UNIVERSIDAD DE CUENCA

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE SISTEMAS

Sistema de Recomendación de Contenido para TV Digital basado en Ontologías

Tesis previa a la obtención del título de Ingeniero de Sistemas

Autores:

Johnny Javier Ávila Montalvo.
Xavier Fernando Riofrío Machado.

Director:

Ing. Kenneth Samuel Palacio Baus, MSc.

Cuenca - Ecuador
2014

Resumen.

Palabras clave: Sistemas de recomendación,Web semántica, Televisión digital.

La tesis "*Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*", forma parte del proyecto aprobado y financiado por la Dirección de Investigación de la Universidad de Cuenca (DIUC) denominado: *Aplicación de Tecnologías Semánticas para Disminuir la Sobrecarga de Información en Usuarios de TV digital.*

La llegada de Televisión Digital, evidencia las profundas transformaciones que esta nueva tecnología incorpora a la experiencia del usuario ya que al posibilitar el transporte ágil de grandes volúmenes de información, propicia la disponibilidad de una excesiva oferta televisiva en los hogares. Este escenario, en el que el usuario se enfrenta a una amplísima cantidad de programación disponible, podrá llegar a abrumar incluso al televidente más entusiasta, al punto que su experiencia televisiva se vea degradada paulatinamente y evolucione en una tediosa e interminable búsqueda de programas de entre miles de opciones.

Este contexto, que enmarca un evidente problema a ser considerado, ha favorecido la creación de Sistemas de Recomendación de Contenido, que contemplados en el campo de la Televisión Digital, surgen como un servicio de asistencia al usuario para la búsqueda y selección de alternativas televisivas de entretenimiento acorde a sus gustos e intereses.

Johnny Javier Ávila Montalvo.
Xavier Fernando Riofrío Machado.

Abstract

Keywords: Recommender Systems, Semantic Web, Digital Television.

This thesis work, "A Digital Television Content Recommender System based on Ontologies", is part of the project "Application of Semantic Technologies in reducing the information overload in Digital Television users" funded by the Research Direction the University of Cuenca, (DIUC - Dirección de Investigación de la Universidad de Cuenca).

Most users will experience very deep changes in the way they watch television with the arrive of Digital Television Technology, since a high volume of information can be transmitted from TV Broadcasters thanks to the set of improvements these technologies incorporate to communication systems. However, this may cause an information overload that will be present in every home with a television, overwhelming even the most enthusiast user when facing this massive tv program offer. Users will have to spend most of their time looking for programs they might like.

This scenario frames a problem that has been approached with the creation of Content Recommender Systems, that in the context of Digital Television, arise as support services that will help users in filtering the tv program offer depending on their interests and preferences.

Johnny Javier Ávila Montalvo.
Xavier Fernando Riofrío Machado.

Índice general

Índice de figuras

Índice de tablas

Yo, *Johnny Javier Ávila Montalvo*, autor de la tesis *Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*, certifico que todas las ideas, opiniones, y contenidos expuestos en la presente investigación, son de exclusiva responsabilidad de sus autores.

Johnny Javier Ávila Montalvo
C.I. 0309022405

Yo, *Xavier Fernando Riofrío Machado*, autor de la tesis *Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*, certifico que todas las ideas, opiniones, y contenidos expuestos en la presente investigación, son de exclusiva responsabilidad de sus autores.

Xavier Fernando Riofrío Machado
C.I. 0104640354

Yo, *Johnny Javier Ávila Montalvo*, autor de la tesis *Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de *Ingeniero de Sistemas*. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, Junio 2014.

Johnny Javier Ávila Montalvo
C.I. 0309022405

Yo, *Xavier Fernando Riofrío Machado*, autor de la tesis *Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de *Ingeniero de Sistemas*. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, Junio 2014.

Xavier Fernando Riofrío Machado
C.I. 0104640354

CERTIFICO

Que el presente proyecto de tesis: *Sistema de Recomendación de Contenido para TV Digital basado en Ontologías*, ha sido realizado bajo mi dirección.

Ing. Kenneth Samuel Palacio Baus, MSc
Director de Tesis

Agradecimientos

Creemos justo expresar nuestro más sincero agradecimiento al Ing. Kenneth Palacio Baus, MSc. por dedicar su tiempo y esfuerzo en la tutoría de este proyecto, por su apoyo absoluto e incondicional y por sus consejos, sugerencias y correcciones a lo largo de toda la realización de esta tesis. De la misma forma, queremos agradecer al Ing. Mauricio Espinoza Mejía Ph.D director del proyecto *Aplicación de Tecnologías Semánticas para disminuir la sobrecarga de información en usuarios de TV Digital* por la ayuda y confianza brindada para la realización de este proyecto. Asimismo hacemos extensivo nuestro agradecimiento al Ing. Victor Saquicela Galarza Ph.D por la ayuda brindada y por incluirnos en la ejecución del proyecto actual. Y por último a la Universidad de Cuenca y a la Facultad de de Ingeniería por la formación académica brindada.

Dedicatoria

Dedico este trabajo a mis padres, hermanos y amigos por apoyarme en la realización de esta tesis.

Johnny Javier Ávila Montalvo.

El presente trabajo esta dedicado a mi patria en la cual tuve la dicha de haber nacido, crecido y de haber podido llegar hasta este punto. Va por ti, mas que el sol contemplamos lucir, ECUADOR QUERIDO!.

Xavier Fernando Riofrío Machado.

Capítulo 1

Introducción.

1.1. Planteamiento del Problema.

El mejoramiento de los sistemas de telecomunicaciones experimentado en los últimos tiempos, ha facilitado un incremento notable de la cantidad de información que puede estar disponible para las personas en múltiples contextos. Particularmente, en escenarios como Internet o en servicios de entretenimiento bajo demanda, existe una infinidad de opciones disponibles sobre cualquier tema que se pueda imaginar. Un sistema de recomendación trata esencialmente de simplificar y agilizar la búsqueda de alternativas para un usuario con un perfil determinado, reduciendo así lo que se denomina *sobrecarga de información*.

Es muy común, que un usuario desconozca todos los servicios y opciones a su disposición, por lo que la incorporación de un sistema de recomendación mantendría al usuario informado sobre sus mejores alternativas de acuerdo a sus gustos o historial de búsquedas. Dos ejemplos muy conocidos de la aplicación de este tipo de sistemas lo conforman, en primer lugar, la mayor tienda de compras en línea *Amazon*[1], donde el usuario recibe sugerencias vía e-mail o mientras navega en la página, sobre objetos, accesorios o complementos relacionados a su historial de búsqueda o de compras, esto, con el objetivo de mejorar su experiencia; por otra parte, está el gigante del entretenimiento audio-visual *Netflix*[2], que busca mantener el interés de sus usuarios tanto en sus contenidos como en utilizar su plataforma de despliegue mediante un sistema de recomendación personalizado. En este contexto, considerando que existe una extensa cantidad de *ítems* como películas y programas

[1] www.amazon.com
[2] www.netflix.com

de televisión, esta alternativa se constituye como el mecanismo ideal para capturar la atención de los clientes y mantener una buena posición en ese mercado tan competitivo.

1.2. Antecedentes.

En el marco del proyecto aprobado y financiado por la Dirección de Investigación de la Universidad de Cuenca (DIUC) denominado: " Aplicación de Tecnologías Semánticas para Disminuir la Sobrecarga de Información en Usuarios de TV digital", se pretende diseñar un sistema de recomendación para la programación televisiva que tome en consideración las preferencias del usuario. El proyecto se divide en dos grandes etapas como se observa en 1.1.

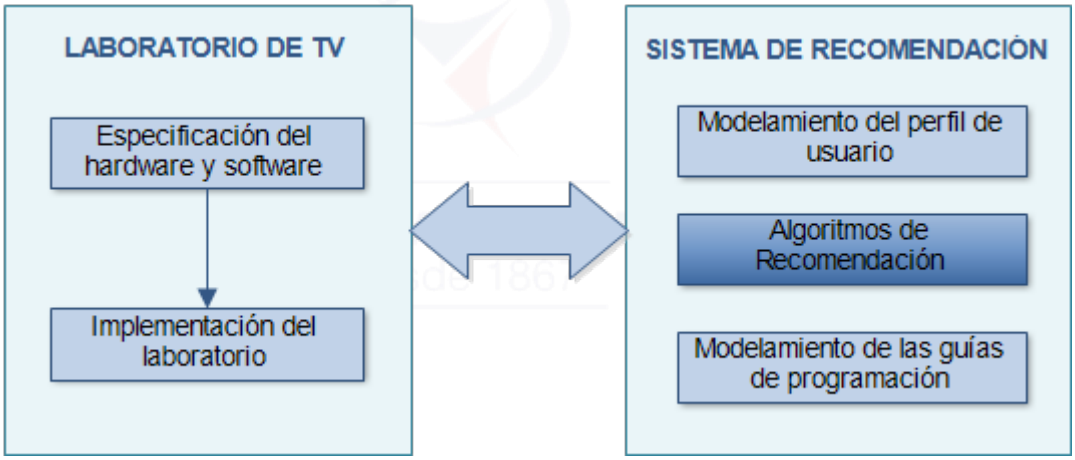


Figura 1.1: *Esquema general del proyecto "Aplicación de Tecnologías Semánticas para Disminuir la Sobrecarga de Información en Usuarios de TV digital"*

La etapa denominada "Laboratorio de TV digital" constituye la primera fase del proyecto. Esta etapa se encargará de buscar las alternativas óptimas para simular un escenario real de transmisión y recepción de señal televisiva. Este diseño estará formado por elementos de hardware y software que posibiliten transmitir y recibir múltiples contenidos televisivos. Por otra parte, la etapa denominada "Sistema de Recomendación", se encargará de diseñar una infraestructura semántica por medio de la utilización de ontologías para captar las preferencias de los televidentes y los contenidos de los programas. Esta etapa usará también algoritmos de recomendación que permitan seleccionar únicamente aquellos programas que son de interés para el televidente en base a sus preferencias, con el objetivo de no so-

brecargarlo de información.

Este proyecto de tesis abarcará el aspecto de los algoritmos de recomendación usando tecnología semántica. Este componente implementará las interfaces adecuadas para interactuar con los otros módulos del sistema de recomendación.

1.3. Justificación.

Con el advenimiento de la era de la Televisión Digital, el problema de la sobrecarga de información disponible para el usuario, se hace más visible, por lo cual es necesario contar con un sistema de recomendación, de tal manera que el televidente no tenga que invertir gran parte de su tiempo examinando la totalidad de canales disponibles de manera individual o a través de una guía de televisión para encontrar lo que desea ver, sino que, mediante este sistema de recomendación, el usuario reciba la programación "ideal" para él por medio de una notificación, donde el término *ideal*, se enfoca en describir una programación televisiva lo más aproximadamente cercana a los gustos y preferencias del usuario.

Las recomendaciones emitidas por el sistema son almacenadas y analizadas en base a nivel de aceptación del usuario y se consideran para la generación de futuras recomendaciones. Lo que significa que mientras más se use el sistema, mejor recomendaciones proporcionará.

