UNIVERSIDAD INTERNACIONAL DE LA RIOJA unir

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Máster universitario en Dirección e Ingeniería de Sitios Web

RAMA PROFESIONAL

[Sistema no supervisado para la recomendación de contenidos educativos basado en un sistema híbrido.]

Trabajo Fin de Máster

presentado por: Sánchez Molano, Boris

Directora: Martínez Muñoz, Miriam

Ciudad: Cali, Colombia

Fecha: junio 11 de 2019

Contenido

Índice de ilustraciones

Estructura de Capítulos

Capítulo 1: Compuesto por la *Introducción* del Trabajo Final del Máster (TFM). Se explica el trabajo realizado, también se presenta la *Justificación* del tema que se va a exponer, el *Planteamiento del Problema* y la *Estructura del documento*.

Capítulo 2: En esta sección se encuentra el *Contexto* donde se describen la problemática y las condiciones del sitio web en donde será implementada la metodología propuesta. Así mismo, se expone el *Estado del Arte* en donde se presentan las diferentes metodologías utilizadas para el desarrollo de los Sistemas de Recomendación, se exponen las técnicas más comunes, se presentan las ventajas y desventajas de cada una.

Capítulo 3: Se definen los *Objetivos generales* y *Específicos* del TFM; se explora las metodologías más apropiadas para las necesidades específicas del sitio web en el que será implementada (Sistemas basado en Contenidos, con poco seguimiento de la actividad de usuario y no estructurada temáticamente).

Capítulo 4: En esta sección se presenta la *propuesta metodológica* específica, se describe los alcances, conexiones, y soluciones propuestas. Básicamente se establece la solución planteada

Capítulo 5: Se plantea los aspectos necesarios para la *implementación de la metodología* propuesta para el TFM, requerimientos, estructuras y prototipos de publicación.

Capítulo 6: Se describen las *conclusiones*, estas asociadas a los objetivos buscados. Se proponen *Recomendaciones* y Temas para *trabajos futuros o explorar*

Capítulo 7: Se presenta la *Bibliografía* de los recursos utilizados en el TFM. También se presentan Anexos las cuales muestran tablas, diagramas o imágenes complementarios

Introducción

Una de las grandes dificultades que presentan los sitios web con alta densidad de información y contenidos es la incapacidad de ofrecerle a los usuarios recomendaciones acertadas y pertinentes sobre contenidos relacionados, que pueden ser encontrados en un sitio web; los altos índices de rebotes o el poco tiempo de permanencia en el sitio web se presume como la incapacidad de retener a los usuarios, o no atender a las necesidades e intereses que les hicieron a ellos ingresar a una página web particular.

Los sistemas de recomendaciones se han convertido en los últimos años en elementos claves para poder entender y atender a los usuarios, identificando las preferencias, opiniones o tendencias (usualmente no manifiestas). Estos sistemas son muy útiles para evaluar, filtrar, sugerir y presentar información disponible en el sitio web particular, usando múltiples métodos y técnicas para ello. Estos sistemas de recomendación se han implementado exitosamente en sitios de comercio virtual, de servicios (viajes, hoteleros, streaming) y redes sociales. La mayoría de grandes sitios están realizando implementaciones en este sentido, Facebook, Amazon, Netflix, etc. Cada uno con diferentes enfoques y técnicas. Sin embargo, hay una poca implementación en sitios de contenidos (Blogs, periódicos, magazines, etc.) en los cuales es más complejo hacer un seguimiento de las interacciones de los usuarios, y presentan contenidos con carácter menos homogéneos o no estructurados.

Para este caso particular se utilizará un sistema híbrido el cual hace uso del filtrado basado en conocimiento en cuanto a la relación usuario-documento y del filtrado basado en contenidos. En el filtrado basado en conocimiento, se hace análisis de los intereses del usuario o de la estructura del contenido a partir de alguna forma particular de conocimiento del dominio de los ítems o del documento. Y en el enfoque basado en contenidos en el cual se analiza contenido, sus características y relaciones con otros contenidos dentro del mismo sistema

La intención de este TFM es estructurar una propuesta metodológica para la elaboración de un sistema de recomendaciones hibrido en donde la obtención de los datos, perfiles de los usuarios y su interacción con el contenido en un sitio educativo se convierten en aspectos claves.

El principal problema está relacionado con la característica del sitio; un sitio de contenidos típico es usualmente es más heterogéneo y tiene menos control sobre los usuarios y sus interacciones; no maneja una tipología definida de las necesidades, gustos y preferencias de los usuarios, lo cual no permite clasificar específicamente las características del producto o servicio que ofrecen; e impiden hacer mejores predicciones de los intereses de los usuarios.

Se busca que se pueda implementar un sistema no supervisado para el análisis de los contenidos, de la relación de los contenidos con los usuarios y otros aspectos que permitan predecir y recomendar contenidos pertinentes.

Una de las ventajas es que un modelo de este tipo puede ser fácilmente adaptado e implementado en cualquier sitio de contenidos con ciertas características, sitios con alta tasa de entrada vía búsqueda, altos porcentaje de rebote y poco tiempo de permanencia en el sitio. La metodología buscaría estrategias para indexar, catalogar, filtrar e identificar contenidos pertinentes para el usuario.

Contexto

Sistema de Recomendación

Los Sistemas de Recomendación (SR) nacieron como una estrategia que permite ayudar al usuario a encontrar la información más pertinente a sus inquietudes y necesidades. El reto para lidiar con el problema de la sobrecarga de información es el que ha permitido a los SR cobrar importancia día a día. "De ahí que, los sistemas de recomendación se convierten en una alternativa para ayudar a los usuarios a obtener información precisa y personalizada de grandes repositorios de información que pueden estar disponibles en Internet."(Valdiviezo-Diaz & Hernando, 2016)

Un sistema recomendador es definido como: "un sistema automático que le da a un usuario u y un conjunto de ítems k algunos de ellos desconocidos para u, desplegando un subconjunto de ítems S personalizados que pueden ser interesantes para u" (Vera-del-Campo, 2012)

Usualmente los SR son ampliamente utilizados en la Web para recomendar productos y servicios. Su uso es extendido en los sitios más populares de internet, en redes sociales como Facebook y Twitter donde los sistemas de recomendación se enfocan en la sugerencia de personas a seguir o amigos; en sitios de repositorios de video o audio tipo Spotify, YouTube, Netflix en los cuales recomiendan títulos dependiendo de las preferencias y visualizaciones de los usuarios; y en sitios de comercio electrónico tipo Amazon, eBay en los que dependiendo de consultas, análisis de compras y otros datos, recomiendan productos bien delimitados a los gustos del usuario.

Una de las principales tareas de un SR es la de ayudar en el proceso de búsqueda de ítems (personas, productos, información) no antes determinados o conocidos. Un SR debe poder filtrar de amplio volumen de información adaptándose a unos requerimientos iniciales de un usuario.

"Un sistema de recomendación es un conjunto de técnicas de recuperación de información que intenta descubrir el interés de los usuarios por determinados objetos, con la finalidad de ofrecerles un conjunto de objetos afines, relacionados a su perfil, en los que podría estar interesado." (Núñez, 2012)

Las funcionalidades de un SR normalmente es la de recomendar ítems de manera ordenada, predecir las valoraciones de un ítem y hacer recomendaciones a partir de un conjunto de información determinado (ya sea contextual o estructurado).

Estructura de los sistemas de recomendación

Según Herrera-Viedma, Porcel, & Hidalgo, (2004) los SR tienen los siguientes elementos fundamentales que pueden ser usados como criterios de clasificación:

- Entradas/salidas del proceso de generación de recomendaciones
- El método usado para generar las recomendaciones

Entradas / Salidas

Dependiendo del tipo de modelo y las técnicas de procesamiento utilizadas, los SR usan la información suministrada del usuario activo (perfil, preferencias, rangos etnográficos, educación, intereses previos, etc.), pero también usa información sobre los ítems, o como las personas han interactuado con esos ítems. Todos estos elementos son utilizados para poder retroalimentar al usuario de manera pertinente

Para poder realizar una recomendación se requiere entender en dónde está situado el usuario, entender el contexto y entender cómo se pueden clasificar esos ítems. La información del usuario puede obtenerse de diferentes formas:

- de carácter ***implícito*** a través de seguimientos de navegación o información recabada de su actividad o evaluaciones sin que el usuario sea consciente de ello cuando se obtienen datos de entorno (tecnologías, entornos móviles, velocidad de conexión, etc.).
- Otra manera de recopilar información es cuando al usuario se le solicita interactuar con los ítems de manera ***explícita***. Se recopilan datos en actividades como la evaluación o rating de un ítem, cuando se le solicita la evaluación y opiniones calificadas. Al momento de definir sus perfiles en donde se contextualiza demográficamente al usuario (género, nacionalidad, educación, etc)
- Otro tipo de datos son los obtenidos a través de ***análisis textual*** de los ítems.

Las salidas es el resultado del proceso, y las constituyen el subconjunto de ítems recomendados. Estas pueden desplegarse de múltiples formas, como sugerencias o listas de ítems; presentarse en forma de recomendaciones basadas en preferencias particulares o grupales (ej. el índice de coincidencia en Netflix o las sugerencias de amigos en redes como Facebook y con un 'podrías conocer').

Hay múltiples formas de representar las recomendaciones se sugiere ser transparente en su funcionamiento y en lo posible presentar los criterios que permitieron la recomendación. De acuerdo con Martín (2016), las salidas del sistema podrían conformarse en:

- **Una recomendación**, lista ordenada formada por los k ítems que se pueden ser más interesantes para el usuario. En ciertas situaciones estas recomendaciones pueden integrar restricciones previas determinadas por el usuario, realizando un filtro previo del conjunto de datos original.
- **Una predicción,** el SR deduce cual puede ser la valoración resultante y se convierte en una opinión anticipada de un usuario sobre un ítem, intenta determinar el grado de satisfacción de un usuario con un ítem determinado. se conoce como Individual Scoring.

Métodos de generación de los sistemas de recomendación

Los SR utilizan diversas técnicas determinadas en la forma como obtienen datos de los ítems, de la interacción con uno o varios usuarios y de la interacción de los usuarios con los ítems. A partir de estos elementos comúnmente se han determinado tres clasificaciones (Caro Martínez, 2017)

- Los SR basados en **filtrado colaborativo**. Se basan en las valoraciones y opiniones los usuarios han realizado para sobre el conjunto de ítems
- Los SR basados en **contenido**. En donde la descripción del ítem o el contenido de este determina las recomendaciones
- Los sistemas de recomendación **híbridos**. Mezclan diversas técnicas.

SR basado en Filtrado Colaborativo

Esta técnica de filtrado está determinada por las recomendaciones y valoraciones previas realizadas por usuarios. Esas valoraciones pueden obtenerse de distintas maneras:

evaluación cuantitativa, me gusta / no me gusta, comentarios sobre el ítem, etc. Estos sistemas basan sus predicciones y recomendaciones basándose en las opiniones de usuarios con características similares al que se le va hacer la recomendación, y conforma conjuntos de usuarios afines a un mismo ítem.

Entre las ventajas de este modelo se puede destacar que:

- Permite recomendar ítems difíciles de analizar, sin mucho contenido o de carácter más gráfico o menos delimitado semánticamente
- Recomienda ítems basados en las preferencias del usuario, ítems que claramente se conforman en intereses particulares de cada usuario, que fueron encontrados o seleccionados respecto alguna subjetividad
- Las recomendaciones son válidas y no necesariamente esperadas o con un mismo carácter o similitud evidente
- Puede usarse en cualquier tipo de ítem o en cualquier formato (música, películas, libros, etc.)

Respecto a las desventajas de esta técnica se encuentran:

- El problema con usuarios nuevos o con perfiles incompletos, estos no permiten encontrar vecinos cercanos ya que no se tiene ninguna información sobre las preferencias de este tipo de usuario
- Mucha dispersión cuando los ítems están poco valorados sucede cuando hay muchos usuarios y muchos ítems, pero pocas valoraciones; no se genera proximidad y las predicciones son muy débiles por la insuficiencia de conexiones
- Usuarios con gustos muy particulares los cuales pueden hacer imprecisas las predicciones, se les denomina oveja negra o *green sheep*.
- Cómo las recomendaciones no necesariamente se basan en similitudes del ítem, los usuarios pueden pensar que las recomendaciones no son confiables
- El análisis de ítems nuevos o que nunca han sido valorados y que no pueden ser clasificados, por lo cual no se pueden realizar cálculos de similitud.

Las técnicas de Filtrado colaborativo usan varias estrategias, basadas en memoria y basadas en modelos

Estrategias Basadas en Memoria

Estos sistemas analizan las valoraciones realizadas por los usuarios para establecer grupos de usuarios con preferencias similares, conocidos como “grupos de vecinos” (método KNN, K-Nearest-Neighbour, por sus siglas en inglés).

Usan “algoritmos que trabajan sobre las filas de la matriz de ratings R, es decir, operan con los ratings hechos por el usuario actual, u_a, sobre los distintos ítems. La formación del grupo de vecinos se implementa en dos pasos: primero se calcula la *similitud* entre todos los usuarios de la matriz R mediante métricas de proximidad y después se genera la vecindad del usuario activo para seleccionar aquellos usuarios más próximos a él.” (Martín, 2016, p. 19)

Estrategias Basadas en Modelos

Para Caro-Martínez, (2017) el principal inconveniente de las estrategias basadas en vecinos es que la complejidad de cálculo es enorme. Debido a esto se necesita implementar estrategias basadas en modelos. Por lo cual es necesario construir modelos estadísticos de patrones de valoración de usuarios y productos, para hacer predicciones automáticas de valoraciones. Se realizan las recomendaciones para el usuario activo teniendo en cuenta sus vecinos más cercanos y sus correspondientes valoraciones.

Los algoritmos de esta categoría incluyen enfoques como matrices de factorización, basados en grafos, redes neuronales o probabilísticos, el más destacado el modelo de factores latentes.

“En esta técnica, se observa al conjunto de ítems, denotado por Iu_a, que el usuario u_a ha valorado y se calcula la *similitud* al ítem objetivo i_j para seleccionar los k ítems más parecidos (i_1, i_2, ..., i_k) según sus *similaridades* correspondientes (Si_1, Si_2, ..., Si_k). Las predicciones pueden calcularse gracias a una media con pesos de los ratings del usuario activo sobre esos ítems similares.” (Martín, 2016. p. 19)

SR basado en Contenidos

Reconocido como Content-based filtering o recomendación ítem-ítem, esta técnica no usa patrones de valoración realizados por usuarios del sistema, se enfocan en el contenido

específicamente, ya sea la descripción de los ítems, el contenido del recurso y la información del perfil de los usuarios para generar la recomendación. El ítem está representado por palabras importantes que lo conforman y se buscan relaciones y coincidencias entre esa representación con el conjunto de ítems existentes y el perfil del usuario.

"Teniendo todo esto en cuenta, es fácil concluir que los sistemas de recomendación basados en contenido son muy usados en contextos donde hay una gran cantidad de información disponible sobre los productos. Son muy adecuados en contextos de gran riqueza textual y dominios no estructurados, como por ejemplo las páginas web." (Caro, 2017).

El contenido de los ítems es el que permite predecir la relevancia de este, en ocasiones su usan análisis de visitas pasadas del usuario, preferencias, referencias perfiles, etc.

Este tipo de técnica no hace un seguimiento de los usuarios, solo usa la información disponible, por tanto, su índice de personalización es reducido.

"La ventaja de este filtrado es que solo necesita de información interna, es decir, sobre los ítems del catálogo, sin necesidad de personalizarlo al usuario que interactúa, lo que puede ser interesante para casos en que no haya historial del usuario a evaluar, o éste sea insuficiente. También puede ser ventajoso cuando no hay muchos usuarios registrados, lo que perjudica la formación de perfiles colaborativos. La desventaja es que la propuesta de valor es idéntica para todos los usuarios, perdiendo la posibilidad de personalización, además de que los ítems deben estar valorados por los usuarios." (Enio, 2017).

Según Martín (2016) un SR basado en contenidos tiene 3 componentes principales:

- **Analizador de contenidos**, el cual se encarga de pre-procesar los items/documentos y permite proceder en las siguientes secuencias del sistema. Tiene que extraer la información de los textos y generar una representación de los términos fundamentales
- **Gestor de perfiles**, recoge datos de necesarios para perfilar al usuario, normalmente se emplean técnicas de aprendizaje como estrategia de generalización. "Para construir del perfil del usuario actual u_a, se define un conjunto de entrenamiento TR_a para u_a que consiste en un conjunto de pares (I_k, r_k) donde r_k

es el rating del ítem l_k por u_a. A partir de ese conjunto se genera un modelo predictivo que será el perfil del usuario y se almacenará en el repositorio Profiles." (Martín, 2016, p. 23).

