the eminent surgeon dr lawrence myrick (gene hackman), but luthan soon finds himself in danger from people who want the hospital's secret to remain undiscovered.	thriller guy luthan (hugh grant), british doctor working hospital york starts making unwanted enquiries body man died emergency room disappears. trail leads luthan door eminent surgeon dr lawrence myrick (gene hackman), luthan finds danger people hospital' secret remain undiscovered.	thriller guy luthan (hugh grant), british doctor working hospital york starts making unwanted enquiries body man died emergency room disappears. trail leads luthan door eminent surgeon dr lawrence myrick (gene hackman), luthan finds danger people hospital' secret remain undiscovered.	thriller guy luthan hugh grant british doctor work hospit york start make unwant enquiri bodi man die emerg room disappear trail lead luthan door emin surgeon dr lawrenc myrick gene hackman luthan find danger peopl hospit secret remain undiscov

Fuente: el autor.

ANEXO 4: presenta una imagen con una tabla de frecuencia de los términos con una presencia superior a 750 dentro del conjunto de datos de MovieLens.

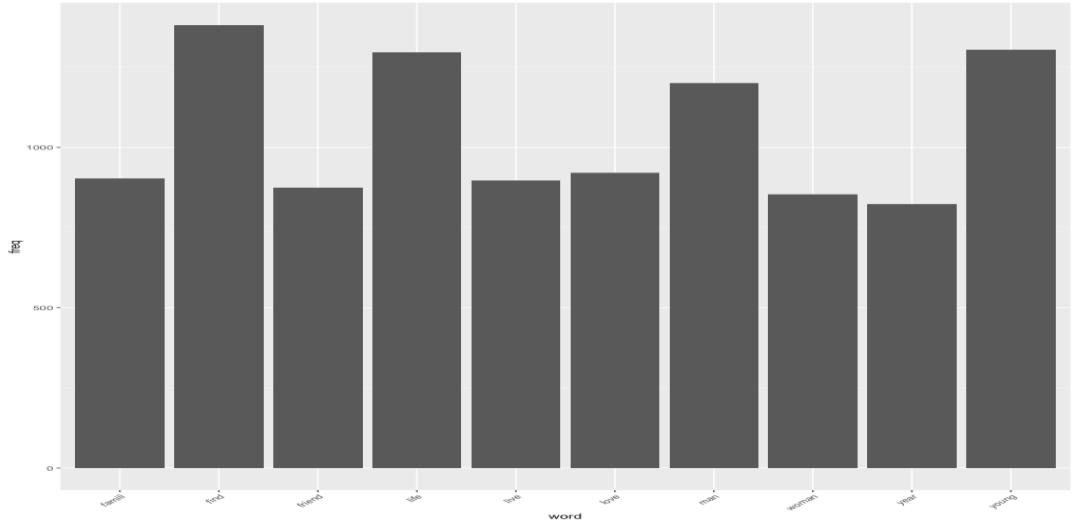


Figura 57. Tabla de frecuencia con términos superior a 750 del conjunto de datos MovieLens.
Fuente: el autor.

ANEXO 5: contiene una tabla comparativa entre las descripciones de las películas y la review disponible en el sitio web de IMDB.

Tabla 18. Comparación entre descripción y review disponible en el sitio web IMDB del conjunto de datos MovieLens.