Considerando que para encontrar el contenido adecuado a recomendar se necesita analizar la información contenida en la programación televisiva y aquella asociada a cada usuario en particular, en este proyecto se propone la implementación de un *Sistema de Recomendación Semántico* (SRS) fundamentado en un modelo que recibirá dos ontologías, la del perfil del usuario y la de guía de programación, que se caracterizan por haber atravesado un proceso de enriquecimiento semántico que facilita el procesamiento inmediato de la información.

Los proveedores de contenido televisivo también se pueden beneficiar del uso de un SRS, ya que contarían con una herramienta que les permitirá fortalecer el interés de un cliente en sus productos, ahorrándole tiempo en primera instancia y luego permitiendo que pueda acceder a sus programas favoritos a través de predicciones basadas en su comportamiento como usuario de la televisión digital.

1.4. Objetivos.

1.4.1. Objetivo General.

Desarrollar un Sistema de Recomendación de contenidos basado en tecnologías semánticas para el proyecto ***Aplicación de Tecnologías Semánticas para disminuir la sobrecarga de información en usuarios de TV Digital*** de la Universidad de Cuenca.

1.4.2. Objetivos Específicos.

- Reconocerlas diferentes tecnologías utilizadas en los Sistemas de recomendación Semánticos – SRS.
- Determinar el estado del arte de los *Sistema de Recomendación* (SR) y comparar los diferentes enfoques presentados hasta la fecha.
- Analizar y utilizar las ontologías del perfil de usuario y guías de programación como entradas que alimenten al SRS a desarrollarse.
- Evaluar la respuesta y los diferentes comportamientos del algoritmo de recomendación para diversos parámetros de ajuste.
- Comparar los resultados obtenidos con el objeto de encontrar el algoritmo de recomendación (o combinación de ellos) con mejor desempeño.

1.5. Alcance.

Este proyecto pretende desarrollar un Sistema de Recomendación de contenidos para Tv digital basado en ontologías, para lo cual en una primera etapa, se investigará el estado del arte de los SR, particularmente aquellos basados en tecnologías semánticas.

A continuación se analizará varios algoritmos de recomendación propuestos en diferentes publicaciones científicas, buscando establecer las mejores opciones, con el fin de incorporarlas durante la implementación del sistema que generará este proyecto.

La etapa investigativa culmina con el desarrollo de un algoritmo prototipo, que irá evolucionando a medida que se efectúen diferentes pruebas y verificaciones acorde al comportamiento de los usuarios del sistema.

Los resultados obtenidos permitirán evaluar el desempeño de los algoritmos estudiados, y compararlo con el del sistema desarrollado en este proyecto.

1.6. Organización de la tesis.

La siguiente tesis de grado estará compuesto por la siguiente estructura:

- **Capitulo 1:** ***Introducción.*** Presentación del proyecto, alcance y objetivos realizados en esta tesis.
- **Capitulo 2:** ***Web Semántica.*** Introducción a la Web 3.0 , arquitectura, tecnologías en uso y aplicaciones.
- **Capitulo 3:** ***Sistemas de Recomendación de Contenidos.*** Historia, actualidad y tipos de los Sistemas de recomendación.
- **Capitulo 4:** ***Desarrollo e Implementación del Sistema.*** En esta sección se explicará cómo el sistema fue desarrollado, su diseño, las características que posee y detallará su de implementación.
- **Capitulo 5:** ***Evaluación del Sistema de Recomendación.*** Consiste de las pruebas realizadas en el proyecto, los parámetros de ajuste evaluados y se exponen los resultados .
- **Capitulo 6:** ***Conclusiones y Resultados Finales.*** Evaluación de resultados, argumentos de conclusión y trabajos futuros.
- ***Anexos.*** Archivos extras.
- **Acrónimos** Una sección que incluye un listado de los acrónimos utilizados en la redacción del documento.
- ***Bibliografía.*** Se incluye toda la documentación utilizada.

Capítulo 2

Web Semántica

2.1. Introducción a la Web Semántica.

En la actualidad, el desarrollo de tecnologías de la información y la comunicación cada vez más accesibles para las personas, ha hecho posible un incremento masivo de contenidos disponibles en Internet que pueden ser accedidos y consumidos desde una amplia variedad de dispositivos tales como computadoras, celulares, televisiones, y en general, cualquier dispositivo que tenga acceso a la red, e igualmente por un amplio rango de usuarios. Sin embargo, en su mayoría, el contenido publicado está pensado para ser interpretado única y exclusivamente por humanos debido a que está representado en forma de texto plano, lo que ocasiona que al realizar una consulta, los motores de búsqueda realicen sus operaciones relacionando palabras clave incluidas explícitamente en el código *HyperText Markup Language* (HTML) de las páginas [5]. Esta característica, propicia que la localización de información relevante para los usuarios y que su relación e integración se convierta en una tarea de alta dificultad para los ordenadores, como se afirma en [11].

Estas limitaciones, se traducen en una serie de problemas que se manifiestan al momento de usar la web para la realización de consultas sobre un contenido específico, entre las cuales se pueden mencionar [12]:

- *Recuperación excedente de resultados y baja precisión*: Una búsqueda puede arrojar un número indiscriminado de resultados, siendo la gran mayoría de estos irrelevantes para el usuario ya que las palabras clave pueden coincidir en miles o incluso millones de sitios aunque estos no tengan relación con el contexto de la búsqueda.

- *Baja o nula recuperación de resultados:* Al contrario que en el punto anterior, una búsqueda puede retornar muy pocos o ningún resultado debido a una mala elección de palabras clave.

- *Alta sensibilidad al vocabulario*: El resultado de una búsqueda puede cambiar drásticamente al cambiar una sola palabra clave, a pesar de que ésta sea semánticamente igual a la anterior.

- *Los resultados son simples páginas web*: Los resultados arrojados por los motores de búsqueda son páginas web de texto plano; si se desea encontrar información relevante es necesario que el usuario la extraiga manualmente. Este proceso en muchos casos puede inducir a una serie de consultas adicionales que pueden resultar tediosas.

Estos problemas se presentan porque, para la máquina, toda la información disponible en la web no representa más que simples cadenas de texto sobre las cuales es incapaz de encontrar relación o significado.

Con la introducción de la web semántica, se extiende la web tradicional y se representa la información de una manera en la que la máquina sea capaz de procesarla y de alguna manera entenderla. Esta representación, contribuye a solucionar muchos de los problemas mencionados anteriormente al hacer que la información pase de ser un simple texto a estructuras interconectadas de datos a las cuales el ordenador ya les puede hallar un significado semántico.

2.2. Ontologías.

2.2.1. Definición.

Según la filosofía, una ontología es una teoría sobre la naturaleza de la existencia y de cómo existen los diferentes tipos de elementos [13]. En la Web Semántica, como literalmente se menciona en [11]: "Una ontología es definida como una especificación de una conceptualización que consiste en un conjunto de conceptos, propiedades y relaciones entre conceptos que pueden existir para un agente o una comunidad de agentes", en otras palabras, una ontología es un documento que formalmente define la relación entre términos de una área de interés. La típica clase de ontología para la web, está formada por una taxonomía y varias reglas de inferencia [13].

2.2.2. Elementos de una Ontología.

Taxonomía. Sirve para definir clases de objetos y sus relaciones entre ellas [13]. Por ejemplo, el contenido televisivo se puede definir como programas de entretenimiento, información o aprendizaje. Todos ellos serán subclases de la clase *programación*, y a su vez, de la clase *entretenimiento* pueden heredar otras clases como *películas*, *series*, *novelas* o *caricaturas*. Así, los noticiarios o programas de deportes serán subclases de la clase *información*; de esta forma se puede seguir dividiendo las clases en subclases cada vez más especializadas hasta tener una granularidad tan fina como sea necesaria para la aplicación que se esté desarrollando.

Todas las subclases heredarán las características de sus clases padres e incluirán sus propios atributos. Las taxonomías representan así, una de las herramientas más poderosas en la web semántica ya que permiten inferir el conocimiento implícito. Por ejemplo, si un usuario señala que le gustan los programas de fútbol, dado que éstos corresponden a una subclase de los programas de deportes, que a su vez son una subclase de los programas informativos, se puede hacer una inferencia y relacionar el fútbol con los canales de programación informativa, sin que en la clase fútbol haya ninguna referencia explícita a la clase información.

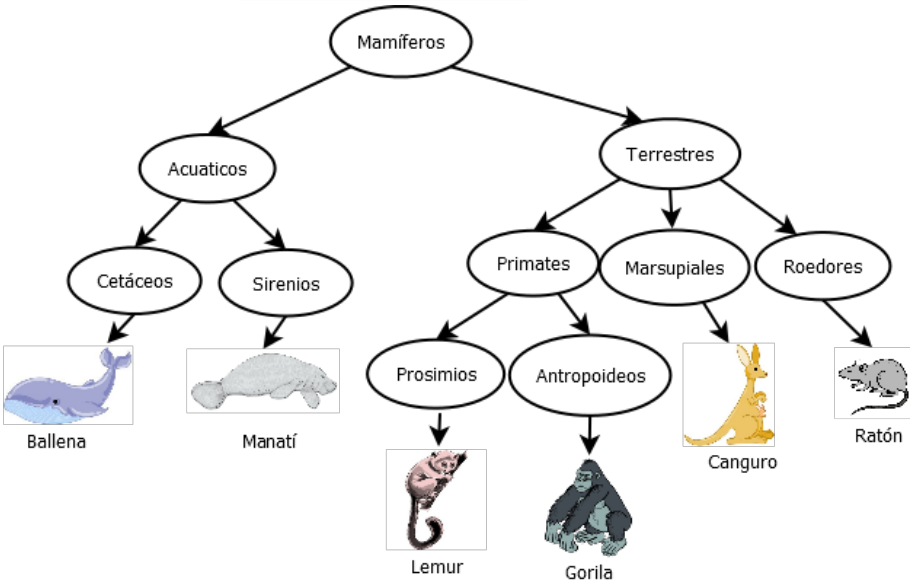


Figura 2.1: *Ejemplo de una taxonomía para la clase mamíferos*

En la figura 2.1 se puede observar otro ejemplo de una taxonomía aplicada a la clase mamíferos en la cual, se divide la clase padre en subclases de mamíferos

acuáticos y terrestres, las cuales a su vez han sido divididas en varias subclases, como se puede en la subclase *primates*, cada subclase puede tener su propia profundidad o nivel de descomposición. Al final se representan instancias de cada clase de la taxonomía. Con estos ejemplos se demuestra que mediante el uso de las taxonomías y las ontologías, se puede definir los tipos de clases, subclases y las relaciones de cualquier elemento existente en la naturaleza, ya sea este real o abstracto.

Reglas de Inferencia. Brindan otra herramienta poderosa para la Web Semántica, ya que pueden encontrar conocimiento implícito relacionando dos o más ontologías de distintos dominios. En los dos ejemplos anteriores se mencionó una ontología de programación televisiva y una segunda ontología de mamíferos; con el uso de las reglas de inferencia, la computadora es capaz de asociar estas dos ontologías para encontrar conocimiento implícito. Suponiendo un escenario en el que un usuario revisa un artículo o una página web sobre gatos, si la página relaciona a los gatos en la ontología de mamíferos, mediante las reglas de inferencia es posible asumir que un mamífero es un animal, y en consecuencia, conectar un concepto de animal en la ontología de mamíferos con un programa de gatos en la ontología de programación televisiva. En otras palabras, las reglas de inferencia sirven para conectar conceptos de dominios distintos e inferir sobre ellos.