- **Componente de Filtrado**, en este módulo se seleccionan los ítems más pertinentes que sean similares a los ítems valorados en el perfil. Las reacciones del usuario ante los ítems recomendados deben ser almacenadas como *feedback*, estas reacciones pueden ser explicitas (por medio de evaluaciones, me gusta u otro método) o implícitas, inferidas de la navegación y uso de los ítems recomendados.

Según (Martin 2016) y (Núñez Valdéz, 2012) las ventajas de esta técnica respecto a las de filtrado colaborativo es que:

- No necesita evaluaciones previas de unos ítems para que tenga un valor predictivo (evita el *cold start* o arranque en frio, inconveniente cuando un ítem es nuevo o no tiene valoración previa).
- Personaliza los intereses, las recomendaciones dependen de la similitud de intereses del perfil de usuario con el de los ítems
- Es más trasparente el proceso de selección de un ítem para recomendarlo, ya que puede explicarse el '*porqué se recomienda'* (ej. El sistema de recomendación de Netflix en la categoría '*Ya que viste X película te recomendamos:*')
- Menos intrusivo porque no exige que el usuario este constantemente evaluando cada recomendación de los ítems para poder hacer nuevas y más pertinentes predicciones.

Respecto a las desventajas plantean (Martin 2016) y (Núñez, 2012) que :

- Hay un análisis limitado, existe un límite natural de características que pueden ser evaluadas y asociadas
- Sobre-especialización, ya que solo se recomienda según lo visitado y/o valorado puede que ítems que potencialmente sean interesantes para ese usuario no sean expuestos, debido a que este tipo de técnicas muestran ítems similares y posiblemente repetidos, (requiere implementar un índice de aleatoriedad)
- Subjetividad de contenidos, hay diferentes tipos de contenidos que son complejos de analizar, aquellos como audios, imágenes, gráficos, videos, elementos no textuales, etc.)
- Requiere costo en tiempo, esfuerzo de aprendizaje y filtrado

SR Híbridos

Estos sistemas combinan múltiples técnicas de recomendaciones con la intensión de tratar de eliminar o solventar las fallas particulares de cada técnica. La premisa es que la combinación de técnicas puede proporcionar recomendaciones y predicciones más precisas y efectivas, para ello utilizan implementaciones separadas de algoritmos o combinación de resultados obtenidos de diversas estrategias

“Para crear un sistema híbrido colaborativo basado en contenido, los perfiles de usuario se mantienen según el análisis de los contenidos de los ítems, y directamente se comparan esos perfiles para determinar las *similitud* entre usuarios para una recomendación colaborativa” (Núñez, 2012, p.29)

Según Caro (2017), existen tres formas de diseñar sistemas de recomendación híbridos:

Diseño en Conjunto.

Para este caso se puede aplicar diversos algoritmos de recomendación (el resultado de un filtrado colaborativo y luego un recomendador basado en contenidos) para generar una predicción más acertada. Este tipo de diseño puede tener dos tipos

- **Diseño en paralelo**. El cual combina todos los resultados obtenidos como entrada de datos para obtener una única salida o predicción. (Ilustración 1)
- **Diseño Secuencial**. Este sistema hace un análisis recursivo de cada entrada y su sucesiva respuesta, esta última respuesta se convierte en la entrada del siguiente paso, y así sucesivamente hasta combinar todas las respuestas en una única respuesta final (Ilustración 2)

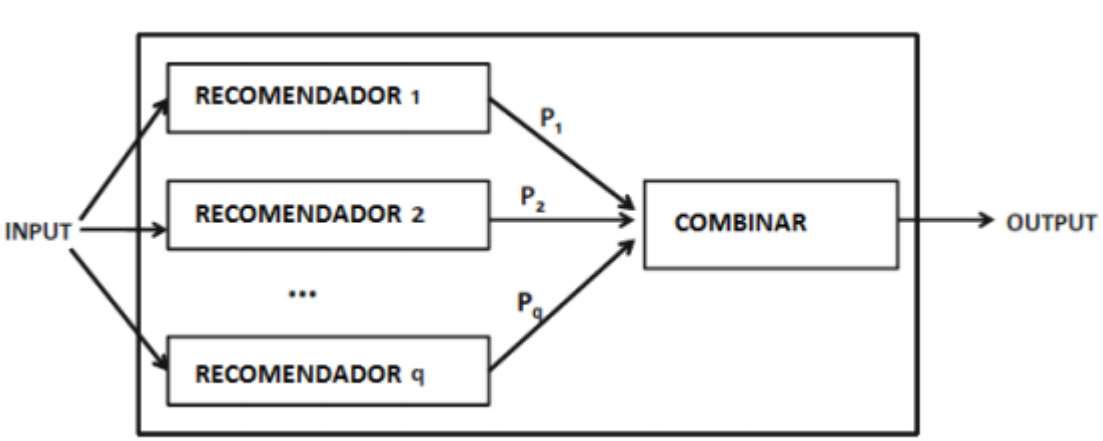


Ilustración 1. Diseño paralelo de un recomendador hibrido. Fuente: (Caro M, 2017)

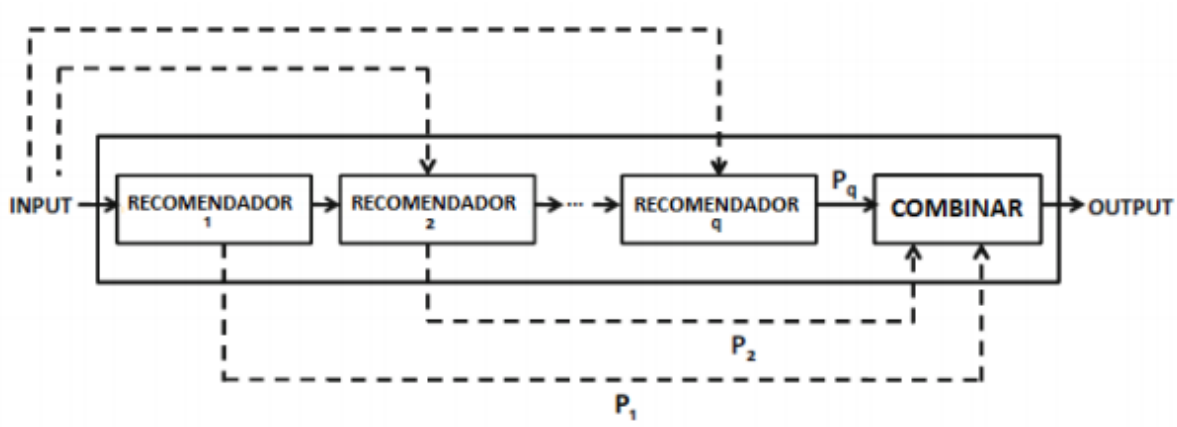


Ilustración 2. Diseño secuencial de un recomendador hibrido. Fuente: (Caro, M. 2017)

Diseño Monolítico

"Se utiliza un único algoritmo de recomendación sobre varios inputs para generar un único output." (Caro, 2017)

Sistemas Mezclados

"Se utilizan diferentes algoritmos de recomendación sobre un único input, como en el caso del diseño en conjunto, pero se obtienen varios outputs. "(Caro, 2017).

Según Martín (2016), Caro (2017) y Colombo (2017); existen 7 técnicas para combinar estas estrategias de recomendación

1. **Ponderación (*weighted*).** La hibridación por ponderación se realiza cuando la puntuación de un ítem recomendado se calculó a partir de utilizar todas las técnicas disponibles en el SR y combinarlas para obtener una única recomendación, la cual estará dada por el ítem con mayor puntuación. Cada recomendador puede tener diferente peso en la calificación. La ventaja es que se puede realizar ajustes al proceso, y la desventaja es que implica realizar el proceso para cada técnica de ponderación.
2. **Conmutación (*switching*).** El sistema puede utilizar (intercambiar) una técnica a otra dependiendo de la necesidad particular. Dependiendo de ciertas condiciones el sistema aplica un recomendador, y si no utiliza otros métodos. La ventaja es que se

le puede determinar al SR la capacidad de entender que sistema funciona más eficientemente (mayor cantidad de respuestas bien ponderadas), y la desventaja es que implica una complejidad extra para seleccionar los criterios de conmutación.

3. **Mezcla (*mixed*).** En este método se despliegan a la vez distintas recomendaciones resultantes de la aplicación de diferentes sistemas de recomendación usadas. Es la predicción en simultaneo en una única lista de ítems, los cuales mostrarán solo los top-n de ellos; esto requiere un criterio de ordenamiento lo cual puede ser una desventaja. La ventaja del método de mezcla es que puede enfrentarse al problema de arranque en frio y descubrir nichos de usuarios.
4. **Combinación de características (*feature combination*).** En un único algoritmo de recomendación, se integran el conjunto de datos resultantes de la utilización de todos los métodos disponibles en el SR, esto con el fin de obtener un *dataset* aumentado.
5. **Cascada (*cascade*).** Un sistema de recomendación escalonado que refina sucesivamente el resultado (conjunto) de recomendaciones dados por otro sistema de recomendaciones, los resultados se convierten en datos de entrada para ser refinadas posteriormente hasta obtener un rating lo suficientemente alto que permita ser recomendado al usuario.
6. **Aumento de características (*feature augmentation*)** Se usa la salida de una técnica como entrada de otra técnica, esta salida es una recomendación que se convierte en una característica adicional que va ser integrada como parte de los datos de entrada de las otras técnicas. Es decir, no solo se publica los *top-n* de cada técnica, sino que esta se convierte en un insumo dentro del sistema (un aumento en los datos de entrada) permitiendo que los resultados sean refinados múltiples veces.
7. **Meta-nivel (*meta-level*).** El modelo aprendido por un sistema recomendador es usado como entrada de otro sistema. Aquí no se usa los ratings como entrada sino el modelo completo, usualmente es la combinación de filtrado colaborativo y recomendaciones basado en contenidos

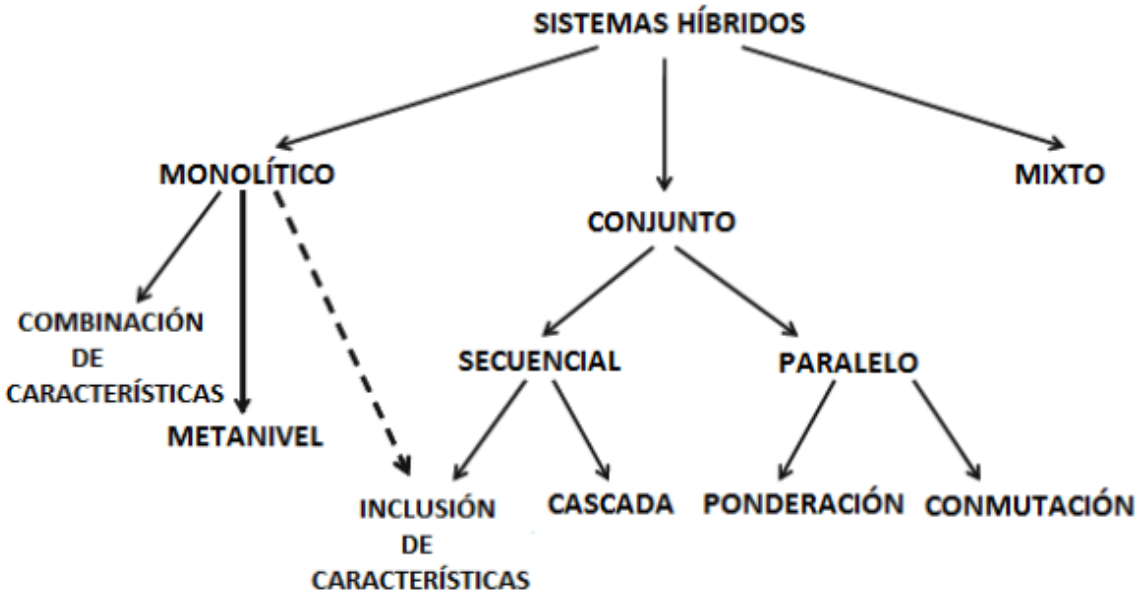


Ilustración 3. Clasificación de las técnicas para combinar estrategias de recomendación en los tres tipos de diseños híbridos. Fuente: (Caro 2017)

Modelo de tópicos probabilísticos

Las diferentes técnicas de los sistemas recomendadores requieren como aspecto fundamental entender los ítems, los usuarios y las relaciones usuario-ítems para permitirse generar predicciones y desplegar un conjunto de ítems que sean particularmente interesantes para el usuario.

Cada método tiene ventajas y desventajas que tratan de ser solventadas a través de diversas estrategias y métodos probabilísticos relacionados con el modelado de factores latentes, para modelar las preferencias del usuario, matrices de factorización y modelado de tópicos.

> "Los modelos de tópicos son un conjunto de algoritmos que proporcionan una solución estadística al problema de la gestión de grandes archivos de documentos. Con los recientes avances científicos en apoyo de los componentes de la máquina de aprendizaje flexibles sin supervisión para el modelado, algoritmos escalables para la inferencia posterior y con el mayor acceso a modelos de conjuntos de datos masivos (dataset), los modelos de tópicos prometen ser un componente importante para la síntesis y la comprensión de nuestros crecientes archivos digitalizados de la información." (Blei, 2012b)

El análisis de datos requiere reducir la dimensionalidad y obtener representaciones efectivas e interpretables, en este escenario en donde existe una gran cantidad de datos el modelado probabilístico de tópicos es fundamental para la extracción semántica con un carácter no supervisado. "Este tipo de modelos se conciben como mecanismos para abstraer del discurso real una representación más compacta que capture el contenido de los documentos analizados" (Acosta, Aguilar, & Araya, 2018).

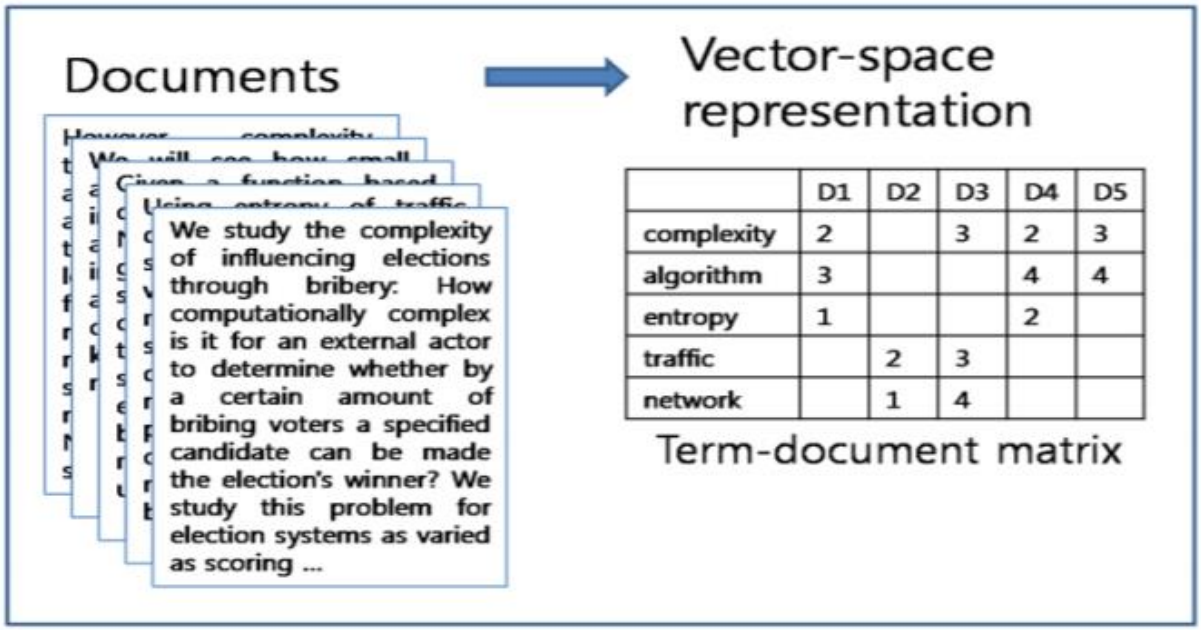
Para el caso de este trabajo se usará método de modelado de tópicos de Asignación Latente de Dirichlet (LDA) debido a las características del sitio Web que se va a atender y las posibilidades que tiene el LDA para modelar grandes *corpus* de texto y capturar la información semántica latente. Para ello es necesario entender los algoritmos de modelado de tópicos probabilísticos.

