UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Rosario

Maestría en Ingeniería en Sistemas de Información

Tesis de Maestría

"Desarrollo de una medida de similaridad para Sistemas de Recomendación en sitios de Community Question Answering. Análisis desde un enfoque Big Data y usando un método de ensamble de clustering"

Ing. Federico Tesone

Director: Dr. Guillermo Leale

Co-director: Dra. Soledad Ayala

Rosario, Santa Fe, Argentina. Febrero de 2021

A mi persona favorita.

Resumen

Los *sistemas de recomendación* (Recommender Systems o RS) tienen la tarea de recomendar ítems a los usuarios de un sitio o aplicación. Los mismos pueden ser aplicados a sitios de preguntas y respuestas colaborativas, llamados *Community Question Answering* (CQA por sus siglas en inglés) y las preguntas que realizan los usuarios de la aplicación pueden considerarse como los ítems a recomendar. En este trabajo, son de interés las preguntas pendientes de ser respondidas, ya que la tarea de recomendar otras preguntas similares que hayan sido formuladas por otros usuarios y tengan la respuesta deseada, puede ser realizada por un RS, minimizando así el tiempo en que un usuario puede encontrar lo que estaba buscando.

Un buen RS debería utilizar una medida de *similaridad* confiable entre preguntas, por lo cual proponemos crear una nueva medida combinada de distancia para textos a través un método de ensamble de clustering basado en acumulación de evidencias, utilizando una arquitectura Big Data. Para este fin, dispondremos de un conjunto de datos de pares de preguntas reales, extraídos del sitio web Quora. Se realizará un análisis comparativo entre el método de ensamble de clustering y las medidas de similaridad utilizadas como punto de partida del mismo.

Este tipo de enfoque es necesario para trabajar con grandes conjuntos de datos y así recuperar, analizar y procesar los mismos con precisión, variabilidad y velocidad, con el propósito de encontrar una medida de similaridad que pueda presentarse como una alternativa a las actuales en términos de mejorar la experiencia del usuario en sitios de CQA, mejorar las medidas de rendimiento y reducir las probabilidades de error en la búsqueda de preguntas similares.

Palabras clave: Community Question Answering, Recommender Systems, Big Data, Ensemble Clustering, Evidence accumulation.

Índice General

Agradecimientos

Índice de Tablas

Índice de Figuras

Capitulo 1

1. Introducción

1.1. Área temática

Los sitios de *Community Question Answering* (CQA) brindan servicios que permiten a los usuarios formular y contestar preguntas sobre temas de cualquier índole. Miles de nuevas preguntas son subidas diariamente en sitios de CQA como Yahoo! Answers[1] , Stackexchange[2] , Stackoverflow[3] o Quora[4] . Estos son portales muy populares donde los usuarios suben diariamente una cantidad importante de preguntas de varios dominios para obtener respuestas de otros usuarios de la comunidad (Anuyah et al., 2017). Del análisis de sitios de CQA, puede observarse que muchas de las preguntas no están respondidas correctamente o no tienen respuestas específicas, ya que, en estas comunidades, hay típicamente un pequeño número de expertos entre la gran población de usuarios (Yang et al., 2013). Por lo tanto, cuando un usuario realiza una pregunta es de interés buscar si la misma ha sido formulada por otro usuario con anterioridad y que, además, esta pregunta tenga la respuesta buscada. Gracias a estos mecanismos, el usuario podría leer las respuestas de dicha pregunta sin tener que esperar a que la misma sea respondida. Esto no siempre es una tarea fácil, ya que podría darse la situación en la que esta pregunta exista previamente en el sitio y haya sido respondida, pero puede estar formulada de una manera completamente diferente en el sentido léxico. Por esta razón, una correspondencia exacta (o casi exacta) no es aplicable. Consideramos el siguiente ejemplo de dos preguntas iguales: *¿Cómo elijo una revista para publicar mi artículo?* y *¿Dónde publico mi*

[1] Yahoo! Answers: `https://answers.yahoo.com/`. Último acceso: Octubre 2020.
[2] Stackexchange: `https://stackexchange.com/`. Último acceso: Octubre 2020.
[3] Stackoverflow: `https://stackoverflow.com/`. Último acceso: Octubre 2020.
[4] Quora: `https://www.quora.com/`. Último acceso: Octubre 2020.

artículo?[5] . Entre estas dos frases, existe apenas una superposición de palabras, sin tener en cuenta *stopwords*[6] . Sin embargo, ambas preguntas tienen la misma respuesta, que referirá a revistas o sitios donde publicar un artículo científico. Con el fin de comparar dos preguntas, se establece una medida de similaridad que se puede intuir como máxima cuando son idénticas y que es inversamente proporcional a las diferencias entre ellas (Lin et al., 1998). Una medida de similaridad de texto entre preguntas basada en características léxicas no las detectaría como preguntas iguales. Esto deja en evidencia la necesidad de utilizar enfoques que además consideren características semánticas.

A partir del análisis anterior puede decirse que la tarea de encontrar preguntas similares en sitios de CQA puede ser llevada a cabo por un Sistema de Recomendación. Los Sistemas de Recomendación (Recommender Systems o RS) son herramientas de software y técnicas que proveen sugerencias de ítems que los usuarios pueden querer utilizar (Ricci et al., 2011). Las sugerencias relacionan varios procesos de toma de decisiones, como por ejemplo qué artículos comprar o qué música escuchar. El término general usado para denotar lo que los RS recomiendan a los usuarios es el de "*Ítem*". Los ítems son objetos que pueden estar caracterizados por su valor o utilidad. El valor de un ítem puede ser positivo si el ítem es útil para el usuario y negativo si no es apropiado y, en este último caso, el usuario tomaría una mala decisión al seleccionarlo. A partir de la dinámica que se construye en los RS, las recomendaciones al usuario pueden ser personalizadas o no personalizadas. Las primeras se basan en comportamientos del usuario o en grupos de usuarios para encontrar sugerencias adecuadas a sus preferencias; las segundas, por su parte, son inherentes a los ítems que el RS sugerirá. Cada una de estas estrategias de recomendación se elabora a partir de diferentes conocimientos y datos recopilados por el sitio o sistema donde el RS esté aplicado. Ejemplos de tales aplicaciones incluyen la recomendación de libros, películas o ítems de compra (Adomavicius y Tuzhilin, 2005). En particular, para los sitios de CQA, los algoritmos de recomendación se aplican principalmente a elementos de texto. Este trabajo se centrará en ese tipo de recomendaciones, que pueden estar clasificadas dentro de RS basados en contenido de texto no personaliza-

[5] Traducción de las preguntas "How do I choose a journal to publish my paper?, Where do I publish my paper?" extraídas desde el conjunto de datos de Quora que se utilizará en el presente trabajo de tesis `https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs`. Último acceso: Agosto 2018.

[6] En informática, se llama stopword a palabras que se filtran antes o después del procesamiento de datos del lenguaje natural (Leskovec et al., 2014).

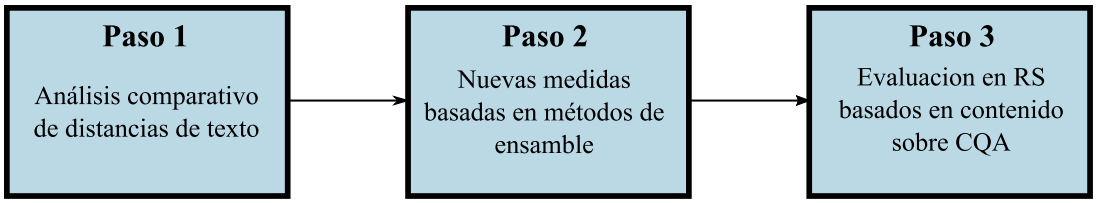

Figura 1: Pipeline para un RS basado en contenido de CQA y en una nueva medida de similaridad.

dos, ya que las mismas están basadas únicamente en la estructura sintáctica y semántica de las preguntas existentes en los sitios de CQA. Los usuarios pueden navegar esas recomendaciones. Luego, pueden aceptarlas o no y, además, proveer inmediatamente o en un paso posterior, una retroalimentación explícita o implícita.

A consecuencia de lo expuesto anteriormente, y sumado a que se dispone de un gran conjunto de datos, se propondrá una arquitectura para utilizar Big Data con el fin de crear una medida de similaridad de texto que alimente a un RS especializado en la tarea de encontrar preguntas similares en sitios de CQA basado en análisis de contenido de texto. Este tipo de enfoque es necesario para procesar una gran cantidad de datos y, de esta manera, optimizar el procesamiento de los mismos, logrando velocidad y con la ventaja de poder aprovechar toda la variabilidad que provee un conjunto de datos de gran volumen. Luego de obtener esta nueva medida de similaridad, se realizará un análisis comparativo de la misma contra las medidas subyacentes utilizadas como entrada del método en cuestión.

1.2. Tema específico

Con el fin de dejar en claro el alcance de este trabajo, se toma como punto de partida el trabajo de investigación de la Universidad Tecnológica Nacional, Facultad Regional Rosario: “Comparative Analysis on Text Distance Measures Applied to Community Question Answering Data”, el cual se centra en el Paso 1 del pipeline que se describe en la Figura 1.

Este proceso descrito en tres pasos, tiene como objetivo construir un RS basado en una medida novedosa de similaridad de texto. En el Paso 1 se realiza un análisis comparativo desarrollado a partir de medidas basadas en distancia, obtenidas de análisis de texto, con el fin de evaluar RS a partir de grandes conjuntos de datos; en el Paso 2, a partir de los resultados arrojados por el Paso 1, se crea una nueva medida construyendo una matriz de similaridad basada en

análisis de clustering, como una de las propuestas en los métodos *Algoritmo de Particionamiento de Similitud basado en Cluster* (Cluster-based Similarity Partitioning Algorithm o CSPA) (Strehl y Chosh, 2002) y *Clustering de Acumulación de Evidencias* (Evidence Accumulation Clustering o EAC) (Fred y Jain, 2005). El método EAC intenta mejorar la calidad de salida para una representación de similaridad basada en texto; por último, en el Paso 3 se debe aplicar la matriz de distancias obtenida en el Paso 2 en un RS basado en contenido, con el fin de evaluar su eficacia en sitios de CQA.