Agentes. Son sistemas que utilizan la semántica y pueden inferir e intercambiar información entre sí mediante servicios. La Web Semántica será cada vez más representativa mientras más información, de diferente procedencia, sea recolectada, procesada y utilizada en programas semánticos; cada uno de estos elementos capaces de conectarse a otros programas para compartir información se convierte en un agente de la Web semántica a pesar de que no sean diseñados explícitamente para este propósito.

Un aspecto de vital importancia para los agentes es que sean capaces de intercambiar "proofs" o pruebas escritas en un lenguaje unificador de Web Semántica, el cual describe con detalle la lógica de las inferencias realizadas y es capaz de justificar la respuesta obtenida por un sistema evitando así que un sistema arroje una respuesta que no corresponde al dominio de la búsqueda.

Firmas Digitales. Corresponden a una característica fundamental cuando se trata de intercambiar información entre agentes. Las firmas digitales verifican que la información que se envía o se reciba proceda de fuentes confiables y que esté debida-

mente encriptada, así, los agentes deben ignorar cualquier información proveniente de un origen que no se haya podido verificar [13]

2.3. Definición de la Web Semántica.

El texto *The Semantic Web* de Kashyap y otros [11] menciona que: "La Web Semántica está definida como una extensión de la Web actual en la cual, a la información se le da un significado bien definido, permitiendo que las personas y ordenadores trabajen juntos de una mejor manera".

Este concepto, requiere que la información esté representada en forma de metadatos procesables por la máquina [5], lo cual se consigue mediante la combinación de las siguientes tecnologías:

- **Meta-datos explícitos.** Toda la información debe llevar consigo su significado mediante un apropiado marcado semántico.
- **Uso de ontologías.** Estas describen las relaciones semánticas entre los términos, y fundamentan el entendimiento compartido entre aplicaciones.
- **Razonamiento lógico.** Herramientas automatizadas de razonamiento deben hacer uso de la información brindada por los metadatos y las ontologías [14].

2.3.1. Historia.

Desde el principio de la década de los 80s se han realizado importantes investigaciones sobre sistemas expertos para mejorar el resultado de las búsquedas realizadas en Internet; éstos trabajos fueron presentados al principio como algoritmos de inteligencia artificial. Más adelante, a finales de los 90s y principios del nuevo milenio, se empiezan a mencionar los primeros algoritmos basados en búsquedas semánticas. Hasta que, en 2001 Tim Berners-Lee publica un artículo en el que propone la web semántica como una extensión de la web tradicional [15].

En 2004 Pat Heyes publica y describe un marco para la descripción de recursos en la web llamado *Resources Description Framework* (RDF), el cual no es más que un simple lenguaje para crear afirmaciones acerca de proposiciones. En este lenguaje, se define un recurso denominado *tripleta*, que consta de un sujeto, un

predicado y un objeto. Este marco después fue extendido a un lenguaje más formal de descripción llamado OWL por Chris Welty en el mismo año.

La figura 2.2 muestra la evolución de la integración de la web semántica desde el año 2007 hasta el 2010. Desde entonces la nube de datos relacionados se ha vuelto demasiado extensa como para poder representarla en un simple gráfico.

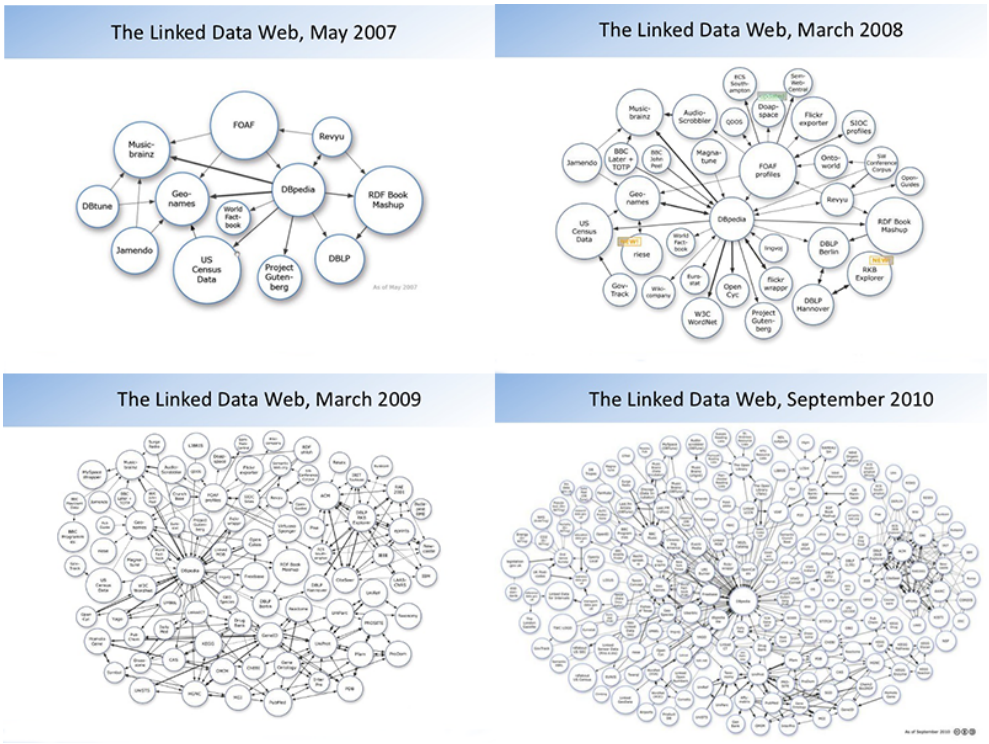


Figura 2.2: *Evolución de la web Semántica.* Fuente: *[6]*

2.4. Arquitectura de la Web Semántica.

La arquitectura de la web semántica ha pasado por un continuo proceso de adaptación basado en las necesidades de los usuarios y los constantes cambios en el entorno tecnológico. En la literatura se has descrito varias propuestas como la presentada en [16], que muestra un enfoque realista de la web semántica en la que una arquitectura de pila única, como tradicionalmente se presentaba, no es suficiente para abarcar las tareas semánticas que pueden presentarse en un futuro. También se pueden encontrar propuestas arquitectónicas más abstractas y simplificadas como explica [11] mediante un ejemplo de la aplicación de la web semántica para un caso médico.

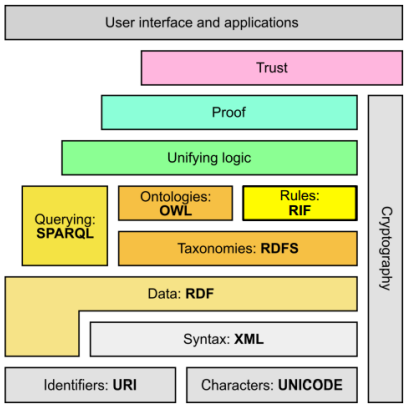


Figura 2.3: *Arquitectura de la Web Semántica* Fuente: *[7]*

En este documento, se presenta la arquitectura adoptada por la *World Wide Web Consortium* (W3C)[1], y se explica brevemente cada uno de sus elementos, para posteriormente estudiar más detalladamente aquellos de mayor relevancia.

- **UNICODE.** Es el conjunto de caracteres internacionales estándar usado en la web semántica.
- ***Uniform Resource Identifier* (URI).** De sus siglas en ingles, corresponde a un Identificador Uniforme de Recursos, que como su nombre lo indica, sirve para identificar los recursos y conceptos disponibles en la Web.
- **XML** Es un lenguaje de marcado extensible usado para compartir recursos a través de Internet o entre aplicaciones.
- **Namespaces.** Es un contenedor abstracto que agrupa uno o mas identificadores y sirve para evitar conflictos entre los URIs definidos por los usuarios.
- **RDF** Es un marco de descripción de recursos basado en tripletas las cuales contienen un sujeto, un predicado y un objeto.
- **Taxonomias RDFS.** *Resources Description Framework Schema* (RDFS) o RDF esquema es una extensión semántica de RDF que introduce los elementos básicos para la descripción de vocabularios.

[1]http://www.w3.org/

- **Ontologías OWL.** Una ontología es un lenguaje de representación que formaliza la conceptualización compartida de un dominio en particular [14], descrito por OWL.
- **Reglas RIF.** Formato de intercambio de reglas o *Rule Interchange Format* (RIF), crea compatibilidad entre las distintas reglas como RDF, OWL y RDFS.
- **Lenguaje de Consulta SPARQL.** Sirve para realizar consultas sobre las tripletas a los repositorios de datos semánticos.
- **Lógica de unificación** Reúne las diferentes ontologías y lenguajes de reglas. También realiza inferencias comunes en el significado de los datos.
- **Proof.** Explica los resultados de la inferencia y la procedencia de los datos.
- **Trust.** Confianza en que el sistema pueda trabajar correctamente y sea capaz de explicar lo que está haciendo, se pueden incluir redes de confianza entre varias fuentes de datos.
- **Encriptación.** Módulo que asegura la integridad y seguridad de los datos que transita por Internet, muchas veces va acompañado de un modulo de firmas digitales.

En esta arquitectura los niveles o módulos inferiores representan los repositorios de datos y los formatos en los que se encuentran almacenados; los módulos intermedios sirven para la interconexión y el razonamiento sobre los datos, y finalmente los módulos superiores interactúan directamente con los usuarios.

2.5. Aplicaciones de la Web Semántica.

El principal objetivo de las tecnologías semánticas es establecer la relación y la inferencia sobre los datos, por lo que su rango de aplicación puede ser muy amplio en el mundo tecnológico actual. Por ejemplo en [17] se muestra un listado de aplicaciones que hacen uso de la Web Semántica donde se puede encontrar desde sistemas en los que es posible realizar una búsqueda utilizando el lenguaje natural, hasta sistemas capaces de aprender las preferencias de un usuario utilizando conexiones y grafos, para luego recomendar ciertos contenidos de acuerdo al área de aplicación del sistema (que corresponde a la categoría en la que se encuentra el

trabajo documentado en este texto); sistemas capaces de manejar planes de viajes, alimentarios o de ejercitación, etc.
Adicionalmente, en la literatura se pueden encontrar muchos más ejemplos de aplicaciones de la Web Semántica; un ejemplo de una aplicación semántica en el campo de la medicina se menciona en [11], y en [18] se utiliza estas tecnologías en el ámbito del turismo. Puede concluirse que las tecnologías semánticas pueden ser usadas en cualquier área en donde se requiera una interconexión de datos e inferencias entre ellos.

2.6. Estándares y Tecnologías de la Web Semántica.

Los estándares que se utilizan en la Web Semántica para el intercambio de información son especificados por el consorcio **W3C** `http://www.w3.org/` el cual es el ente de regulación y publicación de estándares y formatos para la Web.