Identificador de película MovieLens	Descripción disponible en IMDB	Review disponible en IMDB
1	A cowboy doll is profoundly threatened and jealous when a new spaceman figure supplants him as top toy in a boy's room.	A very sweet and fun movie. TOY STORY has great computer animation. A simple yet well told story is also added as well. The voice overs are top notch and everyone gets a fair share in this movie. PIXAR has done a good job placing itself on the map.Set in Andy's room and before the family moves, his mother decides to throw a birthday party for the young lad. Andy's toys all have a conscience and are led by his favorite, Woody. But alas, the big new toy from the party unknowingly takes Woody's mantle.Envy and jealousy are brought up well. So is the ignorance of Andy's toys and the loss of innocence from Buzz Lightyear. TOY STORY is well packaged exceeded only by its sequel. A good rent.
2	When two kids find and play a magical board game, they release a man trapped for decades in it and a host of dangers that can only be stopped by finishing the game.	Among the thousands of films I have viewed, this movie would rank near the top for sheer entertainment. That's not saying it's the best-made or most intelligent or scariest or funniest or features the best effects, etc. etc. But combine all those and you have a film that's tough to beat when you're looking for 100 minutes of escapist fun.The film features some wild computer-enhanced special effects that were new to its day, but now about 10 years later, it's no big deal. In fact, some of it, such as the lion, look pretty hokey compared to the stuff that's out there now. To me, it was story that was the lure, anyway, not the special effects.Because it's so much fun, this is one of the fastest-moving films I've ever viewed. The time flies by. It's not to be analyzed or given much thought, because it's so ludicrous. You just go along for the wild ride in this fantasy-adventure and get a bunch of laughs and thrills along the way. That's one of the big attractions of this of film: the excellent combination of adventure and comedy.Are there annoying things in this movie? Sure. To me, it was Bonnie Hunt's occult beliefs and too many OMGs and the overdone character of the hunter (Jonathan Hyde). Other than that, I loved the film the first I saw it and every time afterward. I've probably viewed this movie as much as any, simply because it was so entertaining.Robin Williams, David Alan Grier and the two kids, Kristen Dunst

		and Bradley Pierce, were all great people to watch and share this adventure with. Now THIS is entertainment!
3	John and Max resolve to save their beloved bait shop from turning into an Italian restaurant, just as its new female owner catches Max's attention.	After the success of "Grumpy Old Men" a sequel was inevitable. The fact that it's also inferior to the hilarious first movie was also inevitable. Plot In A Paragraph: John and Max are still on good terms following the events at the end of last movie. They still greet each other in the same way, but now with affection. They join forces to save their beloveds bait shop from being turned into an Italian restaurant by its new owner (Sophia Loren) who has just took over.Almost all of the cast from the first movie return, Walter Matthau and Jack Lemmon are their usual brilliant self, with their usual awesome charisma. Ann Margaret isn't given much to do this time around. Sophia Loren is certainly still a sex bomb. Burgess Meredith is back, and is as smutty if not smithies than the first time around. Daryl Hannah and Kevin Pollack both return too, and are both as reliable as you'd expect.Sadly, Grumpier Old Men was Burgess Meredith's final film. He looks in bad shape here, but still puts in a good fun, full of life performance.

Fuente: el autor.

ANEXO 6: presenta la Figura 29 que contiene una distribución de frecuencia de las palabras con una frecuencia superior a 1500 veces dentro del conjunto de datos MovieLens.