Para tomar dimensión del volumen de datos que es necesario manejar con este enfoque basado en clustering, si, por ejemplo, tomáramos el conjunto de datos Quora (404301 pares de preguntas, es decir, 808602 preguntas totales), y quisiéramos generar una sola matriz de distancias cruzando absolutamente todas las preguntas entre sí, estaríamos calculando $\frac{n(n+1)}{2} = 326919001503$ distancias, donde $n = 808602$ y el resultado es la cantidad de elementos en la triangular superior. Esta matriz considera sólo una de las distancias de similaridad de texto del estado del arte, por lo cual si deseamos combinar varias medidas mediante un método de ensamble, deberíamos generar al menos una matriz por cada distancia del estado del arte, y luego usar las mismas para aplicar EAC, por lo cual estaríamos generando un número total de cálculos considerablemente mayor al que puede procesar una computadora clásica. Es necesario además tener en cuenta que las distancias pueden llegar a considerar características diversas entre sí, como morfología, sintaxis y semántica de los textos, lo cual añade complejidad y variedad al volumen considerado. Esto conlleva considerar una arquitectura Big Data, optimización de código y técnicas de ejecución paralela entre gran número de núcleos de procesamiento. Con respecto al espacio de almacenamiento, debe notarse que las matrices requieren doble precisión para la representación interna de cada uno de sus elementos (distancias), es decir 750 KB cuando están almacenados en un archivo, por lo cual, si consideramos la matriz ejemplificada anteriormente, necesitaríamos aproximadamente 12 TB para almacenarla, por lo cual, es necesario un esquema de almacenamiento optimizado para la implementación de este método. Con respecto a la velocidad de procesamiento, pruebas preliminares en un procesador potente brindan una estimación de alrededor de 3 años para completar el procesamiento. Con el problema planteado de esta forma, es indispensable aplicar un enfoque de Big Data para satisfacer los requerimientos de volumen, variedad y velocidad que requiere el contexto de análisis, de tal

forma de brindar resultados veraces, y de esa forma cumplir con las premisas de las "V" del Big Data[7] .

Este trabajo de tesis, apuntará entonces a construir una medida de similaridad novedosa desde un enfoque Big Data, tal como se describe en el Paso 2 del pipeline, por lo cual es necesario crear un nuevo software basado en una arquitectura y patrones de Big Data, tomando como punto de partida el desarrollo del estado de arte.

1.3. Objetivo general

El presente trabajo de investigación tiene como objetivo construir una arquitectura Big Data que se aplique a grandes conjuntos de datos de preguntas de CQA y permita encontrar nuevas medidas de similaridad entre textos que puedan ser utilizadas en sistemas de recomendación.

1.4. Objetivos específicos

Se detallan a continuación, los objetivos específicos que son necesarios para lograr el objetivo principal.

1. Identificar medidas de similaridad de texto existentes y un método efectivo de aplicación de las mismas en grandes volúmenes de datos.

2. Diseñar y desarrollar una arquitectura Big Data para cálculo de similaridad en grandes matrices, que requerirá nuevas estrategias para recolectar, procesar y manejar grandes volúmenes de datos.

3. Proponer una nueva medida que permita integrar las medidas de similaridad del estado del arte mediante una arquitectura de software basada en Big Data y sea extensible a otras medidas

4. Brindar conclusiones, pautas y recomendaciones para trabajar con medidas de comparación de textos en grandes volúmenes de datos en sitios de CQA utilizando arquitecturas basadas en Big Data.

[7] Las "V" del Big Data refieren a Volumen, Variedad y Velocidad. También se consideran los conceptos de Valor y Veracidad con respecto al resultado de la aplicación del enfoque Big Data (Gandomi y Haider, 2015).

Capitulo 2

2. Fundamentación

2.1. Motivación de la tesis

La calidad de un RS tiene una relación directa con los datos de entrada que se han generado para alimentarlo. Con el fin de generar una entrada basada en medidas de similaridad, es necesaria la comparación de preguntas formuladas en sitios de CQA usando técnicas de análisis de texto. Un problema importante inherente al análisis de texto, con el fin de cuantificar relaciones entre distintos fragmentos o documentos, es encontrar la medida apropiada de representación (González y otros, 2017). Algunas medidas de similaridad resultantes de algoritmos de recomendación en análisis de texto, son obtenidas mediante algoritmos puramente sintácticos, léxicos, tales como: Term Frequency (Salton y McGill, 1983), Term Frequency/Inverse Document Frequency (Baeza-Yates et al., 1999), basados en ventanas como FastText (Joulin et al., 2016) o Word2Vec (Mikolov et al., 2013), o semánticos, como Semantic Distance (Li et al., 2006). Los algoritmos puramente sintácticos como Term Frequency y Term Frequency/Inverse Document Frequency tienen conocidos problemas, tales como ser invariantes respecto al orden de las palabras o ser sensibles a stopwords, por lo cual, necesitan un gran trabajo de pre-procesamiento. FastText y Word2Vec están fuertemente afectados en el orden en el cual aparecen las palabras. Adicionalmente, ninguna de estas técnicas tiene en cuenta la semántica de las palabras y sus relaciones, como si lo hace Semantic Distance. Sin embargo, esta última técnica, según el trabajo tomado como estado del arte, tampoco alcanza medidas de rendimiento apropiadas para un RS en un sitio de CQA, como por ejemplo: buen desempeño debido a su complejidad inherente.

Resultados experimentales de medidas de rendimiento obtenidas en el trabajo que se toma como punto de partida de esta tesis, arrojan entre un 66 % y un 68 % de precisión y entre un 32 % y un 33.5 % de error usando cada uno de los algoritmos de recomendación descritos anteriormente. Estos valores son

considerados prometedores, ya que las medidas de rendimiento son consistentes en todos los algoritmos seleccionados, lo que denota que la complejidad inherente del conjunto de datos no afecta significativamente la performance de cada uno de ellos. Además, los resultados de prueba no varían significativamente con respecto a los resultados de validación. Dicho esto, la motivación de este trabajo de tesis, así como el de las futuras líneas de investigación, es la creación de un método novedoso que combine medidas de similaridad existentes que pueda aplicarse como entrada para un RS que pueda ser implementado en sitios de CQA. Adicionalmente, se propondrá una arquitectura de software que soporte el procesamiento del método propuesto de una forma eficiente y escalable. Para tal fin, se crearán matrices de distancias (o similaridad), usando cada una de las preguntas del conjunto de datos en estudio, para luego combinarlas usando métodos de ensamble de clustering, ya que, como existen cientos algoritmos de clustering, es difícil identificar un solo algoritmo que pueda manejar todos los tipos de forma y tamaños de cluster, e incluso, decidir qué algoritmo sería el mejor para un conjunto de datos en particular. Fred y Jain (2005) introducen el concepto de clustering de acumulación de evidencias, que mapea las particiones de datos individuales en un ensamble de clustering dentro de una nueva medida de similaridad entre patrones, sumarizando la estructura entre-patrón percibido de esos clusters. La partición de datos final es obtenida aplicando el método *single-linkage* a la nueva matriz de similaridad. El resultado de este método muestra que, la combinación de algoritmos de clustering "débiles" como el *k-means*, pueden conducir a la identificación de clusters subyacentes verdaderos con formas, tamaños y densidades arbitrarias. Por lo cual, teniendo en cuenta diferentes particiones creadas con el método de ensamble desde los mismos datos originales, objetos de textos similares probablemente pertenecerán al mismo cluster.

El desarrollo de matrices de similaridad para la aplicación del EAC que se utilizarán como entrada de RS, claramente implica manipular un gran volumen de datos complejos y realizar un elevado número de cálculos en tiempo real, ya que nos estamos refiriendo a conjuntos de datos cuyo tamaño supera la capacidad de las herramientas tradicionales de bases de datos de recopilar, almacenar, gestionar y analizar la información (De Battista et al., 2016). Esto implica, en principio, considerar una *matriz de co-asociación* entre elementos realizando varias series de corridas y aplicación de clustering. Cada una de esas series está basada en una de las medidas de similaridad. El resultado será un un valor adimensional e

insesgado que puede mejorar la representación para la estructura subyacente de relaciones de texto. El volumen de datos ejemplificado en las secciones anteriores, deja expuesta la necesidad de investigar y desarrollar el tema aquí propuesto con un enfoque distinto al tradicional. Esto implica realizar un muestreo aleatorio de pares de preguntas dentro de una arquitectura que permita generar la mayor cantidad posible de subconjuntos de datos extraídos aleatoriamente. Además, posibilitará que cada uno de ellos sea lo más grande posible para aprovechar toda la *variedad* de los datos. Mientras más se aproveche la variedad de los datos (más subconjuntos de datos y de mayor tamaño), más afectará negativamente en el tiempo de de procesamiento, razones por las cuales se hace necesaria una arquitectura e infraestructura preparada para tal desafío, con una velocidad que haga posible obtener resultados en un período de tiempo razonablemente corto. Un enfoque Big Data es imprescindible para este tipo de procesamiento de datos. No solo se desea hacer referencia a la gran cantidad y complejidad de los datos, sino también a las herramientas utilizadas para procesarlos y las posibilidades de extraer conocimiento útil a partir del análisis de los mismos. Estos procesos y herramientas son el eje central de la definición de Big Data de la consultora Gartner (2012), la cual hace foco en los procesos para manipular activos de gran volumen y variedad con una gran velocidad. Por lo cual, si bien Big Data se refiere a estos activos, demanda formas innovadoras y efectivas de procesarlos, que habiliten tomas de decisiones y automatización de procesos.

Por todos estos motivos, se propone la elaboración de un nuevo método y una arquitectura que lo soporte, que genere una entrada de datos correctamente estructurada para RS y que pueda ser utilizada en sitios de CQA, de una forma eficiente y eficaz.