Tipos de Modelos

Los modelos de tópicos asumen que los *documentos* son una mezcla de *tópicos*, estos modelos tienen en cuenta que un *documento* puede tener múltiples temas y lo que buscan es encontrar las relaciones entre *documentos* y estructurar colecciones de *tópicos*. Los diversos modelos comparten una misma estructura funcional; según Silvestre (2018), los modelos crean un *corpus* de **documentos**, de estos extraen todas la palabras, aplican técnicas probabilísticas y generan uno de los *tópicos*, finalmente se asocia cada *documento* visto como un conjunto finito de palabras a los *tópicos* extraídos.

Los modelos de tópicos parten de la premisa que los documentos son una mezcla de *tópicos*. Se conforma un *tópico* como una distribución de probabilidad sobre un vocabulario fijo. De esta forma, un modelo con estas características es un modelo generativo para *documentos* ya que especifica un procedimiento probabilístico que permite generar documentos (imagen izquierda de la Ilustración 4).

En este sentido, entendemos por documento una secuencia de *n* palabras denotadas por $wi = (w_1, w_2, ..., w_n)$ donde w_i es la *i-ésima* palabra en la secuencia. Para elaborar un nuevo documento, seleccionamos una distribución de *tópicos*. Posteriormente, para cada palabra de un *documento* particular, seleccionamos un *tópico* al azar de acuerdo con esta distribución y, finalmente, se extrae una palabra. (Araya, 2018)



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Ilustración 4. Representación de documentos mediante una matriz palabra-documento. Fuente: (Araya, 2018)

Modelos Generativos

Un modelo generativo para análisis de *documentos* describe una serie de pasos, reglas y procedimientos que describen cómo generar *tópicos* en documentos basado en variables latentes no observadas directamente. Estas variables deben conformarse como un conjunto de variables latentes que puede explicar los datos observados, y hacer manejables estos datos reduciéndolos a los elementos esenciales.

Esta aplicación de modelos probabilísticos permite encontrar patrones semánticos no evidentes; y muy importante también, permiten reducir la dimensionalidad del corpus analizado de manera no supervisada, que pueden ser entrenados para ejecutarse continuamente en múltiples documentos, para ello es necesario aplicar técnicas para extraer la información.

Existen múltiples algoritmos que permiten realizar esta tarea para el caso particular se utilizará la Asignación Latente de Dirichlet

Latent Dirichlet Allocation, LDA

El LDA fue publicado en 2003 por D. Blei, A. Ng y M. Jordan y se presenta como el primer algoritmo de Topic Modeling. Se define como un modelo generativo probabilístico no supervisado que permite modelar grandes *corpus* de texto, “Este modelo asume que cada

término en un documento es generado a partir de un *tópico* que es tomado de una distribución de tópicos para cada documento." (Hammoe, 2018).

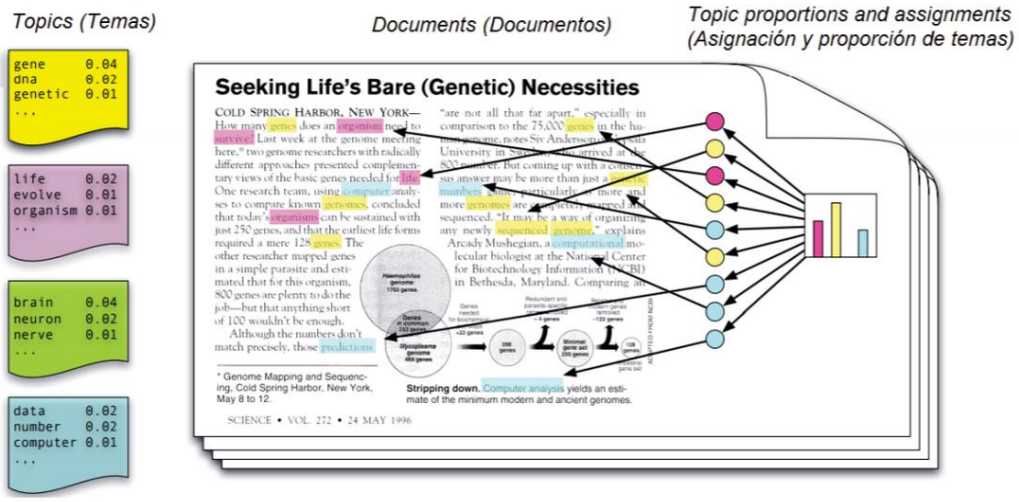


Ilustración 5 - Modelo intuitivo de la tecnica LDA. (Blei, 2012a).

Según Valdiviezo-Diaz & Hernando (2016), LDA asume que:

- En un texto pueden existir varios *tópicos* para un corpus o *documento* D
- Un *documento* puede ser utilizado como una bolsa de palabras ($W_{d,n}$) que está conformada por esos *términos*.
- Cada *documento* D es modelado como una mezcla aleatoria dando como resultados *tópicos* K latentes
- Cada *tópico* está caracterizado como una distribución probabilística sobre esos *términos* (distribuciones multinomiales). Es decir, cada *término* N tiene un índice de probabilidad asociadas a un *tópico* particular.
- Los *tópicos* son considerados como un conjunto (clúster) de *términos* que también poseen una probabilidad, lo cual convierte a cada clúster en un tópico.

"La idea de LDA es lograr reducción de dimensionalidad (con respecto a las representaciones tradicionales de vector de palabras) y fácilmente poder asignar probabilidades a *documentos* nuevos que no fueron parte del conjunto de datos de entrenamiento" (Arias, 2017).

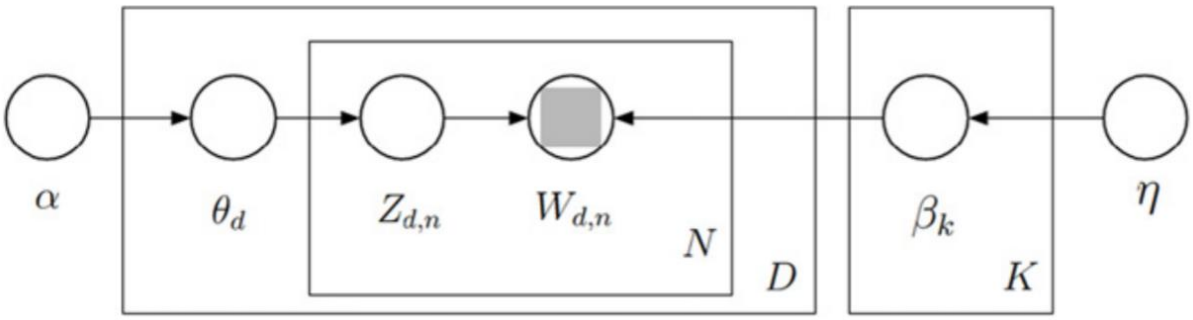


Ilustración 6 - Representación gráfica del modelo LDA. (Blei, 2012a)

Para entender claramente a que hace referencia (Blei, 2012a) crea un modelo gráfico en el cuál hay tres elementos fundamentales:

- K a los *tópicos* que puede tener toda la colección, estos en ultimas son los determinados por el vocabulario
- D son los *Documentos*, cada *documento* está conformado por el conjunto de *tópicos* en diferente distribución o proporción, lo que busca este algoritmo es determinar los *tópicos* que tiene el *documento* de forma no supervisada
- N se refiere al conjunto de *términos* encontrados en un documento

Los otros parámetros hacen referencia a:

- α hace referencia al parámetro de Dirichlet que está determinado por un número mayor que 0. El cual se va a conformar como la distribución de Dirichlet. Entre más cercano a 1 se encuentre el resultado, significa que esta distribución es más uniforme; y entre más cercano a 0 tiende más a ser exponencial y por consiguiente no balanceada. Entre más uniforme se encuentren los tópicos encontrados, son más similares y tienen más probabilidad de ser cercanos.
- β Es una matriz de probabilidad de palabras por *tópico*, entonces el tamaño es ***K-tópicos x amplitud*** (del vocabulario), si este es muy grande es inmanejable y es imposible de determinar la distribución para ver cual tiene más probabilidad de pertenecer al documento.
- $\acute{\eta}$ es el parámetro de la variable de respuesta

Este modelo procede de la siguiente manera:

1. Para cada *tópico* define una distribución de *términos* β_k
2. Se obtienen las proporciones de *tópicos* θ_d para el *documento D*
3. Para cada *termino* N:
    a. Se asigna un *tópico* $Z_n \mid \theta$
    b. Se obtiene un *término* $W_n \mid \beta$ muestreando un *término* del vocabulario y sigue una distribución multinomial

"Considerando la variable multinomial, Z, y el conjunto de distribuciones asociadas a cada termino, $p(W|Z, \beta)$ donde β es un hiper-parámetro. Este conjunto de distribuciones conformará los tópicos del documento. Cada documento pertenecerá a un tópico con cierta probabilidad, θ. Temporalmente se asume un conjunto de tópicos finito, K. Por tanto, la variable Z tomará valores entre los K posibles *tópicos* y θ será un vector de K dimensiones que define la probabilidad de pertenencia a cada tópico." (Silvestre Gómez, 2018).

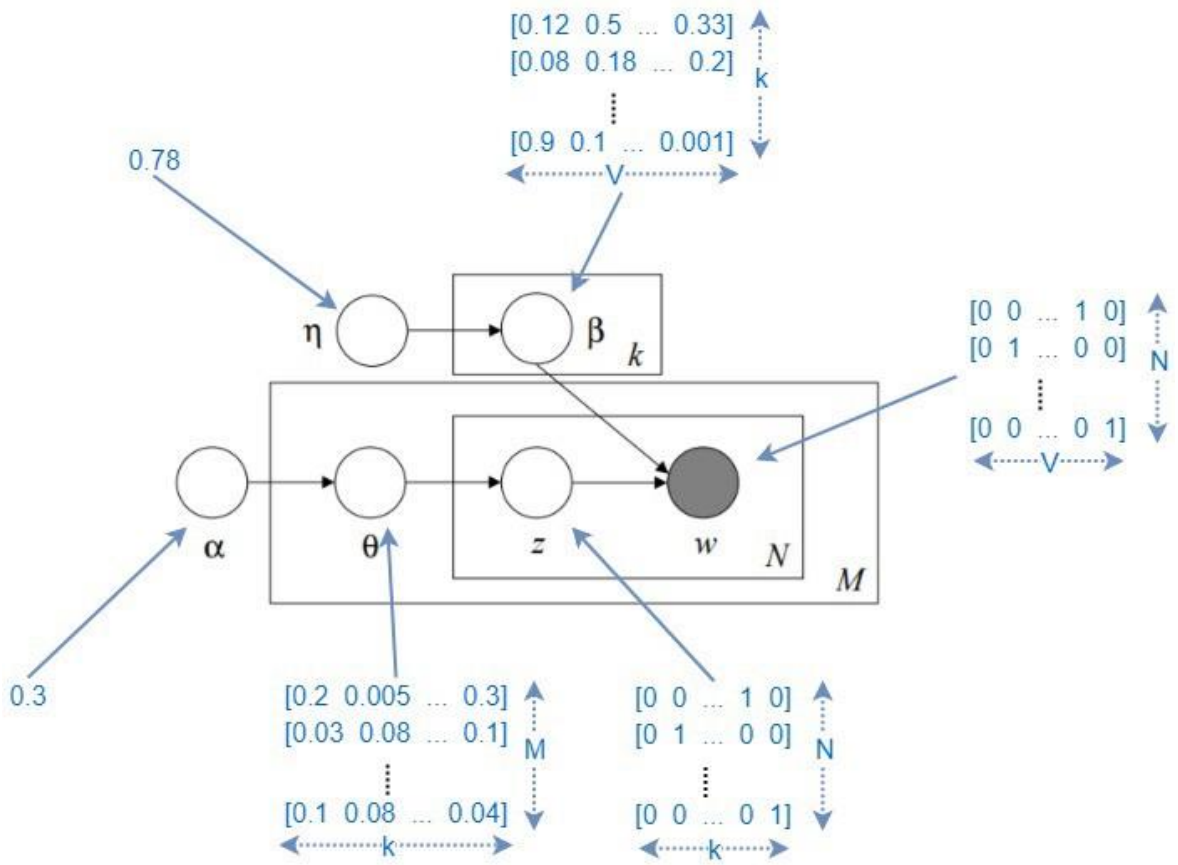


Ilustración 7 - Modelo gráfico del LDA. En este grafico se muestran cómo se comportan las distintas variables (tanto ocultas como observables) excepto para θ, z, y β que son distribuciones o hiper-parametros (Ganegedara, 2018)

En este modelo el orden de las palabras como el de los *documentos* no es un aspecto fundamental, lo importante es identificar los *tópicos* utilizados en cada documento, la frecuencia y la probabilidad de pertenecer a uno de ellos

Este modelo puede presentar problemas en situaciones específicas, al determinar grupos de *tópicos* no los relaciona solo los explicita, si los temas son muy similares no da buenos resultados y requiere análisis de documentos de gran longitud (Torres, 2017)

Similitud de textos

Un elemento que es posibilitado por la característica vectorial que LDA suministra a cada *tópico* y su cuantificación probabilística; es la de hacer comparaciones y cálculos de similitud, Para ello, lo que se calcula es la distancia entre los elementos, existe varios enfoques en esas métricas, es posible distinguir entre: 1) enfoques basados en caminos (grafos), 2) enfoques basados en conjuntos de características ontológicas y 3) enfoques basados en el concepto de “contenido de información” procedente de la teoría de la información. (Colombo Mendoza, 2017)

En este caso, debido a la retroalimentación vectorial del modelado de datos y su característica de conjunto de datos en tópicos y documentos con connotaciones semánticas, comúnmente se utilizan métodos para el cálculo de comparación de vectores, los más comunes y efectivos se refieren a la similitud de Jaccard y la distancia de Coseno

Distancia de Coseno

Esta distancia mide el grado de similitud entre dos documentos d1 y d2 utilizando el coseno del **ángulo** formado por las **r**espectivas representaciones vectoriales, cuanto mayor sea el valor obtenido, más similares son esos dos documentos comparados. Si no tienen términos en común el valor es de cero

“Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a uno. Si los vectores fuesen ortogonales el coseno se anularía, y si apuntasen en sentido contrario su valor sería -1. De esta forma, el valor de esta métrica se encuentra entre -1 y 1, es decir en el intervalo cerrado [-1,1]”, (Wikipedia, 2019).

Similitud de Jaccard

También se conoce como *intersección sobre la unión* o el sistema de *coeficiente de similitud de Jaccard*. Jaccard mide la similitud entre dos sets finitos de datos.

Si se consideran dos vectores de palabras d1 y d2 como conjuntos de términos (no se toma en consideración la frecuencia), se define la similitud Jaccard como el tamaño de la intersección entre d1 y d2 dividido entre el tamaño de la unión de los mismos conjuntos.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Entre mas cercano sea a volr a 1 mayor coeficiente de similitud existe

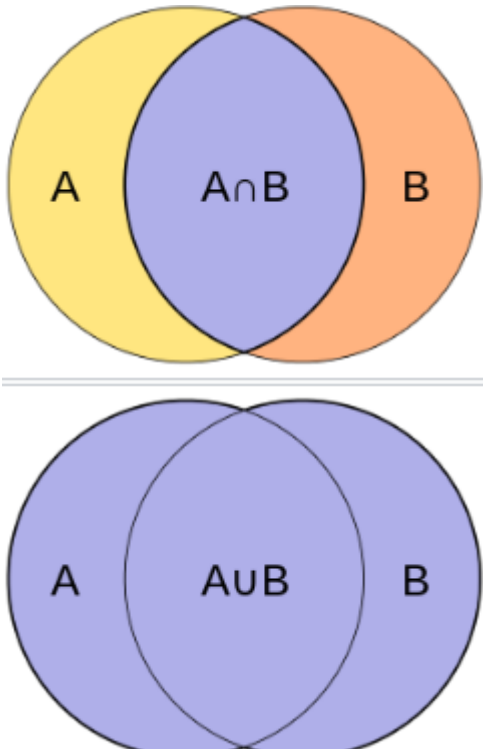


Ilustración 8 - intersección y unión de dos sets de datos A y B (wikipedia, 2019)

Objetivos

Diseñar un modelo de sistema de recomendación auto supervisado para un sitio web enfocado en contenidos

Para poder lograr este objetivo se requiere alcanzar los siguientes objetivos específicos

- Determinar un modelo para el análisis de los contenidos del sitio
- Identificar los tópicos fundamentales de cada documento que permita entender el contexto de cada página y de los usuarios que interactúan en ellas
- Clasificar otros factores claves y externos (análisis de búsquedas, palabras usadas como referencias, palabras usadas en redes sociales, etc.) que pueden afectar la valoración de los criterios de recomendación de ciertos tópicos
- Establecer criterios adicionales identificables no solo basados en contenidos, más enfocados en la actividad o valoración de los usuarios que permitan afinar la normalización de contenidos, añadiéndoles una capa colaborativa que permita recomendar contenidos acertados.
- Describir el proceso necesario para que pueda ser implementado el modelo del sistema de recomendación

Metodología

Cuando un usuario realiza una búsqueda en la Web usualmente tiene un objetivo específico, algo que quiere entender o conocer particularmente; cuando está explorando internet un sitio de noticias o contenidos, este usuario quiere ser informado sobre cosas que puedan ser interesantes para él, que sean relevantes y la intención es que el sitio web le suministre esas recomendaciones. Son estos escenarios en donde la experiencia del usuario depende casi exclusivamente de la calidad del contenido y que las recomendaciones conserven esa relevancia.