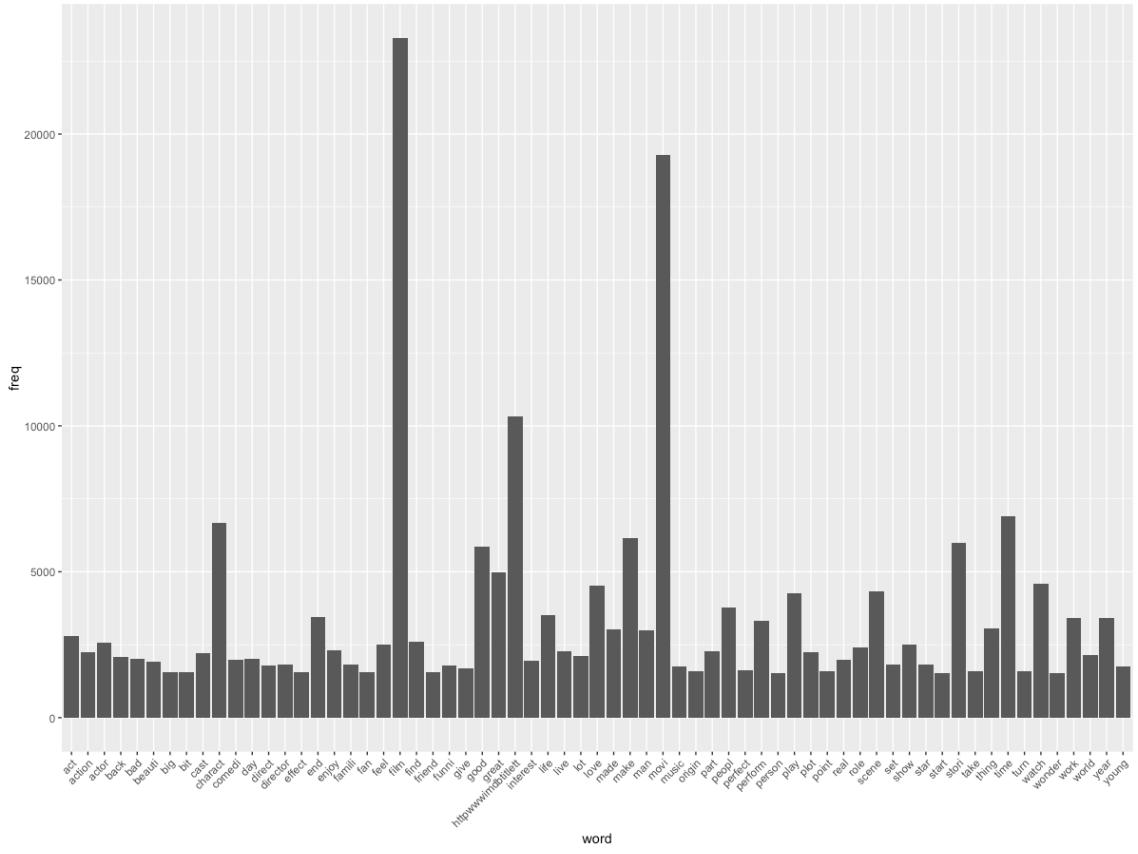


Figura 58. Frecuencia de palabras superior a 1500 de las Review del conjunto de datos MovieLens

Fuente: el autor.

ANEXO 7: Presenta un ejemplo con la review de la película Toy Story del conjunto de datos MovieLens con la finalidad de mostrar que sigue el mismo pre-proceso que con las descripciones de MovieLens.

Tabla 19.Pre-procesamiento de las reviews del conjunto de datos MovieLens.

Descripción original	Normalización del texto	Eliminación de palaras vacías	Segmentación	Lematización
http://www.imdb.com/title/tt0114709 Toy Story (1995) A very sweet and fun movie. TOY STORY has great computer animation. A simple yet well told story is also added as well. The voice overs are top notch and everyone gets a fair share in this movie. PIXAR has done a good job placing itself on the map.Set in Andy's room and before the family moves, his mother decides to throw a birthday party for the young lad. Andy's toys all have a conscience and are led by his favorite, Woody. But alas, the big new toy from the party unknowingly takes Woody's mantle.Envy and jealousy are brought up well. So is the ignorance of Andy's toys and the loss of innocence from Buzz Lightyear. TOY STORY is well packaged exceeded only by its sequel. A good rent.	http://www.imdb.com/title/tt0114709 toy story (1995) a very sweet and fun movie. toy story has great computer animation. a simple yet well told story is also added as well. the voice overs are top notch and everyone gets a fair share in this movie. pixar has done a good job placing itself on the map.set in andy's room and before the family moves, his mother decides to throw a birthday party for the young lad. andy's toys all have a conscience and are led by his favorite, woody. but alas, the big new toy from the party unknowingly takes woody's mantle.envy and jealousy are brought up well. so is the ignorance of andy's toys and the loss of innocence from buzz lightyear. toy story is well packaged exceeded only by its sequel. a good rent.	http://www.imdb./title/tt0114709 toy story (1995) sweet fun movie. toy story great computer animation. simple told story added . voice overs top notch fair share movie. pixar good job placing map.set andy' room family moves, mother decides throw birthday party young lad. andy' toys conscience led favorite, woody. alas, big toy party unknowingly takes woody' mantle.envy jealousy brought . ignorance andy' toys loss innocence buzz lightyear. toy story packaged exceeded sequel. good rent.	httpwwwimdbtitlett toy story sweet fun movie toy story great computer animation simple told story added voice overs top notch fair share movie pixar good job placing mapset andy room family moves mother decides throw birthday party young lad andy toys conscience led favorite woody alas big toy party unknowingly takes woody mantleenvy jealousy brought ignorance andy toys loss innocence buzz lightyear toy story packaged exceeded sequel good rent	toy sweet fun toy great comput anim simpl told ad voic top notch fair share pixar good job place mapset andi room famili move mother decid throw birthday parti young lad andi toy conscienc led favorit woodi ala big toy parti unknow woodi mantleenvi jealousi brought ignor andi toy loss innoc buzz lightyear toy packag exceed sequel good rent

Fuente: el autor.