2.2. Importancia científico-tecnológica

Con respecto a los sitios de CQA en particular, la importancia de este trabajo radica tanto en la posibilidad de construir un RS que, desde el punto de vista del usuario, reduzca tanto el tiempo promedio en que se encuentra una respuesta como, a su vez, mejore la experiencia del sitio. En este sentido, en la mayoría de los casos no será necesario escribir múltiples versiones de la misma pregunta y los lectores podrán encontrar rápidamente la respuesta que están buscando. Por otro lado, se evitará que se creen preguntas duplicadas, lo que significaría un aumento considerable en la calidad y cantidad de la base de conocimiento del

sitio, construyendo una relación biunívoca entre una pregunta y su correspondiente respuesta. Además, se logrará optimizar el tamaño de la base de datos, la integridad de la información, mejorar la velocidad en búsquedas e incrementar de la satisfacción y fidelidad del usuario (Ricci et al., 2011).

Por último, el resultado de la presente investigación también puede ser utilizado para sitios que son fuente de consulta para diversos investigadores dentro del ámbito de la Universidad Tecnológica Nacional, Facultad Regional Rosario, tales como bibliotecas virtuales o foros de consulta para investigaciones científicos-tecnológicos que incluyan I+D+i. Esto permitiría no solo conocer los intereses de otros investigadores y en qué términos formularon sus interrogaciones, sino también conocer quién o quiénes elaboraron las respuestas a dichas preguntas y a qué campo disciplinar pertenecen.

2.3. Formación de recursos humanos

El presente trabajo de tesis, en relación a la formación de recursos humanos, tiene los siguientes objetivos:

- Capacitar a un grupo de estudiantes de la UTN FRRo, con elementos para la investigación y desarrollo en aplicaciones Big Data.
- Realizar grupalmente conocimiento científico, con base teórica sustentable y ejemplos empíricos de aplicaciones funcionales, para presentar en congresos tales como AGRANDA[8] , CONAIISI[9] , o RecSys[10] ; o eventos relacionados con Ingeniería en Sistemas de Información.
- Elaborar material de estudio relacionado con la temática de la minería de datos para materias de grado y/o posgrado.
- Lograr que los estudiantes puedan entender cómo está formado en la actualidad el estado del arte sobre el presente tema y que esto sirva de base para futuras investigaciones en UTN FRRo, ya sean proyectos de investigación, tesis de maestría o de doctorado.
- Desarrollar insumos para el armado de cursos tanto de formación académica como abiertos a la comunidad relacionados con Big Data o análisis de texto.

[8] AGRANDA: Simposio Argentino de GRANdes DAtos.
[9] CONAIISI: Congreso Nacional de Ingeniería Informática - Sistemas de Información.
[10] RecSys: The ACM Conference Series on Recommender Systems.

2.4. Importancia socio-económica

El tema posee una importancia social y económica que permitiría construir contactos y alianzas -económicas, académicas y de naturaleza mixta- con instituciones extranjeras. En otras palabras, a nivel social podría utilizarse para actividades de investigación y en los diferentes niveles educativos, según se adecúen las explicaciones y el vocabulario utilizado, las actividades y los diversos usos. Buscar información en bibliotecas digitales y virtuales, en bases de datos científicos, repositorios digitales, foros especializados de temáticas específicas y diversas o plataformas educativas, son algunos de los sitios donde los RS pueden ser utilizados y aplicados para determinadas actividades cognitivas. Además, el tema puede ser complementado en un futuro con otras líneas de investigación, tales como políticas educativas para la alfabetización mediática, análisis y datos online, fuentes abiertas o la relación entre tecnología y democracia. Estas líneas, de prioridad en la agenda de ciencia y tecnología de países del primer mundo, están siendo desarrolladas entre academia, instituciones de gubernamentales y policy-makers, de manera interdisciplinaria y con el objetivo de mejorar las herramientas que poseen los ciudadanos en relación a la cultura digital y sus mecanismos de funcionamiento estrictamente técnicos y los aspectos culturales que la atraviesan. Por otro lado, en el marco económico, los resultados de la presente investigación posibilitarán continuar con futuras indagaciones referidas a la temática y diseñar/construir nuevas herramientas de software adecuadas en función de ciertos usos y usuarios específicos. Estas acciones permitirían llevar adelante: nuevos proyectos de investigación interdisciplinarios y con subsidios de naturaleza mixta (público-privada), formación de formadores, pequeños emprendimientos para estudiantes avanzados y/o la postulación a becas de formación (nacionales e internacionales).

Capitulo 3

3. Marco teórico

En este capítulo, se explicarán los conceptos utilizados en el método propuesto que se desarrollará en el presente trabajo de tesis, estructurados en cinco grandes aristas: sitios de CQA, Sistemas de Recomendación, Big Data y medidas de similaridad y clustering. Todos estos conceptos se combinarán para luego, en el siguiente capítulo, desarrollar un método que satisfaga los objetivos del trabajo de tesis, de una manera superadora.

3.1. Sitios de CQA

Los servicios de Community Question Answering CQA, son un tipo especial de servicios *Question Answering* (QA), los cuales permiten a los usuarios registrados responder a preguntas formuladas por otras personas. Los mismos atrajeron a un número creciente de usuarios en los últimos años (Li y King, 2010). Una pregunta formulada en Quora, y respondida por su fundador y CEO, Adam DÁngelo, revela que el sitio recibe más de 200 millones de visitantes únicos mensualmente (información actualizada a Junio de 2017), lo que denota la popularidad de este tipo de portales[11] . Desde la creación de este tipo de servicios, se han aplicado diferentes técnicas de software para que los usuarios encuentren respuestas a sus preguntas en el menor tiempo posible y aprovechar al máximo el valor de las bases de conocimiento, por ejemplo, un framework para predecir la calidad de las respuestas con características no textuales (Jeon et al., 2006), incorporar información de legibilidad en el proceso de recomendación (Anuyah et al., 2017), encontrar a los expertos apropiados (Li y King, 2010) o recomendar la mejor respuesta a una pregunta dada, entre otros. Sin embargo, el mecanismo existente en el cual se responden las preguntas en los sitios de CQA todavía no alcanza a satisfacer las expectativas de los usuarios por varias razones: (1) baja probabilidad de encontrar al experto: una nueva pregunta, en muchos casos,

[11] Pregunta formulada en el sitio Quora "How many people use Quora?": `https://www.quora.com/How-many-people-use-Quora-3`. Último acceso: Febrero 2021.

puede no encontrar a la persona con la habilidad de responder de manera correcta, resultando en respuestas tardías y que distan de ser óptimas; (2) respuestas de baja calidad: los sitios de CQA suelen contener respuestas de baja calidad, maliciosas y spam. Estas suelen recibir baja calificación de los miembros de la comunidad; (3) preguntas archivadas y poco consultadas: muchas preguntas de los usuarios son similares. Antes de formular una pregunta, un usuario podría beneficiarse de buscar ya formuladas, y por consiguiente, sus respuestas (Yang et al., 2013).

3.2. Sistemas de recomendación

3.2.1. Contexto Histórico

Es muy frecuente tener que tomar decisiones sin la suficiente experiencia personal sobre las alternativas disponibles. En la vida cotidiana, confiamos en recomendaciones de otras personas ya sea de boca en boca o cartas de recomendación, reseñas de libros y películas o encuestas generales. Los sistemas de recomendación asisten este proceso natural en el ámbito de los sistemas de información (Resnick y Varian, 1997). El primer RS, Tapestry (Goldberg et al., 1992), fue un sistema experimental de correo electrónico destinado a resolver el problema de manejar grandes cantidades de emails filtrando según cuán interesantes son los documentos, utilizando un enfoque basado en el contenido de los mismos y también filtros colaborativos, lo que después se denominaría RS no personalizados y personalizados por Ricci, Rokach y Shapira en el año 2011. Se ha trabajado mucho en mejorar y desarrollar nuevos enfoques con respecto a RS en los últimos años, y el interés en esta área sigue vigente debido a la abundancia de aplicaciones prácticas en las cuales es necesario ayudar a los usuarios a lidiar con la sobrecarga de información[12] y proveer recomendaciones personalizadas, contenidos y servicios. Sin embargo, a pesar de todos estos avances, la generación actual de RS todavía requiere mejoras para que los métodos de recomendación sean más efectivos y aplicables a una gama más amplia de sistemas y/o sitios. Aunque las raíces de los RS se remontan a trabajos en ciencia cognitiva (Rich, 1979), teoría de aproximación (Powell, 1981), recuperación de información (Salton, 1989), ciencias de las predicciones (Armstrong, 2001), ciencias de la gestión

[12] El concepto de sobrecarga de información, del inglés information overload, hace referencia a cuando los usuarios reciben demasiada información, por lo cual, la precisión en sus decisiones empieza a decrecer (Eppler y Mengis, 2004).

(Murthi y Sarkar, 2003) y también al modelado de la elección de consumidor en marketing (Lilien et al., 1992), los RS recién surgen como un área de investigación independiente en la década de 1990, cuando los investigadores comenzaron a centrarse en problemas de recomendación que se basan específicamente en *calificaciones* (Adomavicius y Tuzhilin, 2005). En su formulación más común, el problema de recomendación se reduce a estimar calificaciones para los ítems que no han sido vistos por un usuario.

3.2.2. Funciones de un Sistema de Recomendación

Como se mencionó anteriormente, un RS es un conjunto de herramientas de software que sugiere items a un usuario, que posiblemente utilizará. Haciendo énfasis particularmente en RS comercial, probablemente la función más importante es incrementar el número de ítems vendidos, lo cual es posible porque el RS ofrecerá los ítems sobre los cuales el usuario tiene más probabilidades de querer o necesitar. Esto implica, aumentar el ratio de conversión, es decir, la cantidad de ventas que es posible efectuar sobre un ítem sobre del total de oportunidades que un usuario selecciona el mismo. Por ejemplo, en un sitio de delivery de comida online, cuantas veces un usuario realiza un pedido en un restaurante en particular, sobre el total de veces que observó el menú del mismo. Indudablemente, la conversión va a ser mayor si el usuario recibe recomendaciones de restaurantes que están más cercanos a su gusto personal. Otra función de un RS comercial muy relacionada a la anterior es vender productos más diversos, ya que sería muy difícil para un usuario encontrarlos sin una recomendación precisa.