2.6.1. RDF Schema

RDF provee un lenguaje simple para expresar sentencias sobre algún recurso, sin embargo no tiene la capacidad de representar los vocabularios necesarios en la Web Semántica que indican qué tipo de clase o recurso está siendo descrito [11]. RDFS o RDF Schema es una extensión de RDF que proporciona un vocabulario de modelado de datos para RDF, es decir, RDFS provee un mecanismo para describir grupos de recursos relacionados y las relaciones entre esos recursos, como se menciona en [19], para lograr esto, RDFS describe los recursos usando clases y propiedades, que, a diferencia de los conceptos de la programación orientada a objetos, en la cual una clase está definida por las propiedades que pueden tener sus instancias, RDFS define una propiedad en base a las clases del recurso al que se aplican. Por ejemplo, en RDFS se puede definir una propiedad *autor* la cual está en el dominio *documento* y en el rango *persona*, mientras que en los sistemas orientados a objetos se define una clase *documento* con una propiedad *autor* de tipo *persona*. A continuación se describe los componentes de RDFS.

Clases

Un aspecto básico en un proceso de descripción es poder diferenciar los tipos de recursos que existen. RDFS denomina a esos tipos de recursos como clases, en otras

palabras, un recurso puede ser dividido en varios grupos o tipos a los cuales se les conoce como clases. Un miembro de una clase es conocido como una instancia de una clase a la que se la identifica con una *Identificador Internacional de Recursos* (IRI) y se la describe con sus propiedades.

Una clase en RDFS se define como `rdfs:Class`. Si una clase A es subclase de una clase B, todas las instancias de A también son instancias de B; para definir una clase como una subclase de otra, se usa la propiedad `rdfs:subClassOf`. El término `rdfs:superClassOf` indica lo contrario, es decir, que la clase en cuestión es padre de una subclase de ella [19].

- `rdfs:Resource.` Cualquier elemento descrito en RDF es un recurso. La clase `rdfs:Resource` es la clase del todo, todas las demás clases son `rdfs:subClassOf` de la clase Resource.
- `rdfs:Class` Es un grupo de recursos que forman una clase.

Propiedades

Como ya se mencionó, las propiedades sirven para describir a las clases y los atributos que las caracterizan. Las propiedades se describen usando la clase RDF `rdf:Property`, que es a su vez una instancia de la clase `rdfs:Class`, y está definida dentro de un dominio `rdfs:domain` y un rango `rdfs:range`. Al igual que las clases, una propiedad puede ser subpropiedad de otra, para especificar esto se utiliza `rdfs:subPropertyOf` [11].

2.6.2. Web Ontology Language - OWL

OWL es un lenguaje que permite que la máquina sea capaz de procesar la información contenida en la Web. Corresponde al siguiente paso de los lenguajes RDFS, RDF o XML, en donde los recursos pueden ser únicamente representados pero no procesados [20], OWL puede representar los recursos, el significado de los términos y en vocabularios y las relaciones entre estos términos con una semántica mayor que XML, RDF o RDFS. Se han desarrollado tres sublenguajes OWL con capacidades expresivas incrementales, cada uno de ellos pensado para las necesidades de las comunidades ejecutoras y los usuarios [11].

OWL Lite

Brinda soporte para los usuarios que tienen la necesidad de clasificaciones jerárquicas y restricciones simples. Puede soportar una *cardinalidad*[2] únicamente de 0 o 1. *OWL Lite* proporciona una ruta rápida de migración para *tesauros*[3] y taxonomías [20].

OWL DL

Está diseñado para usuarios que requieren máxima expresividad garantizando que las conclusiones sean computables y finitas, es decir que se pueda encontrar una conclusión en un periodo de tiempo finito. OWL DL contiene todas las expresiones de OWL, RDF, XML o RDFS aunque se aplican restricciones en cuanto a su uso, por ejemplo una clase puede ser subclase de muchas clases pero no puede ser instancia de otra.

OWL Full

Está pensado para usuarios que requieren una máxima expresividad aunque no se garantice la computabilidad y la finitud de las conclusiones. OWL Full permite extender vocabulario preestablecido (RDF, RDFS, XML) aunque es muy poco probable que un software sea capaz de razonar con las características completas de OWL FULL

2.6.3. Extendible Markup Language - XML SCHEMA

XLM Schema describe la estructura de un documento XML [21], por lo tanto, no es específicamente un lenguaje de representación de ontologías sino que tiene un uso más general pudiendo definir la estructura de cualquier documento XML. XML Schema se utiliza para definir una clase de documentos XML por lo que en muchos casos se utiliza el término "Documento Instancia" para describir un documento XML se que adjunta a un esquema predeterminado [11].

[2]Cardinalidad: En las relaciones, la cardinalidad representa el número de veces que una entidad A aparece asociada a una entidad B

[3]Tesauros: lista de palabras, términos o reglas utilizadas para representar un concepto o un lenguaje.

XML Schema define:

- Los elementos que pueden aparecer en el documento.
- Los atributos que pueden aparecer en el documento.
- La estructura jerárquica de los elementos.
- Si un elemento está vacío o puede incluir texto.
- El tipo de datos para los elementos y los atributos.
- Valores predeterminados y fijos para los elementos y atributos.

Al estar escrito en XML, XML Schema es extensible permitiendo que se puedan reusar esquemas XML, crear tipos de datos propios derivados de tipos de datos existentes, referenciar múltiples esquemas en el mismo documento. Además, XML define reglas de escritura más estrictas que en un documento XML normal, es decir, no basta con que un documento esté bien formado, sino que todo documento XML Schema debe seguir las siguientes reglas [21]:

- Debe comenzar con la declaración de XML.
- Debe contener un único elemento raíz (root).
- Son *case sensitive*, es decir que diferencia entre mayúsculas y minúsculas.
- Todos los elementos deben ser cerrados.
- Todos los elementos deben estar apropiadamente anidados.
- Se deben utilizar entidades para caracteres especiales.

En definitiva, OWL usa XML Schema ya que es un lenguaje que brinda todas las características necesarias para la descripción de recursos, clases y atributos, al mismo tiempo que ofrece la flexibilidad necesaria para hacer uso de su máximo potencial al momento de describir la relaciones.

Capítulo 3

Sistemas de Recomendación de Contenidos

3.1. Introducción.

Como se mencionó en el Capítulo 2, en la actualidad el usuario promedio se enfrenta a una abrumadora cantidad de información disponible en Internet y otros medios de difusión, lo que en la mayoría de los casos, puede llegar a convertirse en una desventaja y desmejorar la experiencia del usuario en su búsqueda de alternativas de contenido que se ajuste a su perfil y preferencias. Estudios de la conducta humana, como el que se presenta en [22], han demostrado que al tener demasiados recursos de donde elegir, existe una muy alta probabilidad de que el usuario se sienta menos satisfecho con sus decisiones. Es así, que la sobrecarga de información puede llegar a entorpecer un proceso de búsqueda, por lo que han surgido diferentes iniciativas y se han centrado esfuerzos por brindar una solución que permita filtrar el contenido proveniente de de los medios de difusión de información para hacer llegar al usuario exclusivamente la información que se estima es de mayor utilidad para un usuario específico. Al conjunto de estas soluciones se le ha llegado a denominar "Sistemas de Recomendación" o SR como lo indica Kantor en [23].

3.1.1. Historia

Como se menciona en [24] , alrededor del año 1992 varias bibliotecas tradicionales de Estados Unidos empiezan a afrontar el problema de la *sobrecarga de información* mediante la generación de servicios de difusión selectiva (divulgación de información hacia segmentos diferenciados por valores, preferencias o atributos), mediante los cuales de acuerdo al perfil de usuario previamente suscrito al servicio,

se podrán generar periódicamente alertas que podrían ser de su interés [24].

A pesar de que la web tiene características propias que la diferencian de las bibliotecas, en ella se presentan problemas muy similares relacionados al manejo de la sobrecarga de información. Con el objetivo de solucionar, o al menos atenuar estos problemas que se han enfatizado con el crecimiento en los índices de penetración de Internet y los sistemas de telecomunicaciones, se ha planteado iniciativas similares a aquellas de inicios de los noventas que se aplicaron a las bibliotecas, englobándose en lo que hoy en día se conoce como los sistemas de *filtrado de información* o *sistemas de recomendación* SR. Estos sistemas, analizan los recursos disponibles en la web, generalmente en formato XML para facilitar a los usuarios la recuperación de información de su interés. Adicionalmente, estos sistemas se utilizan ampliamente en la predicción de la valoración o *rating*[1] que un usuario otorgaría a ciertos ítems con el objetivo de medir cuánto le podría llegar a interesar y ampliar la gama de productos que se le pueden ofertar. Este punto de partida para los sistemas de recomendación también se relaciona con la aparición de los *newsgroups*[2], que son servicios de filtrado noticias creados para que los usuarios accedan exclusivamente a las noticias de su interés, y optimizar su tiempo en lo que coloquialmente se conoce como "contenido basura". La necesidad de filtrado de contenido crece a la velocidad en la que se generan los mismos

Otro aspecto interesante de los sistemas de recomendación es que se han utilizado ampliamente por grandes empresas para poder promocionar mejor su información, y un ejemplo de ello, es su aplicación en el escenario de la televisión digital. Los autores Luigi Ceccaroni y Xavier Verdaguer [25] proponen un sistema de recomendación de televisión digital orientado principalmente a televidentes convencionales pero también a usuarios de computadores personales y móviles; ellos sostienen que el uso de estos sistemas permite capturar el interés del usuario.

Actualmente los sistemas de recomendación se han desarrollado de tal manera que su uso se puede encontrar en una amplia variedad de sitios y servicios, especialmente en el ámbito del comercio electrónico. Un ejemplo claro de esto es la tienda en línea *Amazon*[3], usa un SR basado en el historial de navegación y compras de

[1] El término *rating* se utiliza ampliamente hoy en día para describir la calificación de un ítem en múltiples contextos, como en películas, libros, etc

[2] Newsgroups es un término anglosajón que significa grupos de noticias.

[3] www.amazon.com

cada usuario. Hoy en día, se constituyen como una herramienta fundamental para los proveedores en línea, como se sugiere en [26].

3.1.2. Definición

Un sistema de recomendación puede ser definido como un sistema capaz de "aprender" las preferencias de un determinado usuario o grupo de usuarios con el objetivo de filtrar el contenido que recibe desde una fuente de información acorde a sus intereses; más adelante en esta sección se define de una manera formal un sistema de recomendación de contenidos.

3.1.3. Problema de Fondo

El problema de generar una recomendación se puede modelar de la siguiente manera: Suponiendo que $U = (u_1, u_2, u_3...u_n)$ es el conjunto de todos los usuarios suscritos al sistema de recomendación y que $I = (i_1, i_2, i_3...i_n)$ son todos los ítems a recomendar, entonces se trata de tener una función $f(u_m, i_n)$ que mida la utilidad o el grado de interés que produce el ítem i_n para el usuario u_m. Ésta utilidad debe ser medida para cada usuario u, comparado con todos y cada uno de los ítems i, luego de esto se obtiene un vector ordenado de medidas también conocidas como *ratings*, de acuerdo al tipo de aplicación se pueden recomendar los n primeros ítems con mayor utilidad para un usuario dado. Para lograrlo, es necesario que el usuario se registre e ingrese en el sistema sus gustos explícita o implícitamente con el fin de que las preferencias almacenadas en el sistema en forma de características permitan la creación de un perfil de usuario.