La principal dificultad es determinar cuáles son esos aspectos que le parecen interesantes a los usuarios, en el caso de una búsqueda es explicita (palabras claves) pero en el caso de las recomendaciones depende del contexto.

En el caso de las recomendaciones esos *términos* que direccionan los intereses del usuario al no ser explícitos pueden inferirse utilizando múltiples metodologías y algoritmos, y este es el punto central de esta propuesta, el establecimiento de una estrategia que permita identificar tópicos relevantes de un sitio web basado en contenidos (Eduteka).

Contextualización de la implementación

Eduteka es un portal educativo gratuito que hace parte de una propuesta de divulgación de material que permita la implementación de las TIC en los procesos de enseñanza. Tiene más de 20 años de trayectoria en la web y se enfoca en la publicación de contenidos y recursos con un carácter didáctico. Está dirigido a docentes de educación básica y media.

Eduteka tiene más de 1500 artículos no suficientemente bien indexados, solo utilizan un sistema manual de etiquetado que no está muy bien referenciado, ni es muy utilizado en el sitio web. Según cifras del año 2018, tiene más de 1 millón de visitas a esos artículos, el promedio de tiempo es superior a 4 minutos, el 78.6% de las personas acceden vía referencias de Google, el porcentaje de rebote es muy alto 85%. Lo que indica que los contenidos son atractivos, debido al tiempo de página, pero no saben a donde más ir después de visitar ese artículo particular, el cual usualmente es recomendado por los resultados de búsqueda de Google.

Para atender esta problemática, se realizará el planteamiento de un Sistema Recomendador de artículos, basado en un Sistema Hibrido Mixto, que utiliza el análisis de contenidos como factor fundamental, hace uso para mejorar el factor probabilístico de las tendencias búsqueda de cada artículo, así mismo, la retroalimentación implícita de los usuarios basados en el uso del contenido y análisis de “opiniones” previas de los ítems sugeridos. El planteamiento involucra el uso de diferentes enfoques para el análisis de contenidos y de la relación entre ítems y usuarios, buscando ser más acertado en las sugerencias o predicciones que les serán expuestas a los usuarios.

Solución planteada

El análisis de los datos es la etapa más importante, reducir la dimensionalidad es el primer paso, esto permite tener representaciones mejor interpretables y enfocar el análisis en los aspectos fundamentales. Para ello las técnicas de modelamiento probabilístico de tópicos ha sido ampliamente utilizada por su utilidad y fiabilidad.

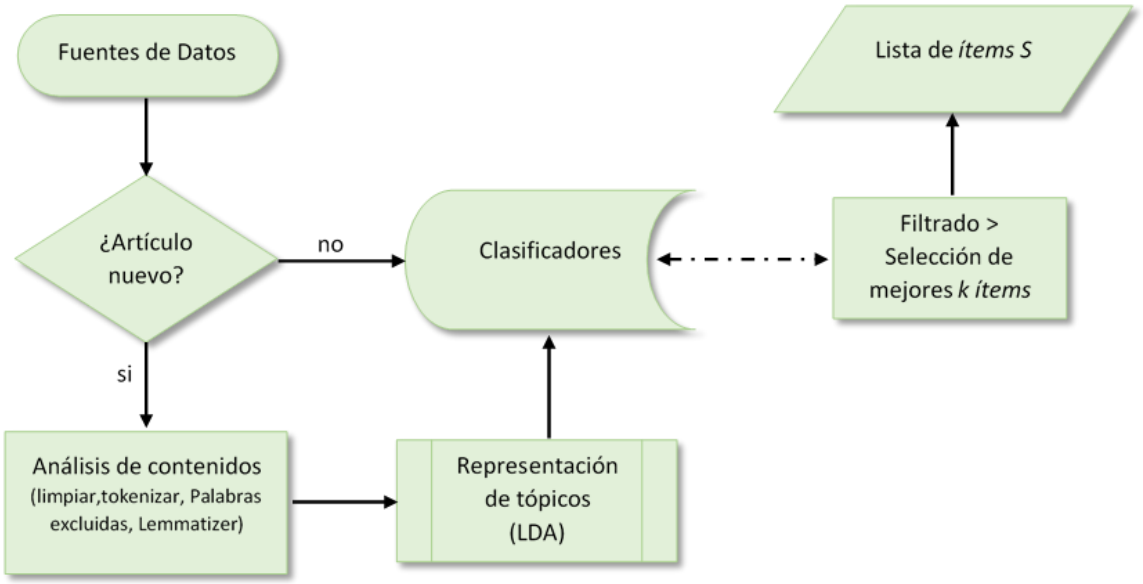


Ilustración 9 - Esquema modelo de implementación Sistema de Recomendación Eduteka. (Sánchez, B. 2019)

Fuentes de información

El primer paso es determinar la fuente de datos y las variables que son pertinentes de analizar, en este caso particular (caso Eduteka). El sitio web tiene como características principales, el conformarse como un sitio web enfocado en contenidos.

Para este trabajo de fin de maestría solo se enfocará en la sección que se denomina *artículos*, la cual corresponde a documentos propios o traducidos, usualmente extensos (después de analizar 30 documentos el promedio por documento es de 1500 palabras). Se conforman en documentos de trabajo y análisis de tendencias en educación, en las que se incorpora las TIC para mejorar procesos de aprendizaje, se encuentran también estudios publicaciones, estándares y otros documentos de carácter educativo

Estos *artículos* están clasificados internamente con unas categorizaciones determinada por los administradores del sitio, existe un sistema de etiquetado manual determinado por el editor en el momento de la creación del documento, existe así mismo, una categorización temática basada en 54 categorías conceptuales o temáticas determinadas por los administradores las cuales pueden ser utilizadas para ayudar al proceso de modelar los resultados.

Se va a utilizar la denominación *términos* a palabras con significado que puedan ser usadas para ser analizadas, estos *términos* al estar juntos pueden hacer referencia a un *tópico* (el cual es un tema con significado), un *documento* puede tener múltiples tópicos, para este caso se equipara el concepto de *documento* a cada artículo que ha publicado el sitio web.

Ya que se va a implementar un modelo *no supervisado* es indispensable hacer un normalización y reducción de los datos que van a ser analizados; primero, porque implicaría una cantidad enorme de recursos el procesamiento de bases de datos de esas dimensiones para cada documento y porqué para mejorar la eficacia de las predicciones es necesario esa normalización, “Una alta dimensión de los datos afecta el funcionamiento de varios clasificadores” (Jordan & Mitchell, 2015)

Si ya se ha realizado previamente el análisis de *tópicos* y *términos* de un *documento* se puede pasar a la etapa de clasificación; de lo contrario es necesario el análisis de contenidos.

Análisis de contenido

Después de tener los datos del *documento*, usualmente se obtiene un *corpus* de una alta dimensionalidad, se requiere determinar qué datos se analizan y cuáles no. Por ello se debe aplicar diferentes métodos de filtrado que permitan simplificar el análisis posterior y determinar datos que puedan ser pertinentes.

El análisis de contenido involucra una reducción inicial del corpus, tarea que debe ser parte del entrenamiento del modelo y se deben establecer las reglas para que van a ser aplicadas a los documentos del sitio, estas implican la Normalización, Stemming, Lemming y otros métodos.

Normalización

La eficacia de un análisis depende de la calidad de los datos. Según Gorakala & Usuelli, 2015, en general cada preprocesamiento de datos implica la limpieza de los datos, transformaciones, identificación de valores faltantes y cómo estos van a ser tratados. Para este preprocesamiento se pueden usar diversas técnicas de transformación que se aplica a cada documento para normalizar los datos

El primer paso de limpieza implica eliminar cualquier etiqueta HTML, URL, diéresis, caracteres especiales como @ o #, y otros que tienen que ser anexados al proceso de manera supervisada identificando cuales de ellos no aportan significativamente al análisis y al proceso como tal.

> *Se eliminan caracteres especiales y símbolos ! ? # $ % & ’ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ‘ { | } ~*

Posterior a ello se deben convertir las palabras a *tokens* o componente léxico, estos se conforman en *términos*. Este procedimiento se realiza sin problema usando librerías de *arrays* nativas de cualquier lenguaje de programación. Se codifica en utf8 para poder compartirse en protocolos como JSON o XML y se convierte a minúsculas.

Se eliminan usando un diccionario de datos palabras consideradas como *stop words* tales como, artículos, adverbios, preposiciones, etc. Palabras que carecen de significado semántico o que son consideradas como innecesarias para el análisis y que pueden generar ruido a los resultados. Se identifican también aquí, palabras de la lista negra, tales como

expresiones vulgares, modismos ofensivos u de otro carácter; usualmente estas supervisadas por editores.

```
"a","ante","bajo","cabe","con","contra","de","desde","durante","en","entre",
"mediante","para","por","segun","sin","sobre","tras","yo","tu","el","nosotro
s","vosotros","ellos","cual","cuales","quien","quienes","el","la","los","las
","un","una","unos","unas","este","esta","ese","esa","aquel","aquella","mi",
"tu","su","mis","tus","sus","al","por","con","de","para","venir","dar","mant
ener","hacer","poner","parecer","dejar","ir","tomar","ser","decir","ver","en
viar","poder","sobre","del","y","que","le","un","una","unas","unos","uno","s
obre","todo","tambien","tras","otro","algun","alguno","alguna","algunos","al
gunas","ser","es","soy","eres","somos","sois","estoy","esta","estamos","esta
is","estan","como","en","para","atras","porque","porqué","estado","estaba","
ante","antes","siendo","ambos","pero","por","poder","puede","puedo","podemos
","podeis","pueden","fui","fue","fuimos","fueron","hacer","hago","hace","hac
emos","haceis","hacen","cada","fin","incluso","primero","desde","conseguir",
"consigo","consigue","consigues","conseguimos","consiguen","ir","voy","va","
```

Ilustración 10 - Ejemplo de "stop words". (Sanchez, B. 2019)

Stemming

El siguiente paso aplicar *Stemming*[1], o una normalización de las palabras a su raíz eliminado posibles desambiguaciones semánticas. *Stemming* elimina los pronombres añadidos (me, se, sela, selo, selas, selos, la, le, lo, las, les, los, nos); gerundios y sufijos (iéndo, ándo, ár, ér, ír, ya, ye, yan, yen, yeron yendo, etc) del *corpus* del idioma español (28390 palabras).

El uso del *Stemming* puede aumentar la cantidad de datos obtenidos puesto que reduce la cantidad de términos a consultar. La decisión de usar *Stemming* o *Lemming* (que se verá a continuación) depende de la cantidad de datos en bruto que se obtengan del *tokenizer*. Por supuesto *Stemming* puede generar algunas divergencias semánticas, ya que para el sistema es igual el termino *Aplicado*, que el termino *aplicación* o el termino *aplicará;* Lo cual limita semánticamente el análisis de tópicos. (ver ilustración 7)

[1] Puede usarse para el Stemming la librería http://stemmer-es.sourceforge.net/ de Paolo Ragone, basado en el algoritmo de Martin Porter. Específicamente, está basado en el Spanish stemming algorithm de Snowball.

Aplaza	**aplaz**
Aplazada	**aplaz**
Aplazamiento	**aplaz**
Aplazó	**aplaz**
Aplica	**aplic**
Aplicables	**aplic**
Aplicación	**aplic**
Aplicaciones	**aplic**
Aplicada	**aplic**
aplicado	**aplic**
aplicados	**aplic**
aplican	**aplic**
aplicar	**aplic**
aplicara	**aplic**
aplicará	**aplic**
aplicáramos	**aplic**
aplicarán	**aplic**
aplicáremos	**aplic**
aplicarla	**aplic**
aplicarle	**aplic**
aplicarlos	**aplic**
aplicaron	**aplic**
aplicarse	**aplic**
aplico	**aplic**
aplicó	**aplic**

Ilustración 11 - Ejemplo uso de Stemming. (Sánchez, B. 2019.)

Lemming o Lematización

Lemming o *Lematización* es un proceso que analiza del corpus lingüístico el *Lema* o unidad semántica con significado (palabra o término en este caso). Ese lema es la reducción principal de esa palabra, es la representante lingüística de las conjugaciones o flexiones lingüísticas del término. El resultado del proceso es la definición de las características esenciales de esa palabra, raíz, lema, significado, estructura y otros.

Para poder utilizar el *lemming* es necesario usar librerías tipo WordNet[2] que permiten transformar términos en su raíz semántica. Wordnet es una base de datos léxica la cual

[2] Wordnet en su versión en español https://adimen.si.ehu.es/web/MCR es suministrada por la Universidad Politécnica de Cataluña , hace parte del proyecto de EuroWordNet estructurado de manera similar que the American wordnet for English (Princeton University, 2010)

agrupa conjuntos de sinónimos llamados *synsets*, los cuales permiten diferenciar las relaciones semánticas entre los conjuntos de sinónimos. (ver ilustración 8).

Aplicado				
categoría **ADJ**	lema APLICADO	género masculino	número singular	
categoría **V**	lema APLICAR	tiempo participio		
Aplicación				
categoría **N**	lema APLICACIÓN	género femenino	número singular	
Aplicará				
categoría **V**	lema APLICAR	número singular	persona **3**	tiempo futuro indicativo

Ilustración 12 - Ejemplo Lemming o Lematización. (Sánchez, B. 2019.)

Frecuencias

Dentro del proceso de filtrado aparecen tres tipologías de *términos*, las del alta, media y baja frecuencia, usualmente las palabras de *alta frecuencia* aportan muy poco a la caracterización del documento por ser palabras comunes que se repiten en casi todos los documentos impidiendo la diferenciación, las palabras de muy baja frecuencia usualmente aumenta la dimensionalidad del *corpus*; son las palabras de media frecuencia las que se caracterizan como palabras claves o representativas.

Por ello es necesario eliminar tanto los picos como la cola de palabras muy específicas para continuar con el siguiente paso que es del modelado de datos.

Representación de los tópicos

Una vez terminado el proceso de análisis de datos, el resultado debe ser modelado para reducir a niveles comprensibles la cantidad de términos y hacer posible un análisis posterior. Para ello se crea un diccionario con los términos encontrados y el corpus o bolsa de palabras aspecto necesario para poder implementar el método de *Asignación de Dirichlet Latente* (**LDA**).

Para poder entender la real importancia de LDA en este paso, supongamos que tenemos un artículo con 1000 palabras o *términos*; y tenemos 1000 *documentos*. Después de limpiar

cada documento encontramos que tenemos 500 *términos* analizables. En este caso tenemos dos problemas; el primero hace referencia a la cantidad de elementos a analizar que se vuelve inmanejable por el sistema (normalmente); y segundo es qué no sabemos cómo categorizar automáticamente cada *documento*.

La primera opción es evaluar cada *término* de cada *documento* con respecto a su aparición en los demás *documentos*, esto generaría una serie de 500*1000 es decir 500.000 hilos de análisis. Eso es un gasto enorme de recursos computacionales y tiempo

Para ello se aplica el LDA. El objetivo es reducir la cantidad de *tópicos*. Un artículo con 1000 *términos* puede tener numerosos *tópicos,* y en un proceso *no-supervisado* esos *tópicos* no son deducibles, están latentes (dispuestos a ser descubiertos). Esa es la información que permite reducir la cantidad de hilos de análisis en este caso se conectan los *términos* con los *tópicos*; y si el resultado es adecuado (es decir una alta probabilidad de que un *término* sea asociado a un *tópico*), se pueden conectar *tópicos* a *documentos,* en función a los *tópicos* que cada *documento* trata

De esta manera si se solicita 10 *tópicos* y cada *tópico* relaciona 500 *términos* cómo habíamos ejemplificado antes, necesitamos para este caso (usando LDA) 1000*500 hilos que conectan *documentos* a *tópicos* y se necesitan 10*500 hilos para conectar *tópicos* a *términos*. De tal manera al aplicar LDA estamos reduciendo para ese documento

EL LDA está evaluado entre las técnicas más efectivas para poder abstraer de un cuerpo textual en termino de *tópicos* y encontrar relaciones semánticas entre ellos.