Desde el punto de vista del usuario, un conjunto de recomendaciones precisas y relevantes aumentarán su satisfacción y fidelidad. El usuario disfrutará usar un sistema donde cada ítem o característica que utiliza está diseñada teniendo en cuenta sus intereses. Aumentar la fidelidad del usuario con el sitio, significa que habrá mucha más interacción y, por lo tanto, el modelo de recomendación se volverá más refinado; pudiendo utilizar este conocimiento para mejorar otros sistemas relacionados, como control de stock o publicidad (Ricci et al., 2011).

3.2.3. Técnicas de Recomendación

Para implementar su función principal, un RS debe *predecir* si vale la pena recomendar un ítem en particular. Para esto, este sistema debe ser capaz de predecir la utilidad de algunos de los ítems, o al menos poder comparar la utilizad

entre algunos de ellos, y entonces decidir qué ítems recomendar basándose en esta comparación. Para realizar estas comparaciones, los RS basan sus estrategias de recomendaciones en 6 técnicas básicas (Ricci et al., 2011):

3.2.3.1. Basados en contenido

Los RS basados en contenido intentan recomendar ítems similares a los que el usuario eligió anteriormente. Como su nombre lo indica, el proceso básico llevado a cabo por estos RS consiste en hacer coincidir atributos del perfil de usuario que posean preferencias e intereses en la búsqueda actual, con los atributos del ítem que se va a recomendar (Lops et al., 2011). Este tipo de RSs es especialmente útil cuando se conocen características de los ítems a recomendar pero no se conocen características del usuario, en otras palabras, estos sistemas intentan recomendar ítems similares a los que el usuario ha elegido anteriormente.

Uno de los limitantes conocidos de estos RS es que son limitados a recomendar ítems del mismo tipo al que el usuario está solicitando. Por ejemplo, no sería posible recomendar música, utilizando videos ya que el perfil de contenido es distinto. Para solucionar esto, muchos de los RS basados en contenido están utilizando algoritmos híbridos con otro tipo de técnicas de recomendación.

3.2.3.2. Filtrado Colaborativo

El *Filtrado Colaborativo* (Collaborative Filtering en inglés o CF) es el proceso de filtrado o evaluación de ítems usando las opiniones de los demás (Schafer et al., 2007). El Filtrado Colaborativo es el enfoque original y el más simple de todas las técnicas de recomendación, y el más utilizado. Se basa en recomendar al usuario activo, los ítems que otros usuarios con gustos similares eligieron en el pasado.

3.2.3.3. Demográficos

La mayoría de los RS utilizan enfoques basados en conocimiento o en contenido, esto implica que se necesita la suficiente información o un conocimiento adicional para poder llevar a cabo las recomendaciones. Los RS demográficos hacen recomendaciones basadas en clases demográficas, la ventaja es que la información histórica no es necesaria. Por ejemplo, una aplicación podría ser utilizar información demográfica para predecir el rating de distintos turistas a atracciones, basándose en enfoques predictivos de Machine Learning (Wang et al., 2012).

Las técnicas demográficas forman correlaciones "personas-a-personas", como los sistemas colaborativos, pero utilizando distinta naturaleza de los datos, en este caso, el perfil demográfico del usuario.

3.2.3.4. Basados en conocimiento

Los RS *basados en conocimiento* (Knowledge-based en inglés), usan el conocimiento acerca de los usuarios e ítems a recomendar para generar la recomendación, razonando acerca de que items satisfacen los requerimientos del usuario (Burke, 2000).

Los RS de filtrado colaborativo, al utilizar datos de otros usuarios, deben ser inicializados con un conjunto de datos considerablemente grande, ya que un sistema con una base de datos pequeña es improbable que sea útil. Además, la precisión del sistema es muy sensible al número de ítems asociados con un usuario dado (Shardanand y Maes, 1995). Esto conlleva a un problema de inicialización: hasta que no exista un número considerable de usuarios cuyas elecciones y hábitos sean conocidos, el sistema no será útil para un nuevo usuario. Lo mismo sucede para los RS que toman enfoques de Machine Learning. Típicamente, este tipo de sistemas se convierten en buenos clasificadores una vez que han aprendido desde una gran base de datos. Los RSs basados en conocimientos evitan estas desventajas. No existe un problema de inicialización, ya que las recomendaciones no dependen de un conjunto de datos grande. Para estos RSs no es necesario recolectar información acerca de un usuario en particular porque las recomendaciones que realizan son exclusivamente basadas en las elecciones de un usuario en particular. Estas características no solo hacen a este tipo de RS muy valioso en sí mismo, sino también como complemento de otros RS que utilicen distintas técnicas.

3.2.3.5. Basados en comunidades

Un *RS basado en comunidades* (community-based en inglés) hace uso del método "boca a boca" digital para construir una comunidad de individuos que comparten opiniones personales y experiencias relacionadas con sus recomendaciones de ítems. Estos sistemas, presentan y agregan opiniones generadas por los usuarios en un formato organizado, las cuales son consultadas a la hora de tomar decisiones (por ejemplo, comprar un producto) (Chen et al., 2009). La evidencia sugiere que las personas están más inclinadas para seguir una sugerencia de sus

amigos que una sugerencia similar que viene desde una persona anónima (Sinha et al., 2001). Este tipo de RS toma importancia cuando se tiene en cuenta la creciente popularidad de las redes sociales abiertas, tal que estos sistemas también son conocidos como *Sistemas de Recomendación Sociales*.

3.2.3.6. Sistemas Híbridos

Una variedad de técnicas fueron propuestas como base de los sistemas de recomendaciones. Cada una de ellas, tienen desventajas conocidas, como el ya mencionado problema de inicialización de los sistemas colaborativos y basados en contenido. Un *RS Híbrido* combina múltiples técnicas de recomendación para encontrar sinergia entre las mismas. Por ejemplo, un sistema basado en conocimiento puede compensar el problema de inicialización de los sistemas colaborativos, para nuevos perfiles de usuario; así como también, el componente colaborativo puede utilizar sus habilidades estadísticas para encontrar pares de usuarios que compartan preferencias no esperadas, las cuales no podrían haber sido predichas por habilidades basadas en conocimiento (Burke, 2007).

3.3. Big Data

3.3.1. Contexto Histórico

Al igual que todos los términos que surgen a partir de avances tecnológicos, no existe un consenso claro de cómo definir *Big Data*. Manyika et al. (2011) definen este concepto como los conjuntos de datos cuyo tamaño está más allá de la habilidad de las herramientas software de base de datos para capturar, almacenar, gestionar y analizar. Nótese que esta definición es agnóstica del tamaño del conjunto de datos, y no define un tamaño mínimo del mismo, sino que, asume que la tecnología avanza constantemente como así también las herramientas, por lo cual, la definición se "mueveçon el tiempo. Por otro lado, también es interesante tomar otra arista en la definición de este concepto. La consultora Gartner en su sitio web[13] lo define como "Big Data son activos de información caracterizados por su alto volumen, velocidad y variedad que demandan formas innovadoras y rentables de procesamiento de información para mejorar la compresión y la toma de decisiones", haciendo énfasis en la multiplicidad de características de Big Data.

[13] Concepto de Big Data en el glosario de Gartner: `https://www.gartner.com/en/information-technology/glossary/big-data`. Último acceso: Febrero 2021.

El comienzo de sobrecarga de información, recientemente mencionado, data del año 1880, cuando el censo de los Estados Unidos tarda 8 años en tabularse. Ante esta situación Herman Hollerith inventó la máquina tabuladora eléctrica basada en tarjetas perforadas[14] . El censo en 1890 fue un éxito rotundo e, incluso, la máquina que él diseñó fue utilizada para los censos de Canadá, Noruega y Austria al año siguiente. En el año 1941, los científicos empiezan a utilizar el término "explosión de la información", que fuera citado en el periódico The Lawton Constitution[15] , haciendo alusión a la dificultad de administrar toda la información disponible. Gradualmente, se identificaron avances concretos en materia de procesamiento de datos y criptografía, motivados particularmente por los sucesos bélicos de la época. Un ejemplo es el dispositivo llamado Colossus (Copeland, 2004) que buscaba e interceptaba mensajes a una tasa de miles de caracteres por segundo. Unos años más tarde, en 1951 el concepto de *memoria virtual* es introducido por el físico alemán Fritz-Rudolf Güntsch, como una idea que trataba el almacenamiento finito como infinito.

A partir de la década del 80', los avances tecnológicos, especialmente en sistemas MRP (planificación de recursos de fabricación), permitieron nuevas formas de organizar, almacenar y generar datos. En este sentido, IBM se destaca y define una arquitectura para los informes y análisis de negocio (EBIS)[16] , que se convierte en la base del almacenamiento de datos en forma centralizada para usuarios finales (Devlin y Murphy, 1988); es decir, el *data warehousing*. Hacia finales de los 80', Tim Berners-Lee, inventa la *World Wide Web* (Berners-Lee y Cailliau, 1992). Invento que implicaría el impacto más grande hasta la actualidad con respecto a la generación, identificación, almacenamiento y análisis de grandes volúmenes de datos de diversa naturaleza.

El inicio de los años 90' marcan un antes y un después en lo relativo al tratamiento y almacenamiento de datos. El crecimiento tecnológico fue explosivo, tal es así que el almacenamiento digital empieza a ser más conveniente y rentable que el papel para almacenar datos (Morris y Truskowski, 2003). Es en 1990 cuando surgen las plataformas de *Business Intelligence* (BI) y los rediseños de software al estilo *Enterprise Resource Planning* (ERP). En este contexto, Cox y Ellsworth (1997) afirman que el crecimiento de la cantidad de datos que debe

[14] Herman Hollerith, US Census Boureau: `https://www.census.gov/history/www/census_then_now/notable_alumni/herman_hollerith.html`. Último acceso: Febrero 2021.

[15] The Lawton Constitution: `http://www.swoknews.com/`. Último acceso: Febrero 2021.