Además de los usuarios, cada ítem debe encontrarse registrado en el sistema con sus propias características, todo esto de acuerdo al contexto de cada aplicación. En este escenario la principal dificultad consiste en encontrar la función adecuada f para obtener la utilidad de cada ítem i_n, con respecto a un usuario u_m. Cabe resaltar que la solución a este problema se encuentra en un subconjunto de todas las combinaciones posibles de usuarios e ítems.

3.2. Tipos de Sistemas de Recomendación.

Los sistemas de recomendación tanto en aplicaciones prácticas como en la literatura pueden categorizarse de acuerdo a la forma en la que manejan la información.

En particular, Kantor [23] distingue fundamentalmente dos tipos de sistemas de recomendación:

- Sistemas de recomendación basados en el contenido
- Sistemas de recomendación colaborativos o sociales

Adicionalmente, la literatura suele contemplar la sinergia de estas disciplinas para definir enfoques híbridos que combinan las mejores características de cada tipo de sistema, según se describe en [27].

3.2.1. Sistemas de Recomendación basados en el contenido

Estos sistemas clasifican los ítems de acuerdo a sus características y a las preferencias de cada usuario, sin tomar en cuenta las preferencias de otros usuarios del sistema. Por ejemplo, si un usuario denota interés en las películas de acción y una película tiene marcada en sus características que pertenece al género de acción, este ítem tiene una probabilidad mayor de ser recomendado por el sistema.

Para la evaluación de un ítem, el sistema se basa en calificaciones anteriores que el usuario realizó sobre ítems con características similares, cada una de ellas, responde a un esquema de ponderación y tiene asignada un peso respectivo, proporcional a su importancia relativa. Para la descripción de un ítem generalmente se usan tags (etiquetas) o palabras clave.

En la estimación de la predicción de la valoración de un ítem, los sistemas de recomendación basados en el contenido usan mayoritariamente modelos heurísticos, técnicas de *machine learning* o aprendizaje de máquina, estadística, modelos de aprendizaje y redes neuronales, como se indica en [28].

En la figura 3.1 se puede observar el funcionamiento de un sistema de recomendación basado en contenidos, en el cual se representa a los ítems de contenido similar con un mismo color de fondo, al ingresar al sistema de recomendación este filtra los ítems basándose en las preferencias del usuario y recomienda únicamente los de su interés.

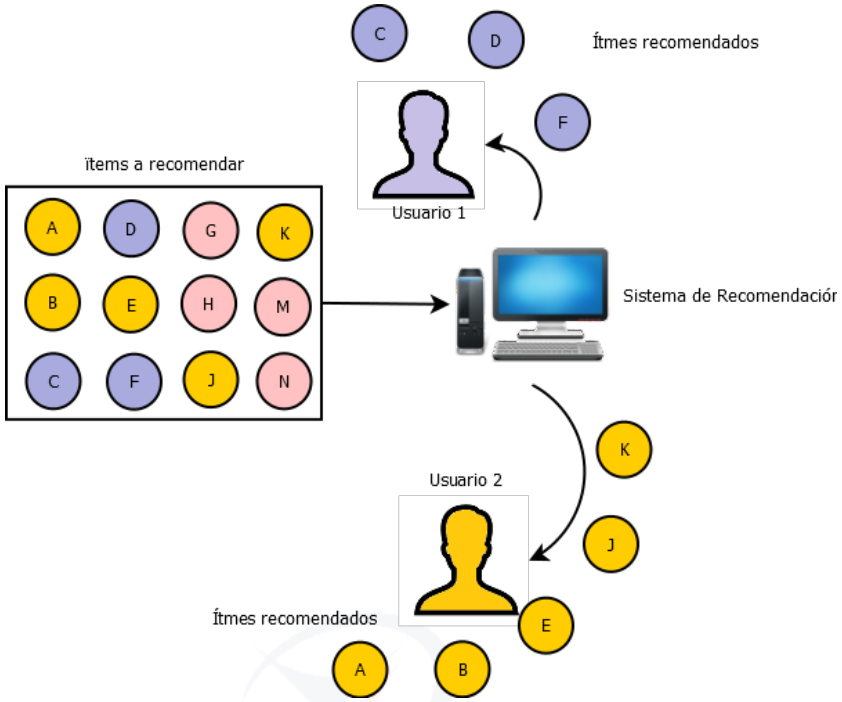


Figura 3.1: *Sistema de recomendación basado en contenidos.*

Limitaciones

A pesar de ser una herramienta poderosa y efectiva al momento de realizar las recomendaciones, los sistemas de recomendación basados en contenido están expuestos a una serie de limitaciones que se listan a continuación:

1. **Análisis restringido de contenidos.**
   Los sistemas de recomendación basados en contenidos están restringidos a las características explícitamente asociadas con los ítems a ser recomendados, por lo tanto, la efectividad de estos algoritmos depende de que tan bien estén descritos los ítems en el sistema. Es decir que para lograr una buena recomendación es necesario tener muchos *meta datos* asociados al ítem [3].

2. **Sobre-especialización del contenido.**
   Estos sistemas únicamente recomiendan ítems similares a los ya valorados por el usuario, así por ejemplo un usuario sin experiencia en películas de drama, nunca obtendrá una recomendación de una película de drama[3]

3. **Arranque en frío (problemas con los nuevos usuarios.)**
   Cuando se registra un usuario nuevo, el mismo tiene que valorar muchos ítems antes de que el sistema pueda hacer recomendaciones acertadas y cercanas a

sus intereses. Al principio, el usuario recibirá recomendaciones pobres debido a la pocas valoraciones otorgadas a otros ítems, como se estudia en [3] y [29].

3.2.2. Sistemas de Recomendación colaborativos

Para realizar las recomendaciones, estos sistemas agrupan a los usuarios con preferencias similares. La utilidad de un ítem es calculada basándose en las calificaciones de los usuarios pertenecientes al grupo del usuario activo, por lo tanto, estos sistemas utilizan la información de usuarios con características semejantes dándole muy poca importancia al contenido de un ítem [30]. Los ítems con valoraciones altas serán recomendados a usuarios similares, también conocidos como vecinos cercanos. Este es el enfoque conocido como basado en usuarios: *user-based collaborative filtering.*

En la figura 3.2 se muestra el ejemplo de una matriz para encontrar la similaridad entre dos usuarios, en la cual, se coloca en las columnas los usuarios a comparar y en las filas los ítems valorados por cada usuario; a partir de esto, se toma en cuenta únicamente a los ítems que hayan sido previamente calificados por ambos usuarios formando para cada usuario un vector de igual longitud que contiene dichas calificaciones, la similaridad entre los usuarios en cuestión puede ser hallada aplicando a los dos vectores varios métodos matemáticos, siendo la más común el método de la similitud del coseno, ampliamente documentado en [31].

	u_1	u_2
i_1	4	3
i_2	4	
i_3	4	5
i_4	5	3
i_5		4
...	...	...
i_n	4	5

Figura 3.2: *Sistemas colaborativos basado en usuarios*

Este proceso se realiza a través de comparaciones entre los usuarios del sistema. Al final se obtiene una matriz que indica el valor numérico de la similaridad de cada usuario con los demás, mientras que el valor de un ítem, se obtiene promediando

las calificaciones que los demás usuarios dieron sobre el ítem en cuestión multiplicadas por su factor de similaridad. Este procedimiento evidencia que este tipo de sistemas no consideran las características propias del ítem analizado[31].

Sin embargo, dentro de este tipo de sistemas también se cuenta con el enfoque basado en los ítems: *item-based collaborative filtering*, en el que el filtrado colaborativo se basa en las similaridades existentes entre los ítems, y se recomienda al usuario aquellos similares a los que demostró interés en el pasado.

Para encontrar la similaridad entre dos ítems se realiza un proceso semejante al que se realiza para encontrar la similaridad entre usuarios, la diferencia es que para este caso, al construir la matriz, se coloca en las columnas los ítems a comparar y en las filas los usuarios que hayan calificado esos ítems. De igual manera, se construye un vector para cada ítem y se procede a calcular la similaridad entre los dos vectores con las técnicas mencionadas anteriormente.

La valoración de un ítem, se obtiene al promediar las calificaciones que el usuario ha asignado a los demás ítems multiplicadas por su valor de similaridad. En la figura 3.3 se muestra la construcción de la matriz para encontrar la similaridad entre ítems.

	i_1	i_2
u_1	4	3
u_2	4	
u_3	4	5
u_4	5	3
u_5		4
...	...	...
u_n	4	5

Figura 3.3: *Sistema colaborativo basado en ítems*

Limitaciones

Al igual que los métodos de filtrado basados en el contenido, los métodos colaborativos presentan una serie de limitaciones que se detallan a continuación:

- **Problema de las valoraciones dispersas.**
  También conocido como *sparcity problem*, se tiene que usualmente las calificaciones disponibles para estos sistemas es mucho menor que la necesaria para realizar buenas predicciones. El éxito de estos algoritmos depende de la disponibilidad de una gran masa crítica. Así, podría darse el caso en que poca gente puede recomendar un mismo ítem y muchos ítems pueden tener pocas recomendaciones; en estos casos habrá ítems rara vez recomendados, como se documenta en [30].

- **Arranque en frío (Problemas con los nuevos usuarios y los nuevos ítems)**
  Se dá el mismo caso que con el sistema basado en contenido, la diferencia radica en que para estos sistemas, el problema puede darse no únicamente para nuevos usuarios sino también para nuevos ítems.

- **Problema de la oveja gris.**
  Se presenta cuando el sistema no genera recomendaciones adecuadas para personas que estén en la frontera de dos tipos de preferencias o que tengan gustos muy dispersos.

- **Efecto portafolio (Poca diversidad)**
  En este caso, el sistema únicamente recomendará los ítems más populares, lo que ocasionará que los ítems nuevos que se ingresen al sistema, tendrán dificultades para ser recomendados, así sea la mejor opción para el usuario.

Ejemplo de filtrado colaborativo

En la figura 3.4 se muestra un ejemplo de filtrado colaborativo, utilizando en este caso la técnica basada en ítems, para lo cual, como se explica en el método, se colocan en las columnas los ítems a recomendar y sobre las filas las calificaciones otorgadas a estos ítems por diferentes usuarios. Al compartir la coloración en azul se indica que el usuario activo tiene una preferencia sobre el ítem E, luego, se comparan las calificaciones de todos los ítems encontrando que los ítems B y D presentan una mayor similaridad con E en las calificaciones otorgadas por los usuarios del sistema, por lo tanto, dichos ítems se convierten en candidatos para la recomendación.