> "LDA logra una importante reducción dimensional como método de representación de texto y, al combinarlo con la divergencia KL, logro resultados comparables a los experimentos que utilizaron vectores de palabras con coseno y Jaccard… logrando abstraer miles de palabras en menos de 60 temas para los experimentos principales." (Arias, 2017)

Para la implementación de LDA se necesita afinar el algoritmo dependiendo de las características del corpus y el número de *tópicos* deseados, si hay muchas palabras repetidas significa que este número K es muy grande, para ello es necesario bajar el número de K especificando así que hay menos cantidad de *tópicos* requeridos para este proceso.

- Num Topics = 2 has Coherence Value of 0.03468
- Num Topics = 8 has Coherence Value of 0.04378
- Num Topics = 14 has Coherence Value of 0.04644
- Num Topics = 20 has Coherence Value of 0.04541
- Num Topics = 26 has Coherence Value of 0.04625
- Num Topics = 32 has Coherence Value of 0.04815
- Num Topics = 38 has Coherence Value of 0.05041
- Num Topics = 44 has Coherence Value of 0.05063

Otro aspecto fundamental es la selección del valor de alfa, este determina que tan similares, o agrupados, o diferentes son unos *documentos* entre sí. Este valor tiene que ser afinado en la fase de entrenamiento, y depende de la similitud temática que tienen los *documentos* analizados. Es importante recordar que el valor de alfa debe ser <0 y >1, cuando los valores de alfa son:

- **muy pequeños >=0.2** en este caso los documentos obtenidos son disimiles y hay pocos *tópicos* para cada *documento*,
- **Intermedio < 0.5 >** en este caso es un valor promedio, este valor debería ser revisado probando cuál es el rango con mayor éxito en el proceso
- **Alt >=0.8 .** Para este caso el algoritmo asume que los *documentos* pueden contener una combinación más amplia de *tópicos*

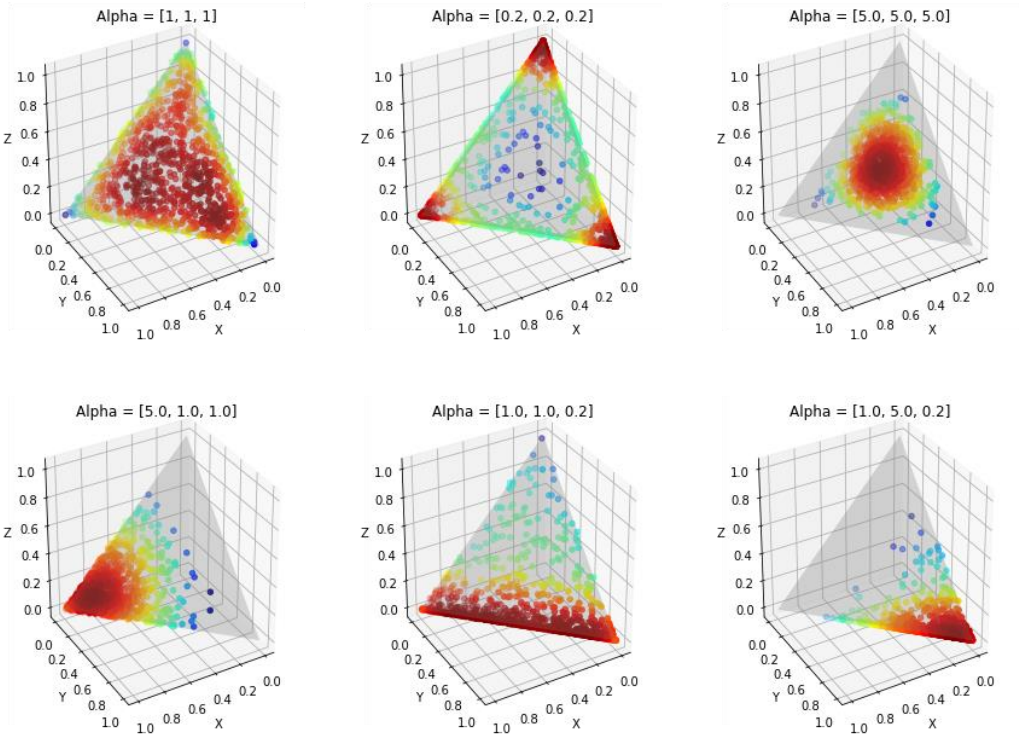


Ilustración 13 - Cambios en la distribución de θ con diferentes valores del α (Ganegedara, 2018)

Ahora, es necesario aclarar que los *tópicos* resultantes no son nominales, es decir el sistema no les pone nombre a esos conjuntos de *términos*, tienen un carácter imaginario. El resultante es un vector con términos acompañados de una probabilidad (0.0505*Computadores,0.0354*Errores,0.0412*Virus,0.0284*Seguridad).

Estos datos son los que tienen que ser almacenados, analizados y perfilados para poder realizar las recomendaciones.

Hasta este punto, se ha determinado las probabilidades de los *términos* y *tópicos* para poder realizar perfiles de, *documentos* y *tópicos* hallados a través de la aplicación del LDA en los datos existentes

Gestor de Perfiles

A partir del modelado de los datos se obtienen unas matrices probabilísticas, en la cuales están representados para cada documento una serie de *tópicos*, con dimensiones variables. En la ilustración 14 se puede observar una representación general del resultado de LDA, en donde se ejemplifica un modelo con K *tópicos* , para d_1, d_3 *documentos* con un total de T_n *términos.*

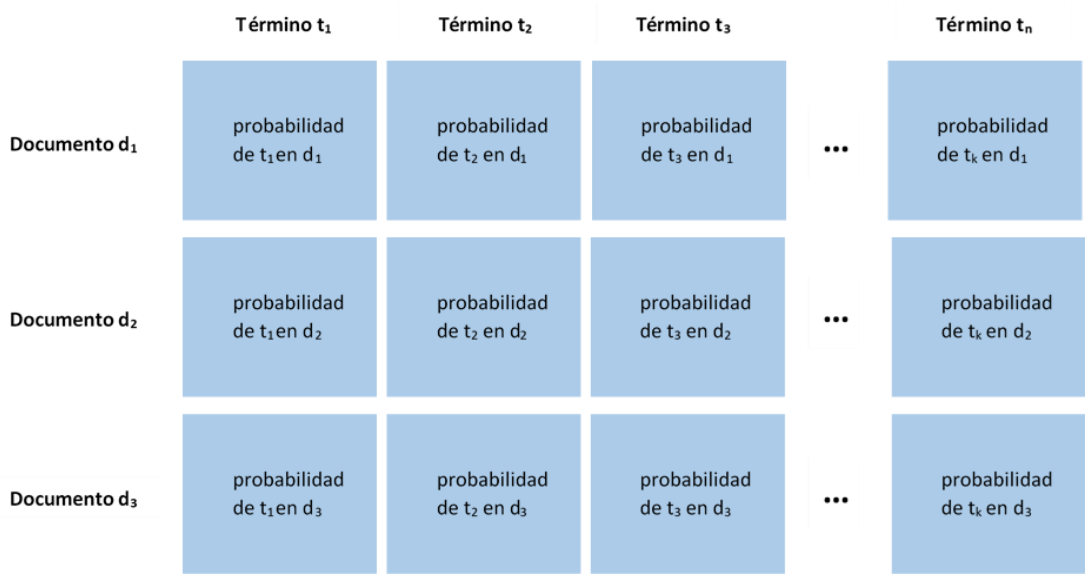


Ilustración 14 - Representación de documentos con LDA (Sánchez, B. 2019)

Con estos datos obtenidos es necesario crear estrategias para establecer perfiles de recomendación, Estos surgen a través del análisis de los *tópicos* y sus relaciones; y también entre *tópicos* y *usuarios*. Estos perfiles deben ser fácilmente consultados para poder

completar el proceso de recomendación. Identificando claramente documentos que puedan ser recomendados

Al realizar el primer análisis LDA se obtiene una serie de *tópicos* que deben ser evaluados para poder determinar el *tópico dominante* para ese documento- Este hace referencia al *tópico* con mayor peso probabilístico manifiesto al conjunto de *términos* que lo conforma. Así mismo se puede evaluar para ese *tópico* los *términos* con mayores relevancias ya que LDA suministra el vector de probabilidad para cada *termino.*

	Topic0	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	dominant_topic
Doc0	0	0	0	0	0	**0.14**	0	0	0	**0.84**	**9**
Doc1	0	0.05	0	0	0.05	**0.24**	0	**0.65**	0	0	**7**
Doc2	0	0	0	0	0	0.08	**0.2**	0	0	**0.71**	**9**
Doc3	0	**0.55**	0	0	0	**0.44**	0	0	0	0	**1**
Doc4	**0.16**	**0.29**	0	0	0	**0.53**	0	0	0	0	**5**
Doc5	0	0	0.05	0	0	0	0	**0.12**	0	**0.83**	**9**
Doc6	0	0	0	0	0	**0.88**	0.1	0	0	0	**5**
Doc7	0	0	0	0	0	**0.99**	0	0	0	0	**5**
Doc8	0	0	0.08	**0.67**	0	0	0	0	**0.24**	0	**3**
Doc9	0	0	**0.74**	0	0	**0.14**	0	0	**0.11**	0	**2**
Doc10	0	0	0	0	**0.41**	**0.16**	0	0.06	0	**0.36**	**4**
Doc11	0	0	0	0	0	0	0	**0.97**	0	0	**7**
Doc12	0	0	0	**0.44**	0	0.04	0	**0.27**	0	**0.24**	**3**
Doc13	**0.14**	0	0	0	0	0.07	**0.57**	0.08	0	**0.13**	**6**
Doc14	0	0	0	0	**0.78**	**0.22**	0	0	0	0	**4**

Ilustración 15 - Ejemplificación de la forma de selección del tópico dominante (Sánchez, B. 2019)

Toda esta etapa involucra un periodo de entrenamiento y modificación de variables para poder obtener los valores adecuados que permitan filtrar los *términos* más probables a los requerimientos del usuario. Este se conforma como uno de los principales inconvenientes del sistema de modelado de datos de carácter no supervisado. Pero después de pasar por esa etapa de entrenamiento, estos métodos de clasificación pueden crear automáticamente las categorías de datos para múltiples documentos con buen eficiencia y efectividad.

Clasificadores

El objetivo en esta etapa es perfilar tanto el *documento* como el usuario para poder determinar de mejor manera la recomendación a realizar. Por tanto, el aspecto que hay que atender es el análisis tanto del contenido como del contexto del *documento*. Para ello se atiende estas tres dimensiones, el filtrado por contenido, por perfil del documento y perfilado del usuario.

Filtrado por contenido:

Se realiza de manera autónoma, no supervisada para entender cómo se estructura el contenido, las relaciones internas y externas de este, la probabilidad de pertenecer a un tema u a otro y la similitud con otros *documentos* o *tópicos*. Para ello, se utiliza algoritmos de probabilidad estadística que tienen en cuenta la valoración obtenida en el modelado de datos.

Para la categorización de *tópicos* usualmente se usan los algoritmos de comparación como *Jaccard*, los cuales permiten por medio de análisis de las representaciones vectoriales de dos documentos de texto analizar su similitud; también se analiza la *densidad* de los *términos* en cada *documento* y *tópico*. Estos algoritmos usan probabilidad para estimar la posibilidad de que un tema pertenezca a un tema específico. Esto permite crear un modelo predictivo que puede ser almacenado, generando una primera asociación entre ítems y preferencias de usuario, las cuales pueden ser evaluadas posteriormente.

Para poder facilitar a comparación de *tópicos* entre *documentos*, se deben identificar los *documentos* más representativos para cada *tópico*, y la distribución probabilística de cada uno de ellos respecto a los *tópicos* dominantes del corpus (es decir de todos los documentos).

Es necesario contrastar la similitud entre estos *tópicos*, para ello se utiliza el método de Jaccard el cual determina un coeficiente de coherencia. Así aquellos clústeres que estaban disgregados o fuera de contexto vectorial quedan con unos índices de similitud bajos y pueden ser excluidos.

Ya que los *tópicos* y *documentos* están relacionados, identificando claramente qué *tópicos* pertenecen a cada documento, y como se relaciona con estos, se pueden hacer seleccionar

y hacen parte de la matriz *K-item* (la cual contendrá todos los documentos que van a ser recomendados)

Filtrado por perfil documento:

El perfil del *documento* está intrínsicamente asociado a la relación *documento* con los *términos*. Desde esa mirada, los *documentos* pueden ser comparados con otros *documentos* directamente, se puede analizar sus similitudes tanto semánticas como probabilísticas (la cuales fueron obtenidas por medio de LDA)

Cada documento es caracterizado como un conjunto de *términos* que puede ser comparable respecto a otros, para ello, se hace uso de métodos de análisis de similitud con la posibilidad de obtener un coeficiente que puede ser evaluado posteriormente. Para mejorar el rendimiento de este proceso las comparaciones se pueden realizar sobre los términos filtrados en el paso anterior; y es debido a esta posibilidad la cual ha determinado que cada *término* pertenece a un *documento* con un vector probabilístico que puede afectar positiva o negativamente la aparición de ciertos *términos* en el momento de la comparación.

Además del conjunto de *términos* que son inherentes a su estructura semántica, cada *documento* puede asociarse otras características clasificatorias. Estas pueden ser asignadas por el editor o autor tales como, etiquetado, clasificación, categorizaciones, etc. Que deben ser tenidas en cuenta y pueden convertirse en términos con algún coeficiente positivo para la clasificación.

Otros elementos que pueden ser asociados para perfilar el contenido del documento corresponde a términos obtenidos en fuentes externas. Nos referimos a información asociada al documento que no necesariamente ha sido caracterizado por el autor, los términos de búsqueda pueden ser comparados con los de los tópicos y asignársele un factor

de corrección superior, términos que usan otros sitios para enlazar con el documento (back links), y otras que deben ser clasificadas y anexadas al análisis.

Consulta	↓ Clics	Impresiones
eduteka	51.379	70.294
taxonomia de bloom	36.095	1.636.985
para que sirve power point	20.766	149.741
proceso de escritura	14.812	54.906
que es power point y para que sirve	9.426	42.557
partes de un rio	9.277	79.690
taxonomía de bloom	7.602	312.860

Ilustración 16 - Palabras de búsqueda usadas para encontrar el sitio web (sánchez, Boris. 2019)

Con estos elementos para cada *documento*, se puede definir qué otros *documentos* están altamente relacionados por similitud, los cuales pueden anexarse a la matriz *k-items* y posteriormente ser evaluados para ser recomendados al usuario.

Filtrado por perfilado del usuario

Uno de los inconvenientes es que el sitio web (Eduteka) no tiene un sistema de control de sesiones de usuario, ni colecciona datos de este. Existe un sistema de registro el cual es usado para almacenar datos de una herramienta específica, por tanto, casi todo el peso del análisis que permitirá la recomendación estará basado en el contenido y el contexto. Sin embargo, se puede hacer uso de algunos recursos externos que permiten convertirse en elementos de juicio en el momento de las recomendaciones

Al no poder garantizar un perfilado especifico del usuario y de sus interacciones, se sugiere realizar un análisis y seguimiento de las interacciones de los usuarios con respecto a los documentos, y en ocasiones con algunos términos. Por ejemplo, al determinar la actividad de los enlaces y el análisis de esos términos de enlace respecto al contenido; la interacción con los enlaces sugeridos y características analíticas de estadísticas del uso de los contenidos por parte de los usuarios.