[16] Acrónimo para EMEA (Europe, Middle East and Africa) Business Information System.

manejar un sistema de información empieza a ser un problema en materia de almacenamiento y visualización de los datos, situación que denominaron como "el problema del Big Data". Así, 1997 es un año clave, en el que se realizan un gran porcentaje de estudios y publicaciones que se enfocan en averiguar cuánta información hay disponible a nivel mundial y su crecimiento[17] , y, en consecuencia, se estima que el crecimiento de Internet es aproximadamente del 100 % anual y que superaría el tráfico de voz para el año 2002 (Coffman y Odlyzko, 1998).

En el año 2001, se introduce el concepto de *Las 3 V's: Volumen, Velocidad y Variabilidad de los datos* (Laney, 2001), fundantes sobre la temática y que sería mundialmente aceptado una década más tarde. Por otro lado, también, en 2001 aparece el concepto de *Software como un Servicio* (SaaS) (Hoch et al., 2001), un modelo disruptivo de servicios centralizados y acceso a los mismos mediante clientes finos (típicamente exploradores web), dando la posibilidad del escalamiento horizontal de sistemas de información y la generación de estándares de comunicación. Esta situación provocó que empresas como Oracle[18] , SAP[19] y Peoplesoft[20] empiecen a centrarse en el uso de servicios web, permitiendo así la generación de datos en forma masiva por usuarios finales. Así, en 2006, nace Apache Hadoop[21] , una solución de código abierto que permite el procesamiento en paralelo y distribuido de enormes cantidades de datos en forma escalable. Posteriormente, en 2008, se empieza a pensar al Big Data como la mayor innovación en informática en la última década, ya que ha transformado la forma en que los motores de búsqueda acceden a la información, las actividades de las compañías, las investigaciones científicas, la medicina, y las operaciones de defensa e inteligencia de los países, entre otras tantas actividades. Más aún, se ha comenzado a ver su potencial para recopilar y organizar datos en todos los ámbitos de la vida cotidiana (Bryant et al., 2008), tales como redes sociales, estadísticas deportivas o avances médicos y genéticos.

3.3.2. Map-Reduce

Map-reduce es un modelo de programación popular para el procesamiento de grandes cantidades de datos mediante computación distribuida (Condie et al.,

[17] Michael Lesk publica *"How much information is there in the world?"* (1997): `http://www.lesk.com/mlesk/ksg97/ksg.html`. Último acceso: Febrero 2021.

[18] Oracle: `https://www.oracle.com`. Último acceso: Febrero 2021.

[19] SAP: `https://www.sap.com`. Último acceso: Febrero 2021.

[20] Peoplesoft: adquirida por Oracle en Enero de 2005.

[21] Apache Hadoop: http://hadoop.apache.org/. Último acceso: Febrero 2021.

2010). En pocas palabras, se especifica una función map que procesa pares clave-valor para generar un conjunto de pares clave-valor intermedios, y una función reduce que combina todos los valores intermedios asociados con la misma clave (Dean y Ghemawat, 2008). En lenguajes de programación funcionales así como también lenguajes de alto nivel modernos, es posible escribir expresiones de estilo lambda, en las cuales es posible paralelizar y ejecutar programas en clusters distribuidos sin la necesidad de tener en cuenta los detalles de partición de datos y subprocesamiento. En este trabajo se utilizará el framework Apache Spark[22] ,

debido a su poder de procesamiento distribuido, su arquitectura basada en datos y su compatibilidad con las librerías necesarias en este trabajo.

3.3.2.1. Arquitectura Hadoop

La librería de software Apache Hadoop es un framework que posibilita el procesamiento de grandes conjuntos de datos entre clusters de computadoras usando modelos de programación simples. Para posibilitar esto, Hadoop se basa en una arquitectura de archivos propia, llamada HDFS[23] (Sistema de Archivos Distribuido Hadoop). En la mayoría de los procesos basados en Hadoop, HDFS es utilizado para almacenar la entrada del paso *map* y la salida del paso *reduce*, pero no los resultados intermedios, ya que ellos se almacenan en el sistema de archivo de cada uno de los nodos (Condie et al., 2010). Según el sitio oficial[24] , HSFS es altamente tolerante a fallos y es diseñado para ser ejecutado en computadoras de bajo costo. Además, HDFS provee una gran productividad accediendo a los datos de una aplicación y es posible usarlo en grandes conjuntos de datos.

HDFS tiene una arquitectura maestro-nodo[25] . Un cluster HDFS (conjunto de nodos maestro-nodos) consiste en un *NameNode*, que es un servidor maestro que maneja el espacio de nombres del sistema de archivos y regula el acceso a archivos. Además, hay un número de *DataNodes*, usualmente uno por cada nodo en el cluster, que maneja el almacenamiento de datos en archivos. Internamente, un archivo es dividido en uno o más bloques, y esos bloques son almacenados en

[22] Sitio web oficial de Apache Spark: `https://spark.apache.org/`. Último acceso: Febrero 2021.

[23] Siglas en inglés para Hadoop Distributed File System.

[24] Arquitectura HDFS: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html. Último acceso: Febrero 2021.

[25] Del inglés, master-node architecture.

DataNodes. Por otro lado, el NameNode ejecuta operaciones tales como abrir, cerrar, y renombrar archivos y directorios.

Tanto el NameNode como el DataNode son piezas de software diseñadas para correr, típicamente, en sistemas operativos GNU/Linux. Además, como HDFS está construido utilizando el lenguaje de programación java, puede ser desplegado en un rango amplio de máquinas.

3.3.2.2. Apache Spark

Apache Spark es una plataforma de código abierto escrita en Scala[26] , para procesamiento de datos a gran escala y preparado para tareas de machine learning iterativas (Meng et al., 2016). En alto nivel, una aplicación spark consiste en un programa "driver" que ejecuta la función "main" y varias operaciones en paralelo en un cluster de computadoras[27] .

El paralelismo en un cluster de computadoras realizado por Spark se basa en particiones de datos, por ejemplo, si se desea procesar un conjunto de datos de un millón de registros y se cuenta con un cluster de 4 nodos (sin contar el nodo principal), cada uno de ellos procesará 250.000 registros. Para conseguir esto de una forma que pueda ser entendible para un desarrollador, la plataforma cuenta con una abstracción de datos llamada RDD o *Resilient Distributed Dataset* (o conjunto de datos distribuido resiliente), la cual es una colección de datos particionados a través de los nodos de cluster que pueden operar en paralelo. Los RDDs son creados desde un archivo HDFS o una colección Scala existente en el programa driver. También es posible almacenarlos en memoria, permitiendo que estas colecciones sean reusadas eficientemente entre operaciones paralelas.

Los RDDs permiten dos tipos de operaciones: *transformaciones*, las cuales crean un conjunto de datos desde uno existente, y *acciones* u *operaciones terminales*, que devuelven un valor al programa driver luego de ejecutar una cierta cantidad de cómputos en el conjunto de datos. Estas operaciones se condicen con el modelo de programación map-reduce en el cual se basa Hadoop. Por ejemplo, map es una transformación que opera en cada uno de los elementos del conjunto de datos y devuelve un nuevo RDD representando los resultados. Por otro lado,

[26] Sitio oficial del lenguaje de programación Scala: https://www.scala-lang.org/. Último acceso: Febrero 2021.

[27] Red de computadoras de alta velocidad que se comportan como si fuesen un único servidor. No confundir con la definición de "cluster" en algoritmos de clustering.

reduce es una operación que agrega todos los elementos de un conjunto de datos usando alguna función y devolviendo un resultado final.

Todas las transformaciones son *lazy*, esto es, secuencias de acciones imperativas las cuales son retrasadas hasta que el resultado es requerido (Launchbury, 1993), es decir, las transformaciones aplicadas a un conjunto de datos son recordadas hasta que es necesario devolver un resultado final al programa driver. Además, es posible almacenar un RDD en memoria (o disco rígido) logrando así mantener elementos disponibles en un cluster para un acceso rápido en caso de que una transformación sea requerida más de una vez.

3.4. Medidas de distancia de texto

3.4.1. Conceptos básicos

3.4.1.1. Information retrieval

Information retrieval (IR) se define como encontrar material (generalmente documentos) de una naturaleza desestructurada (generalmente texto) que satisfaga una necesidad de información de grandes colecciones (generalmente almacenadas en computadoras) (Schütze et al., 2008).

El IR utiliza técnicas probabilísticas, pero también, en los últimos años, investigadores se han centrado en técnicas basadas en conocimiento. Estas últimas, han hecho una significante contribución al IR "inteligente". Más recientemente, la investigación se ha volcado a nuevas técnicas de aprendizaje inductivo basadas en *inteligencia artificial* (IA), las cuales incluyen redes neuronales, aprendizaje simbólico y algoritmos genéticos (Chen, 1995). Cuando hablamos de aprendizaje, nos referimos a un fenómeno multifacético. Los procesos de aprendizaje incluyen la adquisición de un nuevo conocimiento declarativo, la organización del nuevo conocimiento, representaciones efectivas y un descubrimiento de nuevos hechos y teorías a través de la observación y la experimentación. Desde el nacimiento de la era de las computadoras, estas capacidades han querido ser implantadas en las mismas. El estudio y el modelado en computadoras del proceso de aprendizaje en múltiples manifestaciones constituye el propósito principal del *Machine Learning* (ML) (Mitchell et al., 2013).

Los Sistemas de Recomendación a los cuales se hace foco en este trabajo, aplican técnicas que no son más que conceptos derivados del IR y algunos del

ML. Técnicas probabilísticas y determinísticas son utilizadas para extraer información relevante desde diferentes orígenes de datos, así como también generar información valiosa a partir de ellos. Para este último propósito, se han desarrollado métodos basados en el aprendizaje inductivo que demostraron ser muy efectivos y, en algunos casos, tener modelos simples y aplicables que permiten desarrollar RS eficaces y escalables.

3.4.1.2. Unidad de documento

Una *unidad de documento*, es una secuencia de caracteres de longitud fija con las cuales se va a trabajar. Estas secuencias de caracteres pueden estar codificadas por uno o varios bytes o esquemas de codificación multibyte, como UTF-8 o varios estándares específicos de algún país o compañía. Una vez que la codificación esté determinada, se debe decodificar la secuencia de bytes a una secuencia de caracteres.