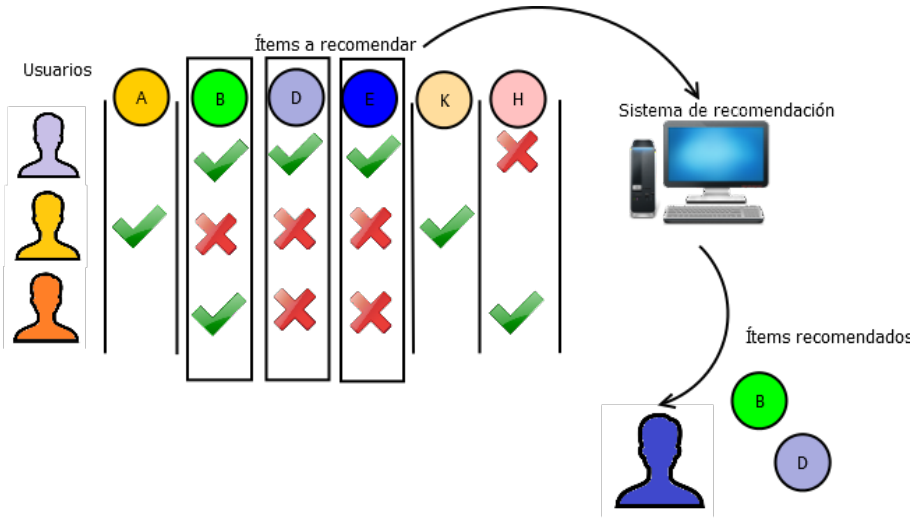


Figura 3.4: *Sistema de recomendación colaborativo basado en items*

En la figura 3.5 se muestra un ejemplo de una matriz de similaridad para un sistema de recomendación colaborativo basada en usuarios; se ha colocado en las columnas a los usuarios del sistema y en las filas los ítems; el usuario en azul representa el usuario activo; como se observa, el usuario denominado como *U2* presenta una mayor similaridad en cuanto a las calificaciones otorgadas a los distintos ítems, por lo que, al predecir una calificación para ítem X para el usuario activo se tomará dicho valor de acuerdo a la calificación asignada por el usuario *U2*.

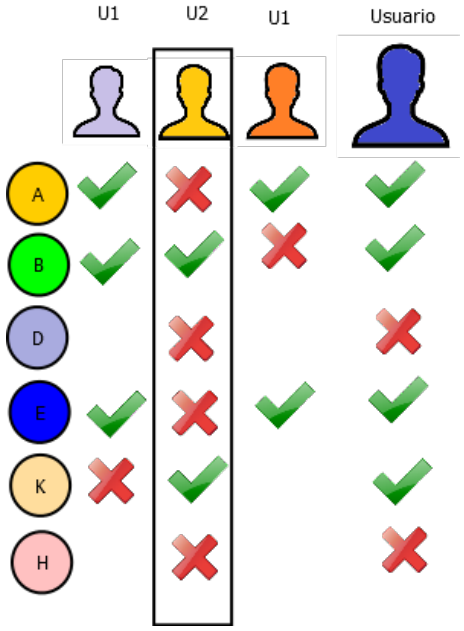


Figura 3.5: *Matriz de similaridad de Usuarios*

Los ejemplos presentados en esta sección han sido únicamente de carácter ilustrativo, puesto que, la creación y el cálculo de las matrices de similaridad tanto para ítems como para usuarios representa una serie de complejos cálculos matemáticos y procedimientos computacionales, además, en sistemas reales dichas matrices están compuestas por miles de filas y columnas, es decir por miles de usuarios e ítems o viceversa.

3.2.3. Sistemas de Recomendación híbridos

Los sistemas de recomendación híbridos combinan las dos técnicas de recomendación estudiadas anteriormente, principalmente para atenuar los problemas de cada una de ellas aplicadas por separado, de esta forma se mejora la eficiencia de la recomendación en términos de rapidez y efectividad. Existen varias maneras de combinar las técnicas de recomendación, a continuación se exponen las principales [3] .

- **Algoritmos de recomendación híbridos ponderados.** En éstos algoritmos la valoración de un ítem se calcula a partir del resultado de aplicar sobre el ítem las dos técnicas de recomendación, y posteriormente, se utiliza por lo general métodos de combinación lineal que permiten obtener la valoración final.

- **Algoritmos de recomendación híbridos mixtos.** Estos algoritmos usan las técnicas de filtrado basado en el contenido usando la descripción textual de un ítem y las preferencias del usuario del filtrado colaborativo. Las recomendaciones de las dos técnicas son combinadas para calcular la valoración final sobre un ítem. Esta técnica es útil para ayudar a mitigar los problemas de inicio frío ya que un ítem nuevo se recomienda según su contenido aunque no haya sido valorado antes por ningún otro usuario.

- **Algoritmos de recomendación híbridos conmutados.** Algoritmos de recomendación híbridos conmutados: Estos algoritmos utilizan algún criterio para conmutar entre las técnicas de recomendación, haciendo que, dependiendo de las características del ítem o del usuario se elija la mejor técnica a aplicar en la recomendación, aunque esto agregue una complejidad extra en cuanto a la estimación de la parametrización que el algoritmo deberá recibir para elegir qué técnica aplicar.

- **Algoritmos de recomendación híbridos en cascada.** En estos algoritmos los ítems son pasados primero por un filtro, ya sea basado en el contenido o colaborativo. Posteriormente, se obtiene un grupo de ítems candidatos, que pasan por un segundo filtro para obtener una mejor recomendación para el usuario.

- **Algoritmos de recomendación híbridos multi-nivel.** Estos algoritmos son similares a los anteriores con la diferencia de que luego de pasar por el primer filtro, se genera un modelo matemático de las recomendaciones. Este modelo sirve luego como entrada del segundo filtro para obtener las recomendaciones.

3.2.4. Otros criterios de clasificación

Además de los tipos de sistemas de recomendación expuestos anteriormente existen otros criterios adicionales para la clasificación de los mismos [32], por ejemplo:

Forma en la que se capturan las preferencias del usuario: En esta clasificación se encuentran los algoritmos **explícitos** en donde se pide que un usuario valore un ítem; y los **implícitos**, en donde se monitorea la actividad de un usuario para calcular la valoración de un ítem.

Según la metodología del filtrado de información: Que puede separarse en dos enfoques, en primer lugar **el filtrado pasivo**, en los cuales se genera una sola recomendación para todos los usuarios del sistema y en segundo lugar, **el filtrado activo**, en donde se genera recomendaciones personalizadas para cada usuario.

3.3. Sistemas de Recomendación semánticos.

Los sistemas de recomendación semánticos o SRS se fundamentan en una base de conocimiento, sobre la cual se hace uso de herramientas de la web semántica para enriquecer los perfiles de usuario a través de el uso de ontologías. Sin importar el tipo de algoritmo de recomendación que se tenga, existe una característica común entre ellos que es el uso de perfiles de usuario para obtener información y

así crear datos de cada usuario que serán usados para generar las recomendaciones. Por lo tanto, es comprensible pensar que los perfiles de usuario se convierten en un componente primordial de un SR para obtener un filtrado eficiente y de mayor calidad. Dado que los perfiles muchas veces son pobres en contenido, este tipo de sistema invoca los componentes de la web semántica para el manejo de la información y así definir procesos de enriquecimiento de dichos perfiles.

Una ontología es un esquema conceptual con un orden determinado basado en etiquetas predefinidas, que abarca uno o varios dominios, con el objetivo de favorecer la comunicación de información entre diferentes sistemas y entidades. Existen varios formatos para describir ontologías tales como RDF (Resource Description Framework o en español Marco de Descripción de Recursos), OWL (Web Ontology Lenguage) entre otros que fueron estudiados en la sección 2.4.

La web semántica (también conocida como Web 3.0) es la evolución de la web actual (Web 2.0), en la que se pretende que la información tenga un significado bien definido y una estructura clara que permita unir conceptos entre sí. Con la web 3.0 se intenta tener una mejor colaboración entre máquinas y humanos [33], buscando que un computador sea capaz de entender, en parte, el lenguaje natural que se usa día a día. Estas ideas se explotan ampliamente en los sistemas de recomendación que utilizan la web semántica, dando paso a la creación de tecnologías que proponen varias soluciones en diferentes ámbitos.

3.3.1. Uso de las Ontologías en los Sistemas de Recomendación

La principal razón para utilizar ontologías es el enriquecimiento de información que puede lograrse. Existen principalmente dos clasificaciones para obtener esto:

- Sistemas basados en redes de confianza
- Sistemas adaptables al contexto

Sistemas de recomendación basados en redes de confianza.

Como se sabe los sistemas de recomendación tiene muchos problemas, sean estos al inicio (arranque en frio) o en el transcurso de su uso, para evitar eso y mejorarlos

se propone usar la unión de una *folcsonomía*[4], es decir se enriquecerá las ontologías con bases de conocimiento con descripciones, categorizaciones, entre otras. Con la unión de estas se creara una "nube" que estará llena de etiquetas donde se usara para construir perfiles mejorados, de esta manera mientras se tenga un mejor perfil mejor será la recomendación [34].

Sistemas adaptables al contexto.

Para mejorar un sistema de recomendación basado en ontología se puede hacer lo siguiente: Analizar y tomar en consideración diferentes factores para inferir el contexto en que se encuentra el usuario de esta manera adaptar la recomendación a las circunstancias (nivel de experiencia del usuario, dispositivo, etc). De esta manera se puede obtener una mejor recomendación ya que se adaptara a cada uno dependiendo su contexto. Por ejemplo el modelo de Kim y Kwon [35] usa cuatro ontologías para adaptar cada circunstancia a cada usuario, este modelo tiene una ontología de productos o ítems, otra donde se definen los diferentes contextos de uso, una tercera sobre el registro histórico de actividades de cada usuarios en este sistema, y una última ontología sobre todos los usuarios.

3.3.2. Modelos que aplican Ontologías en los Sistemas de Recomendación

Existen varios modelos de sistemas de recomendación que incorporan el uso de ontologías para solventar algunas de las limitaciones que se mencionaron en la sección 3.2, entre los cuales se mencionan tres modelos:

- Modelo de Wang y Kong [36].
- Modelo de Khosravi, Farsani y Nematbakhsh [37].
- Modelo Jung y colaboradores[38].

Modelo de Wang y Kong [36].

Es un sistema de recomendación que utiliza una forma ontológica para contra restar el fenómeno del arranque en frío, analizando las características de un ítem en forma ontológica. En este sistema se obtiene una medida similaridad del usuario,

[4]Indexación social

es decir, el sistema encuentra usuarios que tengan gustos parecidos para obtener una recomendación de acuerdo a dicha medida.

La similaridad de cada usuario se mide a través de una media ponderada de tres medidas:

- Similaridad del histórico de evaluaciones de los usuarios.
- Similaridad de datos demográficos.
- Similaridad de interés o preferencia.

Modelo de Khosravi, Farsani y Nematbakhsh [37].

Este modelo utilizado para aplicaciones de comercio electrónico, es un recomendador de productos para clientes cuyo algoritmo utiliza ontologías para describir los ítems y los usuarios con el fin de facilitar el análisis producto-cliente, así, cada ontología estará enlazada entre estos dos. El sistema crea una matriz de evaluación de productos-clientes y su algoritmo se encarga de generar recomendaciones basándose en mecanismo de ponderación y recomendaciones previas. Cabe recalcar que el sistema no realiza ninguna recomendación si el usuario no tiene un cierto número mínimo de compras.

Modelo de Jung [38].