Nivel de ruta de página 2	Número de visitas a páginas	Número de páginas vistas únicas	Promedio de tiempo en la página	Porcentaje de rebote	Porcentaje de salidas
	143.357 % del total: 20,14 % (711.970)	126.864 % del total: 20,38 % (622.367)	00:04:20 Media de la vista: 00:03:47 (14,84 %)	84,59 % Media de la vista: 84,34 % (0,30 %)	81,79 % Media de la vista: 79,70 % (2,62 %)
1. /TaxonomiaBloomCuadro	**8.896** (6,21 %)	8.303 (6,54 %)	00:06:13	91,27 %	90,37 %
2. /Netiqueta	**8.011** (5,59 %)	6.754 (5,32 %)	00:05:10	83,80 %	81,44 %
3. /Quimica_100Preguntas	**6.153** (4,29 %)	5.334 (4,20 %)	00:04:24	86,39 %	85,75 %
4. /ProcesoEscritura1	**4.778** (3,33 %)	4.144 (3,27 %)	00:06:37	88,20 %	86,44 %
5. /Teclado2	**3.986** (2,78 %)	3.106 (2,45 %)	00:04:49	78,55 %	74,91 %
6. /TaxonomiaBloomDigital	**3.919** (2,73 %)	3.531 (2,78 %)	00:04:31	85,02 %	83,69 %
7. /ListaVerbos	**3.857** (2,69 %)	3.596 (2,83 %)	00:05:31	92,87 %	91,03 %

Ilustración 17 - Páginas visitadas, información de Google analytics (Sánchez, B. 2019)

Esa información de comportamiento del usuario en el sitio permite interpretar intereses por parte del usuario, los documentos con mayor visita y tiempo de permanencia pueden sumársele algún factor de corrección, índice de rebotes del documento, posición en los resultados del buscador, documentos con mayor número de back links, y otros factores por delimitar.

Páginas más enlazadas

Página de destino	Enlaces entrantes	Sitios web con enlaces
http://eduteka.icesi.edu.co/	149.895	1.273
http://eduteka.icesi.edu.co/me/ingresar.php	37.516	78
http://eduteka.icesi.edu.co/reduteka/	33.221	25
http://eduteka.icesi.edu.co/articulos/PoliticaUso	31.027	52
http://eduteka.icesi.edu.co/articulos/datosPersonales	30.917	32
http://eduteka.icesi.edu.co/planaula/	30.291	24

Ilustración 18 - Paginas enlazadas Back Link, Google console (Sanchez, B. 2019)

Desafortunadamente la toma de estos datos debe hacerse manual y periódica para que afecte positiva o negativamente el coeficiente de corrección para la recomendación. Asi mismo, el comportamiento del usuario debe ser consignado y evaluado.

Es importante contar con herramientas de evaluación de la interacción del usuario con el documento, hay múltiples sistemas de evaluación que pueden ser suministrados para que el

usuario opine sobre el documento y esta evaluación puede tener un coeficiente en el análisis.

Las redes sociales también son fuente de información, cuantos *likes* o *tweets* tiene un documento particular puede afectar positivamente el peso probabilístico. Así mismo, el número de comentarios realizados, numero de visitar por red, etc.

Para este prototipo no está asumido estos factores matemáticamente, debido a las implicaciones necesarias para que sea una implementación potente, la creación de protocolos de recolección de datos y otras que exceden las posibilidades objetivas del TFM. Pero se deja consignada la importancia de estos factores de personalización que van más allá del análisis semántico del contenido de un documento

A partir de estos análisis se puede determinar los ítems que van a ser publicados, el objetivo del sistema es la selección de una lista de *top-N items* los que tengan *items-S* de mayor probabilidad

Resultados (ítems S)

Esta es la etapa final de proceso; en este punto se ha obtenido un conjunto de enlaces relacionados al contenido del *documento* que pueden ser recomendados al usuario, almacenados en una matriz k-item. Esta matriz ha sido alimentada por los diferentes métodos de filtrado, análisis de contenido y análisis de los documentos.

Es necesario establecer una fase de entrenamiento del sistema para poder determinar cuáles serán los enlaces que se deben mostrar al usuario, en ambos sistemas de filtro se obtienen una lista de documentos con características similares al que se está visitando.

Debido a que se tiene en el sistema una relación de similitud tanto de *términos*, *tópicos* y *documentos* respecto al *documento* que se está analizando, se puede hacer inferencias de cuáles son los enlaces matemáticamente más significativos; pero no necesariamente más pertinentes. varios autores han trabajado respecto a cuál es el método más adecuado para determinar cuáles son las recomendaciones para desplegar.

En este caso se utilizará el concepto del Context-Aware Recommender Systems de (Adomavicius, Mobasher, Ricci, & Tuzhilin, 2012) los cuales plantean que un sistema recomendador se debe hacer un análisis de como interactúan sus recomendaciones y los

usuarios; y esto ayuda a que el sistema “aprenda” a medida que hay mayor número de interacciones. La importancia es que en este modelo el sistema va mejorando sus predicciones a través del comportamiento implícito y explícito de los usuarios frente a las recomendaciones.

Para ello debe almacenarse todas las interacciones con los enlaces y deben convertirse en factores de peso al momento de realizar el análisis de los Top-n ítems que se van a mostrar, en este caso se han tenido en cuenta, el factor de similitud entre *documentos* y el valor probabilístico de los *tópicos* respecto al del *documento* de origen.

Prototipo

El propósito de este TFM es la creación de un modelo no supervisado de un sistema de recomendaciones, dada las características del sitio con un poco grado de personalización y más enfocado en la producción de contenidos la clasificación de los textos es el elemento clave para que la generación de predicciones sea acertada.

Debido a que el lenguaje utilizado para el procesamiento de los datos fue PHP se utilizó un script propio para el preprocesamiento de datos. Para la ejecución de este LDA se utiliza el algoritmo *NlpToolsNatural language processing in php* creado por Sam Hocevar sam@hocevar.net bajo licencia abierta WTFYW, estos scripts fueron modificados para poder integrarlo a las necesidades específicas de este prototipo[3].

Para visualizar y ejecutar los scripts se puede visitar el siguiente sitio http://eduteka.net/LDA/NLP1/tests/index.php , aquí está cada etapa del proceso y se ejecuta en tiempo real. Se utiliza un servidor de prueba de Eduteka con características menos robustas, y se usa a MySQL como base de datos seleccionada, se programó en entorno 5.4 de PHP.

Tabla	Acción	Filas	Tipo
articulos	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,193	MyISAM
LDA_GenDoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	118,824	MyISAM
LDA_SimDoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,370,217	MyISAM
LDA_Simil	Examinar Estructura Buscar Insertar Vaciar Eliminar	5,965	MyISAM
LDA_terminos	Examinar Estructura Buscar Insertar Vaciar Eliminar	500	MyISAM
LDA_TopTopic	Examinar Estructura Buscar Insertar Vaciar Eliminar	29,706	MyISAM
main	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,593	MyISAM
terminos	Examinar Estructura Buscar Insertar Vaciar Eliminar	527,318	MyISAM
terminosFrecuencia	Examinar Estructura Buscar Insertar Vaciar Eliminar	56,668	MyISAM
terminosFrecuenciaLDA	Examinar Estructura Buscar Insertar Vaciar Eliminar	2,925	MyISAM
terminosFrecuenciaTop	Examinar Estructura Buscar Insertar Vaciar Eliminar	2,266	MyISAM
11 tablas	Número de filas	2,117,175	MyISAM

Ilustración 19 - Base de Datos usada para el prototipo (Sánchez, B. 2019)

El primer paso necesario para la consecución del objetivo implica organizar y entender grandes cantidades de información, descubrir patrones, clasificar esos patrones y ver como se relaciona con los documentos analizados. Para este caso se analizarán 1.193

[3] Para la mayor parte de los procesos se usará una herramienta llamada NipTools. http://php-nlp-tools.com/documentation/ escrita en PHP la cual permitirá

documentos con aproximadamente 1500 términos por *documento*, con la información de estos documentos se conforma el *corpus*.

Normalización

Los contenidos del sitio web intervenido es bastante homogéneo (relacionado con educación y tecnología) sin embargo al estructurarse como un *corpus* sus términos son altamente heterogéneos, en ellos se encuentran caracteres especiales y codificaciones de HTML, signos de puntuación y otros, estos datos deben ser organizados para poder ser utilizados posteriormente, para facilitar el procesamiento de datos, se almacenaron en base de datos en una tabla llamada *artículos*

```
TABLE `articulos` (
 `idA` int(6) NOT NULL,
  `titulo` text NOT NULL,
  `intro` text NOT NULL,
 `contenido` mediumtext NOT NULL,
  `autor` varchar(255) NOT NULL,
  `fechaPub` date NOT NULL)
```

Análisis de contenido

El siguiente paso implica la aplicación de métodos para limpiar y estructurar el texto, es necesario reducir la dimensionalidad del corpus a través de procesos de filtrado, esto implica realizar varios procesos:

Convertir caracteres espéciales de HTML (acentos, ñ, etc.), Convertir a minúsculas el texto, espacios en blanco no necesarios y se eliminan caracteres especiales y símbolos ! ? # $ % & ’ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ‘ { | } ~

```
$content = mb_strtolower($content);
$content = strip_tags($content);
$punctuations = array(',', ')', '(', '.', "'", '"',
'<', '>', '!', '?', '/', '-',
'_', '[', ']', ':', '+', '=', '#',
'$', '&quot;', '&copy;', '&gt;', '&lt;',
' ', '&trade;', '&reg;', ';',
chr(10), chr(13), chr(9));

$content = str_replace($punctuations, " ", $content);
$content = preg_replace('/ {2,}/si', " ", $content);
```

También, se eliminan palabras de menos de 3 caracteres

```
$this->encoding = $encoding;
mb_internal_encoding($encoding);
$this->contents = $this->replace_chars($params['content']);

$this->wordlengthmin = $params['min_word_length'];
$this->wordoccuredmin = $params['min_word_occur'];
```

Lo siguiente es filtrar palabras *StopWords*, es decir palabras comunes inherentes a nuestro idioma que no son relevantes, artículos, preposiciones, conjunciones, adverbios, pronombres, etc. Dentro de estas StopWords se anexaron palabras con significado soez, racista o denigrante para ser excluidas. Estas palabras también deben incluir términos genéricos o palabras comunes a los documentos las cuales no tienen un aporte significativo.

```
    "le", "les", "llegó", "lleva", "llevar", "lo", "los", "lt", "luego"
    , "lugar", "lópez", "manera", "manifestó", "mantener", "mayor",
    "me", "mediante", "mejor", "mencionó", "menos", "mi", "mientras",
    "mio", "mis", "misma", "mismas", "mismo", "mismos", "modo",
    "momento", "mucha", "muchas", "mucho", "muchos", "muy", "más",
    "nada", "nadie", "ni", "ninguna", "ningunas", "ninguno", "ningunos"
    , "ningún", "no", "nos", "nosotras", "nosotros", "nuestra",
    "nuestras", "nuestro", "nuestros", "nueva", "nuevas", "nuevo",
    "nuevos", "nunca", "o", "ocho", "otra", "otras", "otro", "otros",
    "p", "para", "parece", "parecer", "parte", "partir", "pasada",
    "pasado", "pero", "pesar", "poca", "pocas", "poco", "pocos",
    "podeis", "podemos", "poder", "podria", "podriais", "podriamos",
    "podrian", "podrias", "podrá", "podrán", "podría", "podrían",
    "poner", "por", "porque", "porqué", "posible", "previos", "primer",
    "primera", "primero", "primeros", "principalmente", "propia",
    "propias", "propio", "propios", "próximo", "próximos", "pudo",
    "pueda", "puede", "pueden", "puedo", "pues", "que", "quedó",
    "queremos", "quien", "quienes", "quiere", "quién", "quot", "qué",
```

```
$s = split(" ", $this->contents);
$k = array();
foreach( $s as $key=>$val ) {
    if(mb_strlen(trim($val)) >= $this->wordlengthmin  && !in_array(trim
        ($val), $common)  && !is_numeric(trim($val))) {
        $k[] = trim($val);
    }
}
$k = array_count_values($k);
$occur_filtered = $this->occure_filter($k, $this->wordoccuredmin);
arsort($occur_filtered);
```

Los resultados de este script se almacenan en la tabla de *términos*, la cual va a contener el nuevo corpus con el cual se va a trabajar, se identifican los *términos* para cada documento. En este caso se procesaron 1.193 documentos y se obtuvieron 527.318 *términos*, este se había reducido casi en un 40% aproximadamente del corpus original con el proceso de filtrado anterior, cada termino se almacena como una fila de la base datos y se identifica a que documento pertenece (idA).

idTB	idA	termino
226	1	retroproyector
227	1	invención
228	1	patrones
229	1	complejidad
230	1	ensayo
231	1	presente
232	1	apoyar

Un ejemplo de la reducción por el proceso de filtrado inicial, para el caso de un Documento en estado original que presenta 1.398 palabras las cuales a través de la limpieza quedaron en 406 palabras

Número total de terminos filtrados: **406**

```
valoración información aprendizaje proceso integral tema mejorar examen aspectos amplio respuestas respuesta enseñanza
evaluar calificación disponible exige profesor tipo áreas docentes tradicional fuente educación integración fácil medir
desempeño cmi problemas mayoría maestro sección conocimiento matriz objetivos julio alumnos juicio suministra
presentamos valorar realización centrada formas portafolios herramienta orden materias positivosaspectos ideal internet
cantidad aumenta calificar presenta jul fecha cognitivos contenido ago correspondiente profundidad assessment realiza
evalúa grant posibilidad productos puedan authentic inglés logros comprensión positivos wiggins motivación actitudes
complejos precisa conciencia niveles involucra educativo currículo edición agosto importancia tics temas reflexionar
artículo modelo memorístico aplicar relativamente materia elaborar prácticamente directa elección breve millán
múltipleaspectos antonio contenidos negativosprueba debemos copia inferior profundiza esenciales solicita justifique
generalmente ofrece sustente tiende pensamiento adivinar oportunidad asegurar completar diferencias resalta procesos
inconexos jerarquización subjetivas imprecisas complejas lectores permite especificar cuáles espera claramente facilita
expresión invitado obtenida incorpora meses rubric apoyo utilizarse usuarios criterios informar falso
tradicionalinstrumentosexamen prentan empezamos identificación relación integralfuentes ubicación negativos comparativa
curso establecidos previamente presentación reporte síntesis continuación escrito correspondencia demostraciones
respeto creación límites derechos autor blanca frías referencias tópico complemento toma aplicación herramientas
implementación resultar septiembre anunciamos costosa silvia hinojosa net pareonline digest getvn asp actualización
publicación 2&n eric case maría elsa kleen alternativas desarrollos méxico trillas editorial diseño participar especto
cubre negativosse desempeños construirse crecimiento refleja plazo oral documentos vacación experimentos proyectos
escritos investigaciones discusión revisión debates madurez situaciones adquiridas habilidades necesita dándose
curricular optimizar docente destaca superior comprometa significativos real vida naturaleza compleja alienta construya
informacióninstrumentosobservación final culminamos exploración requiere permitan cabalidad necesaria consecuencia
progreso alumno compromiso activo transformaciones precipitado estimula cambio cubrimiento paradigma partes mirada
expone regular presente estructurados dificultades evidenciar propósitos tecnologías comunicaciones exámenes dedicamos
hipólito totalmente entrevista gonzález práctica exclusivamente reducida visto conforman otorgar enfocan esperamos
solución publicaciones resultado llevado serie guiar modelos contribuido preparación éxito posibilidades enorme
advenimiento día acceso positivamente cabo publicamos aprovecha reconoce big ocasión profundizar desarrollo
tradicionalmente utilizado pasos competencia ediciones interés enero dedicado documento sirve manejo megatema
limitaciones tomando justificada reflexiva elaboración realizaciones publicaremos casos contrario convencional ocupa
quincenalmente completa reflejar procura actividades semanal consultor investigador forma selección redacción posibles
```

Representación de los tópicos

A partir de los datos normalizados y filtrados que conforma el corpus dimensionado, se procede a analizar usando LDA. Método que permitirá por un lado reducir la dimensionalidad del *corpus* para ser analizado en forma más eficiente; y por otro lado asignar a cada documento unos posibles *tópicos* con *términos* asociados con vectores probabilísticos y semánticos, que permitirán que cada *término* se asocie una probabilidad de pertenencia a otros

Esta etapa requiere una gran cantidad de pruebas de entrenamiento que implica añadir de nuevo palabras al stop *words*, eliminar colas de frecuencia, cambiar parámetros del LDA, etc.

Para este prototipo se establecieron para cada documento 5 *tópicos*, cada *tópico* con 25 *términos*, se realizaron múltiples pruebas con 6 a 9 *tópicos* entre 10 y 25 *términos*, pero debido a lo homogéneo de los contenidos la elaboración de mayor cantidad de *tópicos* se volvía redundante, se optó por obtener pocos *tópicos* y muchos *términos*.