3.4.1.3. Stopwords

En informática, se llama stopword a palabras que se filtran antes o después del procesamiento de datos del lenguaje natural (Leskovec et al., 2014). Generalmente, este tipo de palabras son extremadamente comunes, y podrían tener poco valor en el momento de seleccionar documentos que coincidan con las necesidades de un usuario (Schütze et al., 2008).

Entonces es necesario seleccionar una lista de stopwords, que serán filtradas en el procesamiento de las unidades de documento, estas listas son llamadas *stop lists*. La estrategia general para el armado de stop lists, es ordenar los términos por *collection frequency*, es decir, el número total de veces que aparece un término en la colección de documentos. Una vez hecho esto, es necesario tomar los términos más frecuentes, a veces filtrados a manos según su contenido semántico relativo al dominio de los documentos que están siendo indexados.

3.4.1.4. Tokenización

Dado una secuencia de caracteres y una unidad de documento definida, la *tokenización* es la tarea de dividirla en distintas piezas, llamadas *tokens*, y, quizás en el mismo momento, desechar ciertos caracteres (como signos de puntuación). Un ejemplo de tokenización de una secuencia de caracteres en idioma inglés, podría ser:

```
Input: Friends, Romans, Countrymen, lend me your ears;
Output: [Friends] [Romans] [Countrymen] [lend] [me] [your] [ears]
```

Estos tokens, son frecuentemente confundidos con palabras, pero es importante hacer la distinción entre token y tipo. Un token es una instancia de una secuencia de caracteres en un documento en particular que están agrupados juntos como una unidad semántica útil para procesamiento. Un *tipo* es la clase de todos los tokens que contienen la misma secuencia de caracteres. Un *término* es un tipo, que es incluido en un diccionario de un sistema de IR. Por ejemplo, si un documento a ser indexado es "to sleep perchance to dream", existen 5 tokens, pero solo 4 tipos (ya que hay dos instancias del token "to"). Sin embargo, si "to" es desechada por ser un stopword, habrá entonces 3 términos: "sleep", "perchance" y "dream".

3.4.1.5. Similaridad

Es de interés poder cuantificar la relación entre los objetos de texto. Existen distintas maneras de medir esa relación. En general se habla de medidas de proximidad (Xu y Wunsch, 2008). Las medidas de proximidad son una generalización para las medidas de similaridad y disimilaridad. En este trabajo utilizaremos medidas de similaridad comúnmente encontradas en la literatura (Gomaa y Fahmy, 2013; Harispe et al., 2015; Lin et al., 1998; Resnik, 1995). Debido al propósito de este trabajo, es de interés, en particular, la similaridad semántica en taxonomías (Resnik, 1995), que se basa en el contenido de información de las unidades documentales.

Las medidas basadas en contenido de información de una unidad de documento (o concepto) en una taxonomía, utilizan el siguiente enfoque. Se define $p(t)$ como:

$$p(t) = \frac{cantidad\,de\,palabras\,asociadas\,a\,una\,definición\;\;sus\,hijos}{cantidad\,total\,de\,palabras\,en\,el\,corpus}$$

De esta forma, el nodo raíz tendrá $p(t) = 0$, mientras los nodos hojas, tendrán valores cercanos a 1. Se define el contenido de información $I(t) = 0$ como:

$$I(t) = -\log p(t)$$

Aplicando el logaritmo negativo a $p(t) = 0$ se logra que los nodos hoja, siendo conceptos muy específicos, contengan mucha información, y que los nodos

más genéricos que se encuentran cerca de la raíz contengan un contenido de información que, por lo contrario, tienda a cero.

Resnik (1995) define, en un conjunto de conceptos C en una taxonomía *es-un*, que permite herencia múltiple, que la similaridad entre dos conceptos es la medida en que ellos comparten información en común, indicado, en este tipo de taxonomías, por el nodo inmediato de más alto nivel que los subsume a ambos, el *subsumidor mínimo* (minimum subsumer o ms por sus siglas en inglés).

Considerando dos términos t_i y t_j, y a $S(t_i, t_j)$ al conjunto de ancestros comunes de t_i y t_j, se define al subsumidor mínimo, $ms(t_i, t_j)$, como al término de $S(t_i, t_j)$ que contiene el máximo contenido de información:

$$\max_{t \in s(t_i, t_j)} I(t) = I(ms(t_i, t_j))$$

La medida de similaridad de Resnik (1995) S_R, es entonces, el contenido de información del subsumidor mínimo de dos términos:

$$S_R = I(ms(t_i, t_j))$$

El enfoque anterior, posee la siguiente particularidad: no tiene en cuenta la similaridad de los nodos con respecto a su subsumidor mínimo. Es intuitivo pensar que dos conceptos abstractos (nodos ubicados en posiciones más cercanas al subsumidor mínimo) son más parecidos entre sí que dos conceptos específicos (más alejados del subsumidor mínimo) (Lin et al., 1998). Para solucionar esto, Lin et al. (1998) tiene en cuenta dos aspectos para calcular la similaridad entre dos conceptos en este tipo de taxonomías: (I) La cantidad de información; y (II) La ubicación relativa entre los nodos hijos respecto al subsumidor mínimo. Definida como la similaridad de Resnik (1995).

La medida de Lin es:

$$S_L(t_i, t_j) = \frac{2S_R(t_i, t_j)}{I(t_i) + I(t_j)}$$

El resultado de esta medida, estará normalizada en el rango $[0, 1]$ y obtiene que nodos más generales, con menor cantidad de información, son más similares entre sí, que dos nodos específicos (ya que la cantidad de información de los mismos

aumentará el denominador de la ecuación) para el mismo subsumidor mínimo. Se ha demostrado que esta definición de similaridad produce una correlación ligeramente mayor con los juicios humanos.

3.4.1.6. Medidas de proximidad

Proximidad es la generalización de similaridad y disimilaridad. La función disimilaridad, también conocida como función de distancia, en un conjunto de datos X, es definida para satisfacer las condiciones. Las condiciones mencionadas, son las utilizadas por Xu y Wunsch (2008) y de relevancia en el presente trabajo:

1. Simetría,
$$D(x_i, x_j) = D(x_j, x_i);$$
2. Positividad,
$$D(x_i, x_j) \geq 0 \quad \forall x_i, x_j;$$

De forma, análoga la función de similaridad es definida satisfaciendo las condiciones:

1. Simetría,
$$S(x_i, x_j) = S(x_j, x_i);$$
2. Positividad,
$$0 \leq S(x_i, x_j) \leq 1, \quad \forall x_i, x_j$$

Si bien el término matemático de distancia exige una serie de supuestos rigurosos (Xu y Wunsch, 2008), en este trabajo utilizaremos la noción de distancia y de disimilaridad en forma indistinta, y nos basaremos en las medidas de proximidad habitualmente utilizadas para comparación de texto. Por lo tanto, Para transformar una medida de similaridad $S(x_i, x_j)$ en una de distancia $D(xi, xj)$ que cumpla $0 \leq D(x_i, x_j) \leq 1$, haremos la normalización de la misma en el intervalo $[0, 1]$ y luego aplicaremos el cálculo $D(x_i, x_j) = 1 - S(x_i, x_j)$ y recíprocamente (Leale et al., 2013).

3.4.1.7. Modelo de espacio vectorial

En el modelo de espacio vectorial, un texto es representado como un vector de términos. Si las palabras son elegidas como términos, entonces cada palabra del vocabulario sería una dimensión independiente en el espacio vectorial (Singhal

et al., 2001). Todo texto puede ser representado por un vector en este espacio dimensional. Si un término pertenece a un documento, éste obtiene un valor distinto de cero en el vector, junto con la dimensión correspondiente al término. Como un documento contiene un conjunto limitado de términos (el vocabulario puede contener millones de términos), muchos de los vectores pueden ser muy dispersos. La mayoría de los sistemas basados en vectores trabajan en el cuadrante positivo, es decir, a ningún término se le asigna un valor positivo.

Para asignar un valor numérico a un documento en una consulta, el modelo mide la similaridad entre el vector ingresado en ella y el vector del documento al cual se quiere consultar. La similaridad entre dos vectores no es inherente al modelo. Típicamente, el ángulo entre los dos vectores es usado como medida de divergencia entre los mismos, y el coseno del ángulo es usado como similaridad numérica, ya que el coseno tiene la propiedad de ser tener resultado 1 cuando los vectores son idénticos y 0 cuando los vectores son ortogonales (explicado en detalle más adelante). Como una alternativa, el producto escalar entre dos vectores, es también usado como medida de similaridad. Si todos los vectores están forzados a tener longitud 1, es decir, vectores unitarios, entonces el coseno del ángulo entre los vectores, tiene el mismo resultado que el producto escalar.

Si $\overrightarrow{D}$ es el vector del documento y $\overrightarrow{Q}$ es el vector de la consulta, la similaridad entre $\overrightarrow{D}$ y $\overrightarrow{Q}$ es representada como:

$$S(\vec{D}, \vec{Q}) = \sum_{ti \in Q,D} W_{t_{iQ}}.W_{t_{iD}}$$

donde $W_{t_{i\vec{Q}}}$ es el valor de la componente número i del vector $\overrightarrow{Q}$ y $W_{t_{i\vec{D}}}$ es el valor de la componente número i del vector $\overrightarrow{D}$. También definido como el peso del término i en el documento D. Cualquier término no presente en la consulta o en el documento tendrá valor cero en $W_{t_{i\vec{Q}}}$ o $W_{t_{i\vec{D}}}$, por lo cual es posible hacer la sumatoria solo de los términos en común entre la consulta y el documento.

3.4.1.8. Distancia del coseno

La distancia del coseno puede ser la mayor frecuentemente aplicada en términos de similaridad en IR (Korenius et al., 2007). Al aplicar la distancia del coseno se obtiene un resultado que se encuentra en el rango $[-1, 1]$. El valor -1 significa que los vectores tienen la misma dirección, pero sentidos opuestos. El valor 1, por lo contrario, significa que el ángulo comprendido entre los vectores es cero.