Este sistema tiene una propuesta diferente, la cual se basa en la información personal, los autores de [38] declaran que esto es lo mas adecuado para la web semántica. Este modelo funciona a base de servicios web y perfiles de usuarios con utilización de tripletas RDF, cada ámbito usa un servicio web para recolectar información, donde almacena los consumos anteriores, búsquedas, entre otras cosas. El servicio web convierte los datos de un producto cualquiera en una tripleta RDF. Cuando ya se crearon las tripletas, actúan varios agentes, los cuales son encargados de realizar toda la recomendación; el primer agente extrae los datos del usuario y de los productos, en donde son almacenados en un documento RDF que contiene toda información adquirida. A continuación el agente de recuperación de información extrae las tripletas del perfil de usuario que sean las más relevantes, el siguiente paso es buscar objetos similares, el agente busca coincidencia de tripletas entre usuario y producto, para así llegar a la recomendación.

3.4. Estado del arte de los Sistemas de Recomendación Semánticos.

Con la aparición de la web semántica, muchos sistemas de recomendación basados tecnologías semánticas han sido objeto de análisis y estudio, en esta sección se presenta un recuento de algunos de estos sistemas encontrados en la literatura y se resume brevemente su funcionamiento, orientación y principales características.

3.4.1. Algoritmo de AVATAR [1] [2].

AVATAR es un sistema de recomendación diseñado para un entorno donde el usuario puede recibir y enviar información al proveedor en cualquier instante del tiempo. Su diseño se basa en una arquitectura híbrida, es decir, utiliza tanto la estrategia colaborativa como la basada en contenido, siendo su principal característica, la capacidad para inferir conocimiento combinando las semánticas del contenido televisivo y los perfiles del usuario.

Para la representación del conocimiento los desarrolladores de AVATAR hicieron uso de una ontología que describe el contenido televisivo utilizando el lenguaje OWL la cual almacena clases, instancias y propiedades organizadas jerárquicamente, las cuales identifican a los recursos y relaciones más comunes usados en el dominio de la programación televisiva. Dicha representación se muestra en la figura 3.6.

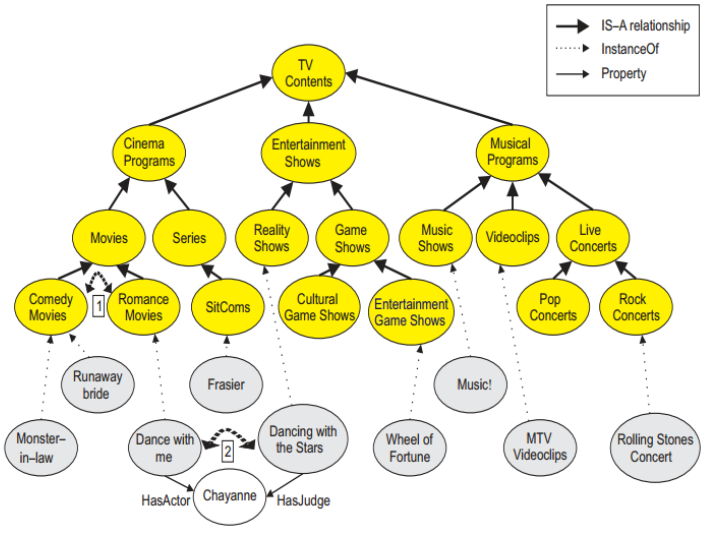


Figura 3.6: *Representación del Conocimiento de AVATAR. Fuente: [2]*

Estrategia de recomendación basada en el contenido.

El modelo de AVATAR utiliza dos métricas para establecer la similaridad entre los contenidos televisivos, en primer lugar se determina la *similaridad jerárquica*, en la cual se identifica qué tan cerca están dos programas en la jerarquía; la segunda métrica utiliza la inferencia para determinar la *similaridad semántica*, que a su vez, sirve para encontrar relaciones entre contenidos televisivos basándose en sus propiedades semánticas como actores, directores, escritores, etc. Al final se evalúan estos dos resultados para obtener un nivel de similaridad entre el programa a recomendar y los programas ya evaluados por el usuario .

Estrategia colaborativa.

El objetivo de esta estrategia es recomendar al usuario programas que hayan sido recomendados a otros usuarios, para ello, AVATAR implementa un mecanismo para encontrar el nivel de similaridad entre los usuarios. Los usuarios más parecidos entre si son conocidos como vecinos cercanos. Para formar las vecindades, se crea un vector para cada usuario que está compuesto por los *Degree of Interest* (DOI) almacenados en cada perfil de usuario, para seguidamente, evaluar dichos vectores de una manera similar a la explicada en la sección 3.2.2. Finalmente el sistema evalúa las preferencias del usuario basándose en las preferencias de sus vecinos más cercanos.

Recomendación Final.

Para brindar la recomendación final al usuario AVATAR implementa un mecanismo de intercambio entre las dos estrategias mencionadas anteriormente. El sistema determina si el contenido a recomendar es similar a los mejores calificados por el usuario y lo recomienda directamente, caso contrario, el usuario es un candidato para el filtrado colaborativo.

3.4.2. Modelo híbrido multi-capa basado en ontologías [3].

En este modelo, los autores proponen un sistema de recomendación de contenidos basado en tecnologías semánticas de cualquier dominio. Al igual que en el modelo de AVATAR, tanto el contenido a recomendar como las preferencias del

usuario están representados mediante ontologías. En esta propuesta, se realizan comparaciones entre los perfiles de los intereses de usuario basados en un concepto o tópico para encontrar similaridades entre dichos usuarios.

A diferencia de los sistemas tradicionales, éste modelo realiza la comparación mediante el particionamiento de los perfiles de usuario en grupos de interés relacionados, y con base a ello, se establecen varias capas de DOI que proporcionan un modelo enriquecido de vínculos interpersonales, representando así de mejor manera la forma en la que la gente encuentra intereses en común. Los intereses de los usuarios se representan como conceptos semánticos en ontologías de dominio, y a continuación se aplica un mecanismo de recomendación colaborativo que tiene en cuenta las similitudes entre los perfiles de usuario.

3.4.3. Modelo de Victor Codina [4].

En este trabajo se presenta un sistema de recomendación semántico adaptativo cuyo dominio está enfocado al turismo, aunque puede ser extendido para trabajar en cualquier otro. Este modelo ha sido pensado para contrarrestar los problemas de las calificaciones dispersas y el arranque en frío. Para el efecto, se han implementado mecanismos que permiten realizar inferencias en información incompleta aplicando inferencias de dominio, lo cual reduce el problema del arranque en frío; al igual que el sistema AVATAR su representación del conocimiento está basado en jerarquías y se usan los mismos criterios de similaridad semántica para encontrar los mejores ítems a ser recomendados para un usuario.

Su arquitectura está basada en el paradigma *Service Oriented Application* (SOA) o Aplicación Orientada a los Servicios, lo que lo convierte en un sistema flexible en cuanto al dominio, ya que para hacer uso del módulo recomendador, solo es necesario consumir su servicio desde una interfaz pública enviándole como parámetro una ontología de dominio compatible pre-definida en formato OWL o RDF. En la figura 3.7 se puede observar el concepto de esta arquitectura donde el servicio recomendador es consumido desde diferentes tipos de dominios.

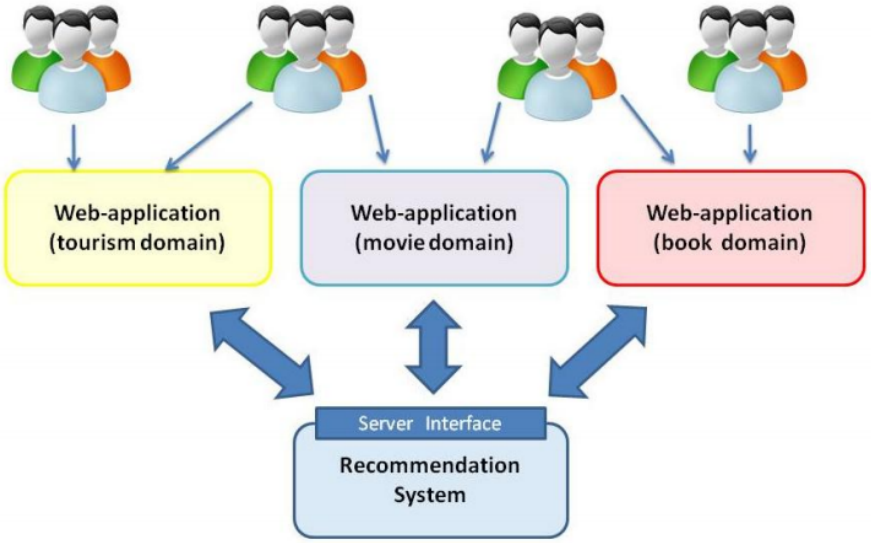


Figura 3.7: *Arquitectura SOA. Fuente: [4]*

Para la adquisición de conocimiento, éste modelo crea perfiles de usuario recolectando información tanto implícita como explícitamente, además, implementa un mecanismo de niveles de confianza sobre los ítems valorados por el usuario. Mediante este mecanismo, el DOI que el usuario otorga a un ítem es multiplicado por un factor de confianza que decrece en función del tiempo, con lo cual se da lugar a la posibilidad de que nuevos intereses puedan ser recomendados a medida que el usuario cambia sus preferencias, en otras palabras, se puede decir que "el sistema crece con el usuario".

Otro aspecto de vital importancia es la implementación de estereotipos para la recomendación: cuando se crea un nuevo perfil de usuario, el sistema analiza su información básica y coloca dicho perfil en un estereotipo correspondiente. La idea principal es que la información del estereotipo complete la información aún desconocida sobre las preferencias del usuario, a medida que el perfil del usuario vaya alcanzando cierta madurez, la información del estereotipo será menos relevante.

3.4.4. Otros Modelos

A más de los trabajos mencionados anteriormente, en la literatura se puede encontrar una gran variedad de sistemas recomendación que hacen uso de las tecnologías semánticas, cada uno de ellos orientados a un dominio diferente o utilizando algoritmos de inferencia diferentes. Puede mencionarse por ejemplo al sistema pre-

sentado en [39] que hace un estudio de los sistemas de recomendación semánticos utilizando árboles de decisiones. Así mismo, se tiene al trabajo presentado en [40] en el que se realiza un sistema de recomendación semántico orientado al dominio de la web; en [41] se presenta un sistema de recomendación de contenidos audiovisuales en donde, a más de presentar recomendaciones para usuarios finales, se hace un estudio de la recomendación para grupos de amigos y familias, en caso de que más de un individuo esté consumiendo el servicio; y para finalizar, se mencionará el trabajo de Juayek [5], en el que se hace una evaluación general de los sistemas de recomendación utilizando para ello una adaptación del sistema de AVATAR. Esta adaptación se utiliza posteriormente en esta tesis para evaluar el funcionamiento de los sistemas de recomendación semánticos, se ha profundizado el estudio de este algoritmo en el capítulo 4.