En este caso se usó 0.75 como valor del hiper-parámetro Dirichlet para los Tópicos y un valor bajo para el parámetro a priori de Dirichlet para la distribución de los *tópicos* de 0.25. Esto es necesario ya que tenemos una distribución más amplia para la distribución de los *tópicos* por documento, pero para cada *tópico* una distribución más homogénea de los *términos*.

```
$lda = new Lda(
    new DataAsFeatures(), // a feature factory to transform the document
        data
    4, // the number of topics we want
    0.75, // the dirichlet prior assumed for the per document topic
        distribution
    0.25  // the dirichlet prior assumed for the per word topic distribution
);

$lda->train($tset,30);
```

También se usó un muestreo con 30 niveles de profundidad., en este modelo LDA se utiliza el *muestreo de Gibbs* para facilitar la distribución multinomial, normalizando la probabilidad (este método es parte del algoritmo NipTool). Cuando nos referimos a *Gibbs* establecemos que:

> "Método propuesto por Griffiths y Steyvers que toma muestras del posterior para aproximarlas con una distribución. Se considera uno de los métodos de Monte Carlo de cadenas de Markov (MCMC, Markov Chain Monte Carlo). Estos métodos construyen cadenas de markov con la distribución deseada a partir de una serie de muestras. " (Silvestre Gómez, 2018)

Después de ejecutar LDA para todos los documentos del *corpus* se realiza la representación de los documentos como secuencias de identificadores numéricos almacenadas en un array bidimensional. Para estos arrays, se representa un *documento*, asociada a la referencia de una instancia de una palabra, es decir, un token, representada por un identificador numérico de probabilidad de ocurrencia de este *término* respecto a otra serie de muestras

Documento 1

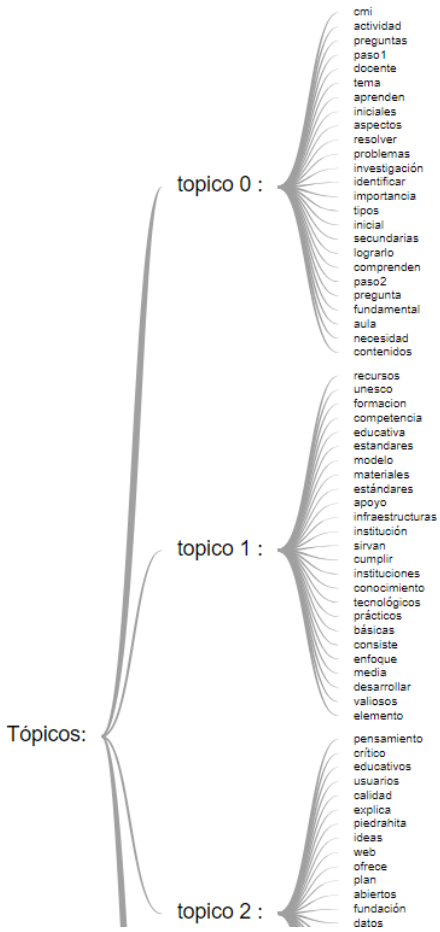
0 - informatica > 0.130434782609
0 - educación > 0.130434782609
0 - prácticas > 0.130434782609
0 - comunicaciones > 0.0217391304348
0 - regular > 0.0217391304348
0 - efectivo > 0.0217391304348
0 - comunicación > 0.0217391304348
0 - presentes > 0.0217391304348
0 - intregración > 0.0217391304348
0 - década > 0.0217391304348
0 - principales > 0.0217391304348
0 - preparación > 0.0217391304348
0 - preocupaciones > 0.0217391304348
0 - integración > 0.0217391304348
0 - enfoque > 0.0217391304348

1 - integrado > 0.130434782609

4 - comunicación > 0.00568475452196
4 - representaciones > 0.00568475452196
4 - básica > 0.00568475452196
4 - software > 0.00568475452196

5 - matemáticas > 0.0334065934066
5 - matematicas > 0.0312087912088
5 - artículos > 0.0202197802198
5 - aprendizaje > 0.0158241758242
5 - tecnología > 0.0114285714286
5 - publicas > 0.0114285714286
5 - proyecto > 0.00923076923077
5 - algebra > 0.00923076923077
5 - tema > 0.00923076923077
5 - nacional > 0.00923076923077
5 - formacion > 0.00923076923077
5 - comprensión > 0.00923076923077
5 - escolares > 0.00923076923077
5 - consejo > 0.00703296703297
5 - declaración > 0.00703296703297

Sobre el corpus filtrado se estableció un análisis de frecuencias de los *términos* extraídos de LDA con respecto a los *términos* genéricos se recopilaron 118.824 *términos* en los diferentes tópicos para cada *documento*, de los cuales se encontraron 56.658 términos únicos, cada *término* con su pertenencia a un clúster *tópico* y su probabilidad individual. Para cada *documento* hay 4 *tópicos* y 100 *términos*,



```
cmi actividad preguntas paso1 docente tema aprenden iniciales aspectos resolver problemas investigación identificar
importancia tipos inicial secundarias lograrlo comprenden paso2 pregunta fundamental aula necesidad contenidos recursos
unesco formacion competencia educativa estandares modelo materiales estándares apoyo infraestructuras institución sirvan
cumplir instituciones conocimiento tecnológicos prácticos básicas consiste enfoque media desarrollar valiosos elemento
pensamiento crítico educativos usuarios calidad explica piedrahita ideas web ofrece plan abiertos fundación datos fgpu
intelectuales web2 disposición diseño destrezas comunicación edades cursos búsqueda mitica actual colombia publicas
informacion temas desarrollo politicas matematicas nacional programa entrevistas matemáticas internacional incluye básica
educativas derechos integración déficit pisa universidad naturales icesi decenal cooperación
```

EL planteamiento primordial de LDA es que un se supone a priori que un *documento* puede tener diferentes *tópicos* unos más cercanos que otros, pero uno de ellos, es el más representativo probabilísticamente, a este *tópico* se le denomina *TopTopic*.

El *TopTopic* presenta un vector probabilístico mayor que los demás, para este caso se determina el *tópico* con mayor probabilidad de ese *documento* especifico

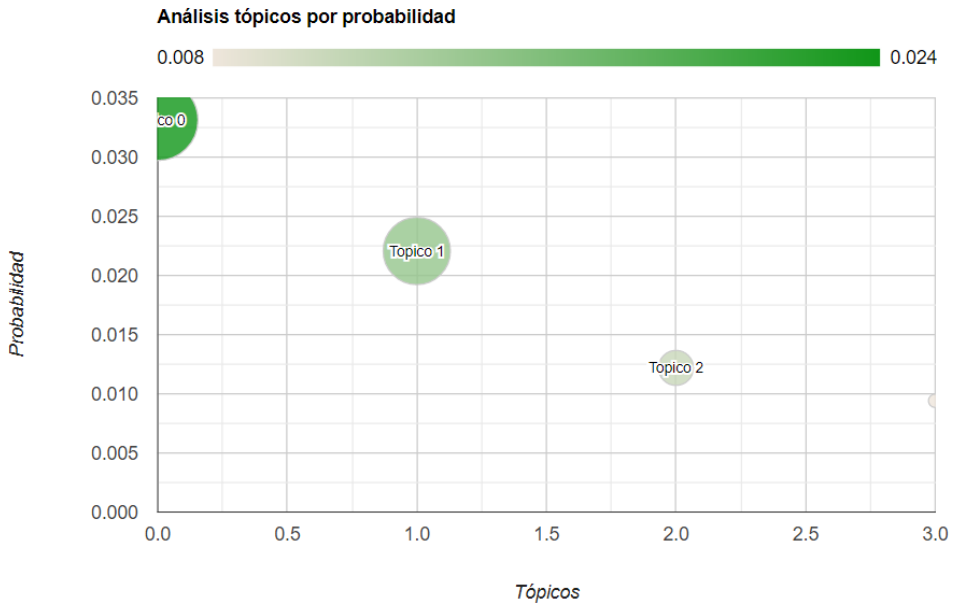


Ilustración 20 - Representaciónd el topTopic (Sánchez, B. 2019)

Terminos del mejor Tópico [#0 según LDA]

```
cmi, actividad, preguntas, paso1, docente, tema, aprenden, iniciales, aspectos, resolver, problemas,
investigación, identificar, importancia, tipos, inicial, secundarias, lograrlo, comprenden, paso2,
pregunta, fundamental, aula, necesidad, contenidos,
```

Extraigo los *TopTopics* para cada *documento*, con ello obtengo 29.706 *términos* repartidos en los 1.193 tópicos, lo cual es una enorme reducción con sentido de la dimensionalidad del corpus y con unos niveles probabilísticos aceptables.

Se extraen las densidades para cada *término* en cada *corpus* se almacena en la base de datos, para ellos se creó un script que permitiera esto

```
$result2 = mysqli_query($conexion,"SELECT terminoDoc FROM LDA_GenDoc ");
while ($rowAdemas2 = mysqli_fetch_array($result2, MYSQL_ASSOC)) {
    $terminosBag[] = utf8_encode($rowAdemas2[terminoDoc]);
}

// determino el número de términos en la tabla LDA
$terminos = $terminosBag;
$numTerminos = count($terminos);

// El array cuenta las repeticiones e incrementa la frecuencia,pone como clave
    el término
$cuantaTerminos = (array_count_values($terminos));

foreach ($cuantaTerminos as $key=>$val) {
    // Determino la densidad de este término para el corpus LDA
    $densidad = number_format(($val/$numTerminos)*100,6,'.','.');
    $terminoLDA = utf8_decode($key);
    echo "$key = $val. Densidad: ".$densidad."%<br/>\n";
    $query = "INSERT INTO terminosFrecuenciaLDA VALUES (NULL, '$terminoLDA',
        '$val','$densidad')";
    mysqli_query($conexion, $query);
}
```

Al comparar la densidad de *términos* entre las palabras obtenidas en el primer paso y las obtenidas a través del proceso LDA encontramos que se obtienen 1.593 *términos* únicos

Comparacion densidad entre términos sin filtro y LDA

	Término No filtro	d	Término LDA	d	Término TOP	d
1	forma	0.133	sociales	0.671	sociales	0.667
2	desarrollo	0.125	estándares	0.638	mundo	0.653
3	utilizar	0.12	desarrollo	0.637	educativos	0.613
4	recursos	0.12	mundo	0.636	estándares	0.599
5	mundo	0.119	enseñanza	0.621	desarrollo	0.592
6	crear	0.108	educativos	0.614	enseñanza	0.579
7	tipo	0.107	naturales	0.587	naturales	0.569
8	general	0.107	pensamiento	0.576	habilidades	0.525
9	proceso	0.105	cmi	0.545	pensamiento	0.522
10	problemas	0.103	habilidades	0.534	problemas	0.518
11	investigación	0.103	competencias	0.502	cmi	0.505
12	conocimiento	0.101	recursos	0.501	aula	0.502
13	tema	0.101	aula	0.5	recursos	0.481
14	personas	0.101	lenguaje	0.498	docente	0.475
15	habilidades	0.101	problemas	0.495	ofrece	0.471
16	desarrollar	0.099	educativas	0.478	competencia	0.468
17	importante	0.099	iste	0.471	web	0.468
18	permite	0.097	competencia	0.46	digitales	0.468
19	español	0.097	escolares	0.454	educativas	0.461
20	temas	0.097	ofrece	0.453	reseña	0.454

Ilustración 21 - Comparación de densidades entre términos (Sánchez, B. 2019)

En un gráfico de densidad para los tres estados en una distribución normal la cual hace referencia al *corpus* inicial después del filtrado de datos; El segundo estado en una distribución mediada por LDA, y el último una con la selección de los *términos* con mayor probabilidad para cada *documento*, podemos observar la clara variación de la densidad de los *términos claves*.

Ilustración 22 - Análisis de densidad de términos antes y después del LDA (Sánchez, B. 2019)

En general se puede apreciar como hay una distribución diferente de los *términos* en el *corpus*, en un primer momento es bastante homogéneo, pero al ejecutar el LDA adquieren mayor relevancia y otros incluso la pierden. Un ejemplo del de la palabra **desarrollo** la cual en un estado normal tiene un índice de densidad de 0.125, en el segundo estado esta adquiere relevancia adquiriendo respecto al corpus LDA un índice de densidad de 0.637 y se normaliza a 0.592 en el caso de los términos TOP.

Clasificadores

Filtrado por contenido:

En este punto, es factible el análisis entre los *tópicos* para cada *documento*, y como estos *tópicos* se relacionan con otros *documentos*. Para este modelo, estas relaciones estarán indexadas en base datos (se podrían hacer onLine, pero eso exige para cada transacción

del usuario un alto gasto de recursos de computo). Se han establecido scripts que permiten al cargar un nuevo documento hacer el análisis y la relación con otros documentos.

idA	topico	terminoDoc	prob
182	0	eficiente	0.00620347
182	1	lenguaje	0.0486936
182	1	enseñar	0.0106888
182	1	comunicación	0.0106888
182	1	computador	0.0106888
182	1	habilidades	0.0106888
182	1	investigaciones	0.0106888
182	1	procesador	0.0106888
182	1	ofrece	0.0106888
182	1	kathleen	0.00593824
182	1	teclado	0.00593824
182	1	claros	0.00593824
182	1	especial	0.00593824

idA	topico	sim
1057	9	0.074074
1057	22	0.074074
1057	40	0.035714
1057	31	0.035714
1057	41	0.035714
1058	40	0.029412
1058	41	0.029412
1058	13	0.029412
1058	12	0.029412
1058	38	0.029412

idA	terminoDoc	prob	frecuencia	densidad
1206	basado	0.00162866	10	0.033663
1206	exploran	0.00162866	22	0.074059
1207	exploran	0.00208768	22	0.074059
1207	subyacentes	0.00208768	2	0.006733
1207	conceptos	0.00208768	57	0.19188
1207	traducción	0.00208768	45	0.151485
1207	basado	0.00208768	10	0.033663
1205	elementos	0.00223714	69	0.232276
1205	rubrica	0.00223714	20	0.067326
1205	cierre	0.00223714	7	0.023564
1205	correos	0.00223714	37	0.124554
1205	observaciones	0.00223714	5	0.016832

Ilustración 23 - Tablas con los datos para el análisis de términos y tópicos (Sánchez, B. 2019)

Ya dimensionado el *corpus*, y determinado los *tópicos* y *términos* con mayor probabilidad y densidad representacional, se hace el análisis de los documentos. Tanto en la relación de los *tópicos* entre ellos como de los *documentos* en si mismos. El análisis entre *tópicos* está determinado por la revisión de la distribución de estos entre los documentos, siendo observable la cercanía y las connotaciones semánticas respecto a su posición vectorial

En el gráfico 24, se puede apreciar como hay 4 clúster formados por *tópicos* relacionados a varios *documentos*, en este caso muestra el *término* con mayor probabilidad de cada *tópico*,

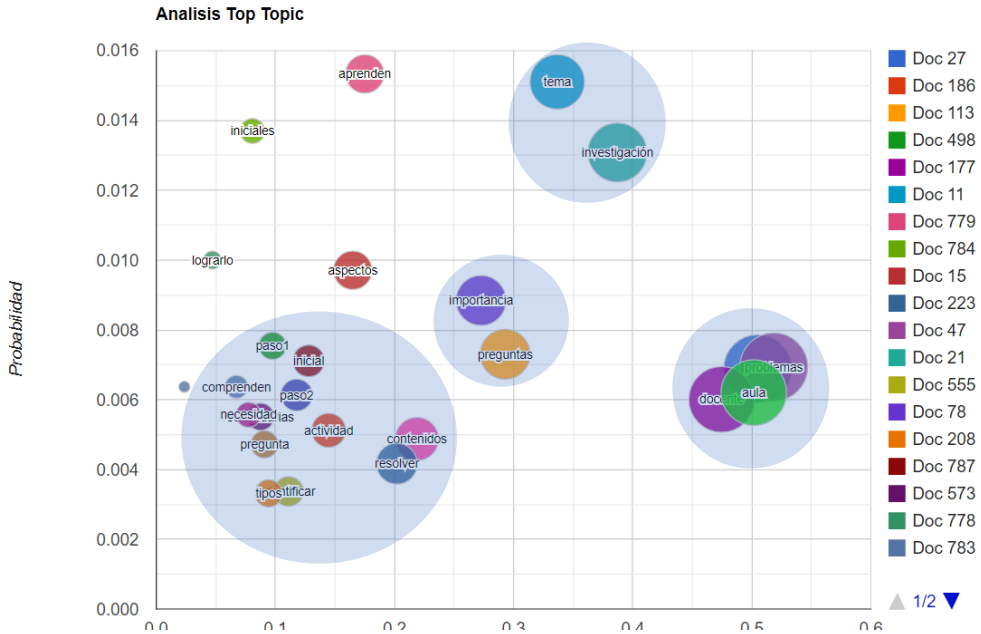


Ilustración 24 - distribución de los tópicos de un documento (Sánchez, B. 2019)

estos se conforman como grupos con características similares, para determinar la relación de similitud entre los *tópicos* requiere el uso de métodos de comparación para este caso se usará el método de *Jaccard*, el cual pondera el tamaño de la intersección de las representaciones vectoriales de dos *documentos* de texto,