Particularmente en IR, es de interés el intervalo $[0, 1]$ ya que todos los componentes de un vector que representa a un documento, son no negativos. De esta interpretación, se deriva la definición de la distancia del coseno restando la medida del coseno, de su máximo valor:

$$d_c(\vec{D}_i, \vec{D}_j) = 1 - cos(\vec{D}_i, \vec{D}_j) = 1 - \frac{\vec{D}_i' \vec{D}_j}{\sqrt{\vec{D}_i' \vec{D}_i}\sqrt{\vec{D}_j' \vec{D}_j}}$$

donde $i \leq i, j \leq n$. Los símbolos $\overrightarrow{D_i}$ y $\overrightarrow{D_j}$ son documentos en forma de vectores y dces la distancia entre ellos.

Para simplificar, y teniendo en cuenta que estamos hablando de documentos en forma de vectores, la distancia del coseno se puede derivar de la fórmula del *producto escalar* (producto punto). Siendo $\overrightarrow{D_i}$ y $\overrightarrow{D_j}$ dos documentos en forma de vectores, se define el producto escalar entre ellos, como:

$$\vec{D}_i . \vec{D}_j = \left\| \vec{D}_i \right\| . \left\| \vec{D}_j \right\| .cos(\theta)$$

siendo $\left\| \overrightarrow{D_i} \right\|$ y $\left\| \overrightarrow{D_j} \right\|$ los módulos de los vectores $\overrightarrow{D_i}$ y $\overrightarrow{D_j}$ respectivamente, y θ el ángulo formado entre ellos. Entonces:

$$cos(\theta) = \frac{\vec{D}_i . \vec{D}_j}{\left\| \vec{D}_i \right\| . \left\| \vec{D}_j \right\|} = \frac{\sum_{i=1}^{n} d_{ii} d_{ji}}{\sqrt{\sum_{i=1}^{n} d_{ii}^2}\sqrt{\sum_{i=1}^{n} d_{ji}^2}}$$

Donde d_i y d_j son los componentes de los vectores $\overrightarrow{D_i}$ y $\overrightarrow{D_j}$ respectivamente.

4. Problema de investigación

4.1. Hipótesis de trabajo

A partir del relevamiento del estado del arte se infiere que las medidas de rendimiento obtenidas en las entradas de RS no son los suficientemente eficientes para mejorar la experiencia de usuario y reducir las probabilidades de error en sitios CQA.

Por tal motivo, y como respuesta a la hipótesis planteada, se presentará un método basado en una arquitectura Big Data que posibilite aplicar ensamble de clustering a grandes conjuntos de datos y lograr medidas de rendimiento superadoras.

4.2. Procedimiento de desarrollo

Este trabajo comenzará con una búsqueda de material científico relacionado a RS en general, RS no personalizados basados en análisis de texto, su aplicación en sitios de CQA, un análisis de algoritmos de comparación de texto del estado del arte y su aplicación a grandes volúmenes de datos mediante métodos de ensamble de clustering y, también, una evaluación de arquitecturas de software adecuadas para un enfoque Big Data e infraestructuras acordes. Esto puede ser realizado mediante sitios o librerías digitales, tales como Google Scholar[28] , IEEExplore Digital Library[29] , ScIELO[30] , Harvard Library[31] o el portal del CAICYT-CONICET[32] , entre otros. Definida la hipótesis correctamente y el plan de trabajo, se iniciará el desarrollo de un software de código abierto partiendo del proyecto "text comparison"[33] perteneciente al repositorio Git del departamento de Ingeniería en Sistemas de Información de la UTN FRRo. Se importarán las piezas de software del código del proyecto del estado del arte recientemente mencionado para usarlas mediante un enfoque Big Data, con nuevas herramientas basadas en Cloud Computing, Hadoop y una arquitectura de software comple-

[28] Google Scholar: `https://scholar.google.com.ar`. Último acceso Agosto 2018.

[29] IEEExplore Digital Library: `http://ieeexplore.ieee.org`. Último acceso Agosto 2018.

[30] SciELO: `http://www.scielo.org`. Último acceso Agosto 2018.

[31] Harvard library: `https://library.harvard.edu`. Último acceso Agosto 2018.

[32] Centro Argentino de Información Científica y Tecnológica del CONICET: `http://www.caicyt-conicet.gov.ar/sitio`. Último acceso Agosto 2018.

[33] Repositorio GitHub: `https://github.com/Departamento-Sistemas-UTNFRRO/text_comparison`.

tamente nueva que optimice este tipo de desarrollo. Una vez que se inicie el desarrollo del proyecto, serán evaluadas distintas opciones de herramientas y entornos que se utilizarán, Esto incluye:

- Lenguajes de programación y librerías inherentes al mismo.
- Almacenes de datos, frameworks y proyectos de terceros que puedan ser incorporados en la arquitectura Big Data.
- Arquitecturas de software, patrones, modelos y buenas prácticas.
- Infraestructura: local, distribuida en una red de computadoras físicas, o distribuida y virtualizada en la nube.

Paralelamente al desarrollo, se identificará y documentará la nueva solución de acuerdo con los requerimientos de la Maestría en Ingeniería en Sistemas de Información, a fin de obtener un trabajo de investigación de tesis de maestría de excelencia, y acorde con los parámetros que caracterizan a la institución. Por último, una vez finalizado el desarrollo, se realizará un registro con los indicadores resultantes, se validará la propuesta, se explicitarán los resultados obtenidos y se elaborarán las conclusiones, a fin de abrir y/o profundizar en nuevas líneas de investigación.

4.2.1. Método propuesto

Se propone el método EQuAL (*Ensemble method for community Question Answering sites based on cLustering*), que mejora la calidad y eficiencia para recomendar preguntas en un sitio de CQA. Este método está basado en una arquitectura Big Data distribuida y tiene en cuenta diversas distancias de texto, combinadas mediante un método de ensamble de clustering.

El desarrollo para este trabajo de tesis está basado en dos pasos, como se muestra en la Figura 2. El primer paso es la generación de un conjunto de particiones. El mismo comenzará aplicando los distintos algoritmos de medidas de similaridad de texto del estado del arte al conjunto de datos de entrada. Este procedimiento tendrá como resultado un número D de matrices de distancias. Por cada matriz de distancias, se aplicarán N corridas de algoritmos de clustering, cada uno con un número k de elementos seleccionados al azar, que es un parámetro de entrada del algoritmo de clustering. Esta combinación de D matrices y N clusters, resultará en $D \times N$ corridas del proceso de clustering en total. Esta

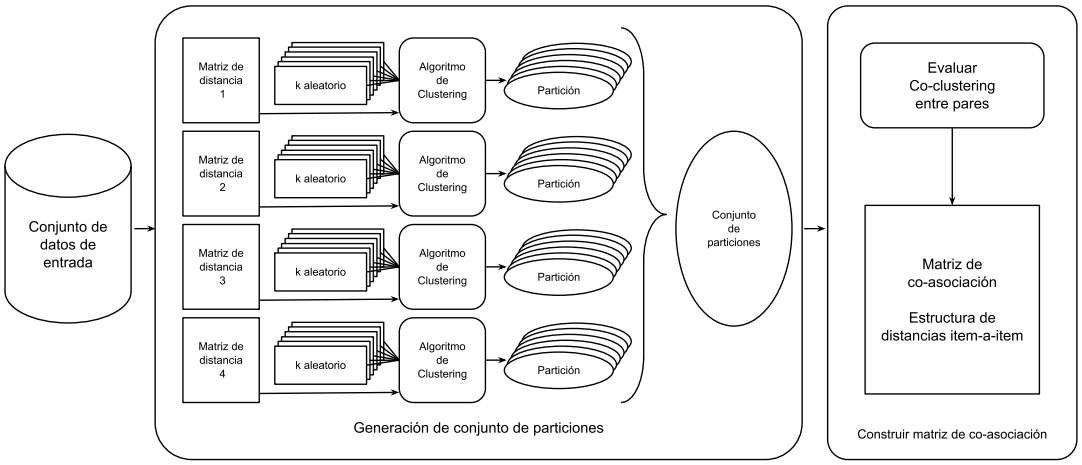

Figura 2: Método EQuAL para la generación de matrices de co-asociación desde el conjunto de datos original.

configuración obtendrá un conjunto de particiones como resultado, con el fin de resumir la estructura de cada una de las particiones generadas por los algoritmos de clustering. El segundo paso es construir una matriz de co-asociación a partir del conjunto de particiones. Para tal fin, se aplica un algoritmo de ensamble de clustering de acumulación de evidencias, que combinará cada una de estas particiones, dando como salida una matriz de co-asociación, que contiene en cada posición la proporción de veces que los elementos i, j caen juntos en el mismo grupo de la salida de clustering, a lo largo de las $D \times N$ particiones. La matriz de co-asociación, que es una representación integrada de las relaciones subyacentes entre los datos originales, será la entrada para RS en sitios CQA. Además, tiene la característica de ser adimensional, insesgada y comprende toda la variabilidad propia de los algoritmos de clustering, por lo cual, mejora la estructura de distancia item-item que es necesaria como entrada para un RS basado en contenido, incorporando varios aspectos de las distancias entre elementos de texto, en lugar de usar solo una simple medida basada individualmente en aspectos de cada una de las medidas de distancia.

El armado de matrices, la combinación de las mismas y la aplicación de estrategias estadísticas, implica un aumento significativo del volumen de datos y requiere una capacidad de cálculo intensiva. Una arquitectura Big Data que realice el procesamiento distribuido de los mismos es fundamental para este proceso. Además del volumen de datos con el cual se trabajará, se variarán distintos parámetros, tales como la medida de similaridad y valores de umbral involucrados en procesos de clustering, con el fin de obtener resultados confiables; lo cual

redunda en múltiples ejecuciones de toda la solución. Debe destacarse que, en un primer momento, se implementarán experimentos basados en una infraestructura MapReduce aplicados con frameworks basados en Hadoop y cluster computing, desplegados en servidores elásticos en la nube, lo cual provee la ventaja de procesar grandes cantidades de datos en instancias dinámicamente escalables.

5. Experimentos

6. Resultados

7. Conclusiones

Bibliografía

Adomavicius, Gediminas y Tuzhilin, Alexander (2005). «Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions». *IEEE transactions on knowledge and data engineering*, **17(6)**, pp. 734–749.

Anuyah, Oghenemaro; Azpiazu, Ion Madrazo; McNeill, David y Pera, Maria Soledad (2017). «Can Readability Enhance Recommendations on Community Question Answering Sites?».

Armstrong, Jon Scott (2001). *Principles of forecasting: a handbook for researchers and practitioners*, tomo 30. Springer Science & Business Media.

Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier et al. (1999). *Modern information retrieval*, tomo 463. ACM press New York.

Berners-Lee, Timothy J y Cailliau, Robert (1992). «World-wide web».

Bryant, Randal; Katz, Randy H y Lazowska, Edward D (2008). «Big-data computing: creating revolutionary breakthroughs in commerce, science and society».

Burke, Robin (2000). «Knowledge-based recommender systems». *Encyclopedia of library and information systems*, **69(Supplement 32)**, pp. 175–186.

Burke, Robin (2007). «Hybrid web recommender systems». En: *The adaptive web*, pp. 377–408. Springer.

Chen, Hsinchun (1995). «Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms». *Journal of the American society for Information Science*, **46(3)**, pp. 194–216.

Chen, Pei-Yu; Chou, Yen-Chun y Kauffman, Robert J (2009). «Community-based recommender systems: Analyzing business models from a systems operator's perspective». En: *2009 42nd Hawaii International Conference on System Sciences*, pp. 1–10. IEEE.

Coffman, Kerry G y Odlyzko, Andrew M (1998). «The size and growth rate of the Internet». *First Monday*, **3(10)**, pp. l–25.

Condie, Tyson; Conway, Neil; Alvaro, Peter; Hellerstein, Joseph M; Elmeleegy, Khaled y Sears, Russell (2010). «MapReduce online.» En: *Nsdi*, tomo 10, p. 20.

Copeland, B Jack (2004). «Colossus: Its origins and originators». *IEEE Annals of the History of Computing*, **(4)**, pp. 38–45.

Cox, Michael y Ellsworth, David (1997). «Application-controlled demand paging for out-of-core visualization». En: *Visualization'97., Proceedings*, pp. 235–244. IEEE.

De Battista, Anabella; Cristaldo, Patricia; Ramos, Lautaro; Nunez, Juan Pablo; Retamar, Soledad; Bouzenard, Daniel y Herrera, Norma Edith (2016). «Minería de datos aplicada a datos masivos». En: *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*, .

Dean, Jeffrey y Ghemawat, Sanjay (2008). «MapReduce: simplified data processing on large clusters». *Communications of the ACM*, **51(1)**, pp. 107–113.

Devlin, Barry A. y Murphy, Paul T. (1988). «An architecture for a business and information system». *IBM systems Journal*, **27(1)**, pp. 60–80.

Eppler, Martin J y Mengis, Jeanne (2004). «The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines». *The information society*, **20(5)**, pp. 325–344.

Fred, Ana LN y Jain, Anil K (2005). «Combining multiple clusterings using evidence accumulation». *IEEE transactions on pattern analysis and machine intelligence*, **27(6)**, pp. 835–850.

Gandomi, Amir y Haider, Murtaza (2015). «Beyond the hype: Big data concepts, methods, and analytics». *International Journal of Information Management*, **35(2)**, pp. 137–144.

Goldberg, David; Nichols, David; Oki, Brian M y Terry, Douglas (1992). «Using collaborative filtering to weave an information tapestry». *Communications of the ACM*, **35(12)**, pp. 61–70.

Gomaa, Wael H y Fahmy, Aly A (2013). «A survey of text similarity approaches». *International Journal of Computer Applications*, **68(13)**, pp. 13–18.

Harispe, Sébastien; Ranwez, Sylvie; Janaqi, Stefan y Montmain, Jacky (2015). «Semantic similarity from natural language and ontology analysis». *Synthesis Lectures on Human Language Technologies*, **8(1)**, pp. 1–254.

Hoch, Fred; Kerr, Michael; Griffith, Anne et al. (2001). «Software as a service: Strategic backgrounder». *Software & Information Industry Association (SIIA)*.

Jeon, Jiwoon; Croft, W Bruce; Lee, Joon Ho y Park, Soyeon (2006). «A framework to predict the quality of answers with non-textual features». En: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 228–235. ACM.

Joulin, Armand; Grave, Edouard; Bojanowski, Piotr; Douze, Matthijs; Jégou, Hérve y Mikolov, Tomas (2016). «Fasttext. zip: Compressing text classification models». *arXiv preprint arXiv:1612.03651*.

Korenius, Tuomo; Laurikkala, Jorma y Juhola, Martti (2007). «On principal component analysis, cosine and Euclidean measures in information retrieval». *Information Sciences*, **177(22)**, pp. 4893–4905.

Laney, Doug (2001). «3D data management: Controlling data volume, velocity and variety». *META group research note*, **6(70)**, p. 1.

Launchbury, John (1993). «Lazy imperative programming». En: *Workshop on State in Programming Languages, Copenhagen, Denmark, ACM*, .

Leale, Guillermo; Milone, Diego H; Bayá, Ariel E; Granitto, Pablo Miguel y Stegmayer, Georgina (2013). «A novel clustering approach for biological data using a new distance based on Gene Ontology». En: *XIV Argentine Symposium on Artificial Intelligence (ASAI)-JAIIO 42 (2013).*, .

Leskovec, Jure; Rajaraman, Anand y Ullman, Jeffrey David (2014). *Mining of massive datasets*. Cambridge university press.

Li, Baichuan y King, Irwin (2010). «Routing questions to appropriate answerers in community question answering services». En: *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1585–1588. ACM.

Li, Yuhua; McLean, David; Bandar, Zuhair A; O'shea, James D y Crockett, Keeley (2006). «Sentence similarity based on semantic nets and corpus statistics». *IEEE transactions on knowledge and data engineering*, **18(8)**, pp. 1138–1150.

Lilien, GL; Kotler, P y Moorthy, KS (1992). «Marketing models Prentice-Hall». *Englewood Cliffs, NJ*.

Lin, Dekang et al. (1998). «An information-theoretic definition of similarity.» En: *Icml*, tomo 98, pp. 296–304. Citeseer.

Lops, Pasquale; De Gemmis, Marco y Semeraro, Giovanni (2011). «Content-based recommender systems: State of the art and trends». En: *Recommender systems handbook*, pp. 73–105. Springer.

Manyika, James; Chui, Michael; Brown, Brad; Bughin, Jacques; Dobbs, Richard; Roxburgh, Charles y Byers, Angela H (2011). «Big data: The next frontier for innovation, competition, and productivity».

Meng, Xiangrui; Bradley, Joseph; Yavuz, Burak; Sparks, Evan; Venkataraman, Shivaram; Liu, Davies; Freeman, Jeremy; Tsai, DB; Amde, Manish; Owen, Sean et al. (2016). «Mllib: Machine learning in apache spark». *The Journal of Machine Learning Research*, **17(1)**, pp. 1235–1241.

Mikolov, Tomas; Chen, Kai; Corrado, Greg y Dean, Jeffrey (2013). «Efficient estimation of word representations in vector space». *arXiv preprint arXiv:1301.3781*.

Mitchell, RS; Michalski, JG y Carbonell, TM (2013). *An Artificial Intelligence Approach*. Springer.

Morris, Robert JT y Truskowski, Brian J (2003). «The evolution of storage systems». *IBM systems Journal*, **42(2)**, pp. 205–217.

Murthi, BPS y Sarkar, Sumit (2003). «The role of the management sciences in research on personalization». *Management Science*, **49(10)**, pp. 1344–1362.

Powell, Michael James David (1981). *Approximation theory and methods*. Cambridge university press.

Resnick, Paul y Varian, Hal R (1997). «Recommender systems». *Communications of the ACM*, **40(3)**, pp. 56–58.

Resnik, Philip (1995). «Using information content to evaluate semantic similarity in a taxonomy». *arXiv preprint cmp-lg/9511007*.

Ricci, Francesco; Rokach, Lior y Shapira, Bracha (2011). «Introduction to recommender systems handbook». En: *Recommender systems handbook*, pp. 1–35. Springer.

Rich, Elaine (1979). «User modeling via stereotypes». *Cognitive science*, **3(4)**, pp. 329–354.

Salton, G y McGill, M J (1983). «Introduction to modern information retrieval». *International Student Edition*.

Salton, Gerard (1989). «Automatic text processing: The transformation, analysis, and retrieval of». *Reading: Addison-Wesley*.

Schafer, J Ben; Frankowski, Dan; Herlocker, Jon y Sen, Shilad (2007). «Collaborative filtering recommender systems». En: *The adaptive web*, pp. 291–324. Springer.

Schütze, Hinrich; Manning, Christopher D y Raghavan, Prabhakar (2008). *Introduction to information retrieval*, tomo 39. Cambridge University Press.

Shardanand, Upendra y Maes, Pattie (1995). «Social information filtering: algorithms for automating "word of mouth"». En: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210–217.

Singhal, Amit et al. (2001). «Modern information retrieval: A brief overview». *IEEE Data Eng. Bull.*, **24(4)**, pp. 35–43.

Sinha, Rashmi R; Swearingen, Kirsten et al. (2001). «Comparing recommendations made by online systems and friends.» En: *DELOS*, .

Strehl, A y Chosh, J (2002). «Knowledge reuse framework for combining multiple partitions». *Journal of Machine learning Research*, **33(3)**, pp. 583–617.

Wang, Yuanyuan; Chan, Stephen Chi-Fai y Ngai, Grace (2012). «Applicability of demographic recommender system to tourist attractions: A case study on trip advisor». En: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, tomo 3, pp. 97–101. IEEE.

Xu, Rui y Wunsch, Don (2008). *Clustering*, tomo 10. John Wiley & Sons.

Yang, Liu; Qiu, Minghui; Gottipati, Swapna; Zhu, Feida; Jiang, Jing; Sun, Huiping y Chen, Zhong (2013). «Cqarank: jointly model topics and expertise in community question answering». En: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 99–108. ACM.