Para este caso se analizó la similitud de los *tópicos* respecto a distribución de probabilidades de otros *tópicos* en otros *documentos*; y así extrayendo el id correspondiente al documento con características similares. Para ello,

1. se extrae el *tópico dominante,*
2. de este *tópico domínate* se extraen los *términos* con mayor índice de probabilidad a ese *tópico* particular,
3. con este *término* se seleccionan los *términos* similares de otros *tópicos* dominantes de otros documentos
4. se organizan primero los que tienen mayor probabilidad
5. Se extrae el ID de cada *documento*
6. se obtiene una matriz con los k-items pertenecientes al análisis topológico.

idA	terminoDoc	prob ▲ 1	frecuencia	densidad
887	contenidos	0.00483811	65	0.218811
887	necesidad	0.00483811	23	0.077425
887	aula	0.00483811	149	0.501582
887	fundamental	0.00483811	7	0.023564
887	tipos	0.00632676	28	0.094257
887	inicial	0.00632676	38	0.12792
887	secundarias	0.00632676	26	0.087524
887	lograrlo	0.00632676	14	0.047129
887	comprenden	0.00632676	20	0.067326
887	paso2	0.00632676	35	0.117821
887	pregunta	0.00632676	27	0.090891
887	importancia	0.00632676	81	0.272672
887	identificar	0.00781541	33	0.111089

```
$maxTopic = getBestProbTopicById($idA,$conexion);
$BestTopicTerm = getBestTerminosTopicById($idA,$maxTopic[0],$conexion);

for ($w=0;$w<count($BestTopicTerm);$w++){
    $terminob = utf8_decode($BestTopicTerm[$w]);

    // selecciono los 20 documentos con mayor probabilidad con esos mismos
        terminos
    $result2 = mysqli_query($conexion,"SELECT idA,terminoDoc,prob,frecuencia
        ,densidad FROM LDA_TopTopic WHERE terminoDoc LIKE '$terminob' GROUP BY
        terminoDoc ORDER BY prob DESC LIMIT 0,20");
    while ($rowAdemas2 = mysqli_fetch_array($result2, MYSQL_ASSOC)) {

        $idAz = $rowAdemas2[idA];  ;
        $selectLDAdoc[] = $idAz;
    }
}
```

Ilustración 25 - Codigo que ejemplifica el proceso de selección de items (Sánchez, B. 2019)

Filtrado por perfil documento:

Este análisis de contenido se hace usando el primer filtro LDA de *términos*, de esta manera el análisis no se hace sobre los 1.500 *términos* que conforman un *documento* original, sino sobre los 100 *términos* principales, facilitando así el uso de los recursos del sistema, para este caso se debe hacer la comparación de estos *términos* sobre cada uno de los

documentos perteneciente al corpus. Se usa el método de comparación de *Jaccard* y se obtiene un coeficiente de *similitud* entre cada documento a partir de los términos comunes que estos tengan. Este índice se guarda en base de datos para facilitar su consulta.

idRId ▲ 1	idA	idAx	sim
1	1	1	1
2	1	2	0.603063
3	1	3	0.639945
4	1	4	0.409062
5	1	5	0.612527
6	1	6	0.352332
7	1	7	0.657671
8	1	8	0.619411
9	1	9	0.62061
10	1	10	0.619521

En la ilustración se observa que existe un *id* de documento original *idA* y un *id* para el que se hace la comparación *idAx* y su correspondiente *índice de similitud*, entre más se acerque a 1 es similar (en este caso, cuando se compara respecto al mismo documento se obtiene 1)

Con estos índices la recomendación se puede dar en base a ese coeficiente, sin embargo, a pesar de poder tener elementos comunes respecto a términos claves, no necesariamente la correlación semántica es coherente. Hay palabras con significado ambiguo o genéricas que no aportan un sentido inferidle. Así pues, se organiza una matriz *k-ítem* usando las relaciones de esta tabla, obteniendo un segundo grupo de datos para analizar.

Otro tercer grupo de ítems k-*ítems* se pueden obtener a través de una mezcla entre los *tópicos* principales para cada documento y el análisis LDA de todos los *documentos* como conjunto de datos.

Previamente se había realizado un análisis general de todas los *Top-tópicos* (29.706 términos) obtenidos para todos los documentos usando LDA. Al hacerse este análisis se puede obtener clústeres genéricos de la distribución de estos *términos* predominantes y la relación de estos con un *documento* especifico (obteniendo así 50 *tópicos* con 10 *términos* cada uno).

topico ▲ 1	termino	prob
12	tipos	0.00947075
12	crítico	0.00947075
12	sociales	0.00947075
12	politicas	0.00947075
12	investigación	0.00947075
13	educativos	0.0199015
13	computadores	0.0149754
13	ofrece	0.0139901
13	básica	0.0130049

Estos *tópicos* pueden comparase como grupos y permiten validar algunos *términos* y su probabilidad de pertenecer a un *documento* especifico, permitiendo que se pueda obtener un tercer grupo de *k-items;* A este grupo pertenecen ítems que tienen términos comunes con alta probabilidad para cada *documento*.

En este caso a diferencia del primer grupo no se tiene en cuenta la relación tópico – *documento* sino la de *término-documento*. Se obtienen los id de cada *documento* con similitud semántica y se conforma el tercer *K-item*

Filtrado por perfilado del usuario

Este caso va a ser previsto para el prototipo pero no va a ser implementado, tal como se explicó en la metodología. Básicamente se analizará la relación usuario-documento por un lado y la relación del usuario con las recomendaciones Top-S sugeridas con el modelo.

Con ello se puede obtener un índice de frecuencia de los *documentos* más visitados usando Google analytics, se pueden obtener los datos de tiempo de visita, índice de rebotes, palabras usadas en la búsqueda, análisis de páginas de entrada y de salida, etc. Los cuales pueden convertirse en factores positivos y negativos a la hora de realizar las recomendaciones.

Resultados (ítems S)

Los resultados resultan del cómputo de los diferentes ítems obtenidos para ello

1. Se hace una ponderación de los ítems comunes a los 3 grupos y se le asigna un coeficiente de influencia,
2. Para cada ítem, se identifica el valor de probabilidad de cada documento a partir de la probabilidad de su *TopTopic*
3. A cada ítem, se suma el índice de probabilidad de sus 3 primeros términos LDA del conjunto
4. A cada ítem, se verifica su similitud por su distancia de coseno entre los documentos

Así se podría seleccionar en una lista los ítems con mayor valoración

C6: Recomendados intersección métodos

1. **Recursos para Robótica en Internet** [sim: 0.636476] > 645
2. **Investigación: Impacto de Scratch en el desarrollo del pensamiento algorítmico** [sim: 0.62023] > 2302
3. **Declaración de la NCTM sobre el uso de la tecnología** [sim: 0.618739] > 215
4. **Replantear la educación: ¿Hacia un bien común mundial?** [sim: 0.615303] > 2354
5. **Investigación: incidencia de Scratch en el desarrollo de Competencias Laborales** [sim: 0.601889] > 2178
6. **ISTE 2010: Nuevas Tendencias en Educación y TIC** [sim: 0.599066] > 1144
7. **La WebQuest y el uso de la Información en los Modelos de CMI** [sim: 0.594534] > 62
8. **Octavo Año de Eduteka en el ciberespacio** [sim: 0.592997] > 988
9. **Imágenes Digitales en la Clase de Historia** [sim: 0.592928] > 467
10. **Cuarto Año de Eduteka en el Ciberespacio** [sim: 0.591745] > 447
11. **Visiones 2020: Tecnologías y Aprendizaje** [sim: 0.590879] > 152
12. **Declaración de la NCTM sobre el uso de calculadoras** [sim: 0.590806] > 189
13. **Proyectos de Clase listos para utilizar en el aula** [sim: 0.580211] > 452
14. **EDUTEKA ganó el Premio Colombiano de Informática Educativa** [sim: 0.579947] > 258
15. **Plan Nacional colombiano de Tecnologías de la Información y las Comunicaciones y la Educación** [sim: 0.579659] > 878
16. **Competencias de Educación Digital** [sim: 0.578427] > 2436
17. **La propuesta de los Centros de Aprendizaje en la sociedad de la información** [sim: 0.577844] > 16
18. **Noveno Año de Eduteka en el Ciberespacio** [sim: 0.57706] > 1147
19. **Los Proyectos de Clase y su lista esencial de chequeo** [sim: 0.575566] > 2142
20. **Seminario: Recursos digitales mediados por GeoGebra** [sim: 0.575032] > 2380
21. **Definición de Diagramas de Flujo + Ejemplo** [sim: 0.489424] > 714
22. **Instrumentos Matemáticos Computacionales.** [sim: 0.461888] > 12
23. **Una mirada diferente a las Tecnologías en la Educación** [sim: 0.461741] > 22
24. **La Taxonomía de Bloom y el Pensamiento Crítico** [sim: 0.456211] > 109
25. **Aprendizaje visual, otro aporte de las TIC a la educación** [sim: 0.429589] > 77

Ilustración 26 - Ejemplo de salida de recomendaciones para un documento especifico (Sánchez, B. 2019)

Conclusiones.

Este trabajo ha presentado la arquitectura para la implementación de un sistema de recomendaciones no supervisado para un sitio web, basado en la producción de contenidos. La metodología que se ha propuesto está sustentada en teorías de modelado de datos y análisis inteligente de contenidos. Es altamente escalable, no está imitada solamente a un sistema de contenidos específicos, sino que puede ser entrenada para funcionar en cualquier ámbito que involucre análisis semántico de textos. Para ello, se ha realizado un prototipo altamente funcional elaborado para el sitio web online y disponible para todos los contenidos que fueron analizados (1.193 artículos).

El uso de herramientas de análisis de lenguaje es una de las tendencias actuales para afrontar la sobre carga de información, y no lleva poco más de una década de ardua investigación, con diferentes propuestas en la elaboración de sistemas y metodologías para el análisis autónomo de contenidos y contextos.

La definición e implementación de este modelo, ha sido una apuesta interesante para afrontar el desborde de información y la posibilidad autónoma de generar estrategias para entender la estructura que subyace en cada documento. Sobre todo, cuando se está atado a unas condiciones particulares que hacían difícil la obtención de información por parte del usuario; y a pesar de que no fueron implementadas en el prototipo, creo que el poder mencionar cuál podría ser su impacto, permite seguir trabajando en la posibilidad de integrarlas a la metodología planteada.

La implementación del prototipo permitió visualizar claramente las etapas necesarias en un proceso de análisis autónomo de contenidos. Conceptos probabilísticos avanzados se podían ejecutar en tiempo real y afectar directamente tanto el corpus, cómo el análisis de la estructura semántica de cada contenido; entender las relaciones toponímicas de los temas y términos que conforman un documento. Y de esta manera entender formas de relación entre distintos conceptos y métodos, que unidos pueden mejorar la implementación de un sistema autónomo de aprendizaje de la estructura semántica de un texto.

En la implementación del prototipo se evidencia la importancia de la fase de entrenamiento y prueba para dejar a punto el modelo, se requirió muchas horas de análisis de las variables para alcanzar un estado satisfactorio (el cual puede ser mejorado), de hecho, esto se ha conformado como una de las criticas a este tipo de implementaciones, puesto que involucra

un alto grado de esfuerzo y tiempo para ello. Uno de los aspectos claves es la configuración del tamaño de lo los tópicos y términos, los cuales influían notablemente en los resultados; el filtrado de datos en una etapa fundamental para reducir la dimensionalidad y eliminar términos vacíos en significado y probabilidad.

El trabajo permitió determinar un camino factible para la recomendación de contenidos, pero más allá, estos análisis semánticos pueden ser extrapolados para otros usos taxonómicos, tales como etiquetados autónomos, categorizaciones, análisis de otros tipos de contenidos que pueden ser clasificados y relacionados en esquemas no vistos previamente, etc.

Respecto a los objetivos planteados se pudieron alcanzar en un alto grado, sin embargo, la implementación de criterios adicionales a los contenidos para ser analizados e implementados, desde esta metodología de modelado de datos se convierte en una tarea matemáticamente compleja, la cual excede por mucho los objetivos de este TFM, requiere habilidades y conocimientos probabilísticos que sería incapaz de formular. Pero si se clarifica en qué instancia deben ser tomados en cuanta, cuáles posibles valores pueden ser asumidos y que fuentes pueden utilizarse para ello.

Bibliografía

Acosta, O., Aguilar, C., & Araya, F. (2018). QuarryMeaning : Una aplicación para el modelado de tópicos enfocado a documentos en español. *Procesamiento Del Lenguaje Natural*, *61*, 197–200. https://doi.org/10.26342/2018-61-31

Adomavicius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (2012). Context-Aware Recommender Systems. *Journal of Software*, *23*(1), 1–20. https://doi.org/10.3724/SP.J.1001.2012.04100

Araya, F. (2018). *Metodología para la construcción automática de un corpus de dominio específico*. Pontificia Universidad Católica de Chile.

Arias, J. (2017). *Evaluacón del uso de distintas métricas de distancia de texto en un algoritmo agregado para la imputación de valores faltantes mediante clasificación*. Instituto Tecnológico de Costa Rica.

Blei, D. M. (2012a). Probablistic Topic Model. *Communications of the Acm*, *55*, 77–84. https://doi.org/10.1109/MSP.2010.938079

Blei, D. M. (2012b). Surveying a suite of algorithms that offer a solution to managing large document archives. Probabilistic topic models. *Communications of the Acm*. https://doi.org/10.1145/2133806.2133826

Caro Martínez, M. (2017). *Sistemas de recomendación basados en técnicas de predicción de enlaces para jueces en línea*. Universidad Complutense de Madrid.

Colombo Mendoza, L. O. (2017). *Tecnologías para la recomendación semántica y filtrado colaborativo de contenidos y servicios* (Universidad de Murcia). Retrieved from https://www.tdx.cat/handle/10803/482136

Enio Walid Ghobar. (2017). *Un sistema de recomendación basado en perfiles generados por agrupamiento y asociaciones* (Universidad Politécnica de Valéncia). Retrieved from http://www.brainsins.com/es/

Ganegedara, T. (2018). Intuitive Guide to Latent Dirichlet Allocation. Retrieved July 16, 2019, from Towards Data Science website: https://towardsdatascience.com/light-on-math-

machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

Gorakala, S. K., & Usuelli, M. (2015). *Building a recommendation system with R : learn the art of building robust and powerful recommendation engines using R*. Packt Publishing.

Hammoe, L. (2018). *Detección de tópicos utilizando el Modelo LDA*. Instituto Tecnologico de Buenos Aires.

Herrera-Viedma, E., Porcel, C., & Hidalgo, L. (2004). Sistemas de recomendaciones: herramientas para el filtrado de información en Internet. *Hipertext*, *2*, 1–14. Retrieved from http://dialnet.unirioja.es/servlet/articulo?codigo=1098917&orden=32947&info=link

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*. https://doi.org/10.1126/science.aaa8415

Martín, V. E. (2016). *SISTEMAS DE RECOMENDACIÓN SEMÁNTICOS PARA LA COMPARTICIÓN DE CONOCIMIENTO Y LA EXPLOTACIÓN DE TESAUROS: Un enfoque práctico en el ámbito de los sistemas nutricionales*. Universidad de Granada.

Núñez Valdéz, E. R. (2012). *Sistemas de Recomendación de Contenidos para Libros Inteligentes* (Universidad de Oviedo). Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17295951

Princeton University. (2010). About WordNet. WordNet. *Princeton University.* https://doi.org/http://wordnet.princeton.edu/wordnet/

Silvestre Gómez, M. (2018). *Implementación de Asignación Jerárquica Latente de Dirichlet para Modelado de Temas*. Escuela Superior de Ingenieriía, Universidad de Sevilla.

Torres, M. C. (2017). *Text Analytics para Procesado Semántico*. Universidad de Vigo.

Valdiviezo-Diaz, P., & Hernando, A. (2016). Una Comprensiva Revisión de los Métodos de Recomendación basados en Técnicas Probabilísticas. *Iberian Conference on Information Systems and Technologies, CISTI*, *2016-July*, 604–610. https://doi.org/10.1109/CISTI.2016.7521413

Vera-del-Campo, J. (2012). *Contributions to security and privacy protection in recommendation systems*. (August). Retrieved from http://www.tdx.cat/handle/10803/113673