Capítulo 4

Desarrollo e implementación del sistema

4.1. Introducción.

El presente capítulo realiza una exhaustiva explicación de los algoritmos creados y utilizados en el proyecto, se detalla su funcionamiento, las entradas que recibe, y todas las operaciones elaboradas para lograr los objetivos planteados, entre ellos, el de obtener recomendaciones para un usuario determinado. Adicionalmente, se presentan las herramientas utilizadas en para el desarrollo integral del sistema y la descripción de los conjuntos de datos que sirven para realizar las pruebas del sistema simulando usuarios reales.

4.2. Características del sistema.

4.2.1. Sistema Base

Para la realización y adaptación del sistema de recomendación se ha tomado como base el trabajo realizado por Marcos Juayek Ferreira Pinto y Alejandra Scuoteguazza Mintegui de la Universidad de Montevideo Uruguay documentado en [5], en el cual se evalúan los sistemas de recomendación audiovisuales basados en técnicas inteligentes. Para ello proponen un sistema de recomendación de contenidos basado en el algoritmo del sistema AVATAR [42], el mismo que se estudia más adelante en este mismo capítulo. El sistema propuesto en este trabajo constituye una versión avanzada del sistema de Juayek y Scuoteguazza, la cual surge de una modificación en el algoritmo núcleo y la implementación de una serie de nuevos módulos enfocados a contrarrestar los problemas de los sistemas de recomendación que se mencionan en el capítulo 3. Adicionalmente, en esta tesis se analiza la creación de

un sistema híbrido que utiliza el algoritmo original y una propuesta de calificación mediante la técnica de vecinos cercanos KNN (K vecinos más cercanos).

Modelo del Sistema AVATAR [42]

Como se mencionó en la sección 3.4.1, el sistema Avatar es un SR de contenidos de TV basado en inferencia semántica, el cual encuentra relaciones entre contenidos buscando secuencias en base a las propiedades de las clases ontologías. Por ejemplo, el sistema conecta la película "Rocky" con el reality show "El retador" ya que en los dos casos aparece como actor Silvester Stallone. Las secuencias encontradas, permiten establecer una relación de similaridad semántica entre el contenido a recomendar y las preferencias del usuario que al final es calificada y posibilita obtener un valor para el ítem a recomendar [42].

4.2.2. Herramientas de software utilizadas

Java

Para el desarrollo del sistema de recomendación se utilizó el lenguaje de programación orientado a objetos Java[1] ya que contiene las librerías necesarias para la manipulación de los estándares utilizados en la Web Semántica y además proporciona amplias facilidades a la hora de la creación y publicación de servicios en la Web.

Java es un lenguaje de Programación de propósito general basado en clases, que presenta varias ventajas al estar definido como un lenguaje pensado para que el código escrito y compilado sea independiente del sistema operativo y del dispositivo en el que se ejecute. Además de esto, debido a la gran accesibilidad que presenta al ser un lenguaje de programación libre, Java se ha convertido en uno de los lenguajes más utilizados para aplicaciones de propósito general, en especial para aplicaciones Cliente-Servidor. Adicionalmente, cabe recalcar la robustez que este lenguaje tiene en cuanto al soporte de la comunidad para el desarrollo de cualquier tipo de aplicación.

[1] www.java.com

NetBeans

NetBeans[2] es un *Integrated Development Environment* (IDE) o entorno de desarrollo integrado para varios lenguajes de programación, aunque principalmente se especializa en el lenguaje Java. NetBeans es una herramienta que facilita la gestión de un proyecto, ayudando a organizar los paquetes y archivos, las librerías usadas en el programa, los servicios consumidos, las configuraciones, y además, brinda un soporte integrado para el desarrollo en equipo usando las últimas tecnologías para el control de versiones. Esta herramienta proporciona una plataforma robusta para la ejecución y depuración de un programa, la detección y corrección de errores pre compilación e incorpora además un generador de código para que el desarrollador no pierda tiempo escribiendo códigos comunes como constructores o funciones de Get y Set [3].

Apache Jena

Apache Jena[4] es un marco de desarrollo open source escrito en Java utilizado para la construcción de aplicaciones de Web Semántica y Linked Data[5]. Está compuesta por diferentes APIs que interactúan entre sí para la procesar los datos RDF. Brinda soporte para los estándares publicados por las recomendaciones de la W3C [43].

Entre sus aplicaciones, Jena integra un motor de inferencia basado en reglas y ontologías para realizar razonamiento semántico usando OWL o RDFS, además de varios mecanismos para el almacenamiento de tripletas RDF ya sea en memoria o en disco [5].

MySQL

MySQL[6] es un *Data Base Managment System* (DBMS) o sistema gestor de base de datos de propósito general ideal para aplicaciones pequeñas y grandes. Su motor de base de datos es rápido, confiable y fácil de usar, siendo desarrollado y Soportado por Oracle Corporation[7]. MySQL es un sistema de software libre

[2] www.netbeans.org

[3] las funciones Get y Set se utilizan para obtener y asignar respectivamente el valor de una propiedad de una clase.

[4] jena.apache.org

[5] Se conoce a Linked Data como un conjunto de datasets relacionados disponibles en la web.

[6] http://www.mysql.com/

[7] www.oracle.com

que se distribuye bajo la licencia GNU GPL[44] que ofrece grandes ventajas en cuanto al uso de recursos de hardware y software, ya que al ser una base de datos relativamente liviana, no necesita una gran cantidad de recursos para su correcto funcionamiento. Dado que este proyecto no se centra en el manejo de una base de datos, sino en los sistemas de recomendación, MySQL se ha utilizado como una herramienta auxiliar liviana y de fácil acceso.

OMDB API

The OMDB[8] es un servicio web gratuito y abierto que recibe como parámetro el identificador de un contenido audiovisual y devuelve información relacionada a sus actores, escritores, directores, título, rating, número de votos, géneros, entre otros; ya sea en formato XML o JSon[9] de acuerdo a cómo se especifique.

4.2.3. Arquitectura del Sistema (Modelo de Programación)

El sistema de recomendación ha sido desarrollado de forma modular de forma que resulte sencillo organizar el código agrupando las funciones y las clases de acuerdo al papel que desempeñan. El código entero está divido en seis módulos o paquetes principales. En la figura 4.1 se muestra la arquitectura del sistema desde la perspectiva de la programación.

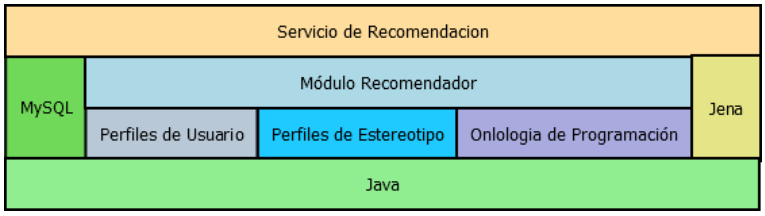


Figura 4.1: *Arquitectura del Sistema desde la perspectiva de la programación*

- **MySQL.** Es el módulo que se encarga de realizar la conexión y las transacciones con la base de datos del sistema, contiene funciones para crear la conexión con el servidor de bases de datos, crear las tablas necesarias e insertar y leer los datos de los perfiles. Éste modulo interactúa directamente con los módulos de los perfiles de usuarios, estereotipo y el módulo recomendador.
- **Jena.** Módulo que se encarga del manejo de la librería *Apache Jena*, permite la manipulación de las ontologías o de los recursos RDF, RDFS o OWL.

[8] www.themoviedb.org
[9] json.org

Realiza las consultas necesarias a las ontologías de perfil de usuario y de programación e interactúa directamente con los módulos de perfiles de usuario, perfiles de estereotipo y con el módulo recomendador.

- **Perfiles de Usuario.** Módulo programado para el manejo de los perfiles del usuario. Contiene funciones para la creación y enriquecimiento de las ontologías de prueba para los usuarios y la lectura y escritura de perfiles de usuario en la ontología.

- **Perfiles de Estereotipo.** De similar funcionalidad que el módulo de perfiles de usuario, se diferencia en que este módulo se encarga de separar a los usuarios en varios grupos de acuerdo a ciertas consideraciones demográficas, cada uno de estos grupos es llamado un estereotipo.

- **Ontología de Programación.** Este módulo realiza la lectura y escritura de la programación televisiva e interactúa con el módulo recomendador.

- **Módulo Recomendador.** Contiene los algoritmos necesarios para realizar la recomendación. Específicamente, es el módulo encargado leer los perfiles de usuario o estereotipo y la programación televisiva, realizar los procedimientos necesarios y emitir una recomendación final destinada a un usuario determinado.

- **Servicio de Recomendación.** Este servicio escucha las peticiones de recomendación e inicia el trabajo de los módulos inferiores. Como resultado, éste módulo devolverá un listado con la programación recomendada para un usuario.

En las secciones posteriores se brindará una explicación profunda de la operación de cada uno de estos módulos así como de las principales funciones involucradas en la obtención de las recomendaciones.

4.2.4. Entradas ontológicas.

Un componente clave del sistema de recomendación semántico está constituido por sus entradas ontológicas, cuya procedencia se define en [42], y que fue implementado por [5]. En el presente proyecto, esta implementación experimentó diversas modificaciones que serán explicadas con detalle más adelante en la sección 4.2.6.

4.2.5. Conjunto de datos.

Para la creación de los perfiles de usuarios se utilizó la base de datos libre de MoviLens[45], cuyas principales características son:

- 6040 usuarios, con un identificador numérico (id) individual en el rango de 1 al 6040.
- 3952 películas, con un identificador numérico (id) individual enumerado entre 1 y 3952.
- Contiene los ratings o calificaciones realizadas por cada usuario a determinadas películas.

Usuarios.

Cada usuario tiene las siguientes características:

1. **Género.** Este campo es denotado por una "**M**" para Masculino y por "**F**" para femenino.
2. **Edad.** El campo edad no está representado por un número, sino que está representado por un rango como se muestra en la Tabla 4.1.

Representación	Rango
1	Bajo 18
18	18-24
25	25-34
35	35-44
45	45-49
50	50-55
56	56+

Tabla 4.1: *Rango de edad de Usuarios.*

3. **Ocupación.** Este campo tiene 20 variaciones, cada una con una representación numérica, la cual se muestra en la Tabla 4.2.

Representación	Ocupación
1	Otro o no especificado
2	Profesor
3	Artista
4	Oficinista
5	Universitario
6	Servicio al cliente
7	Doctor
8	Ejecutivo
9	Granjero
10	Estudiante
11	Abogado
12	Programador
13	Retirado
14	Vendedor
15	Científico
16	Autónomo
17	Ingeniero
18	Artesano
19	Desempleado
20	Escritor

Tabla 4.2: *Ocupación de usuarios.*

Películas.

La información que tiene cada una de las películas es la siguiente:

1. **ID.** Un identificador numérico entre el 1 y el 3952.
2. **Título.** Es el nombre de la película según *Internet Movie Data Base* (IMDB); este campo está en inglés.
3. **Género.** Cada película tiene 3 géneros, los mismos son de IMDB.

Ratings.

La base de datos posee más de un millón de ratings, ya que cada usuario tiene al menos 20 calificaciones realizadas y además, existen usuarios con mas de ochocientas calificaciones. Cada usuario califica una película en un escala entre 1 a 5 estrellas.

4.2.6. Ontología en uso.

Como ya se mencionó anteriormente, el proyecto está basado en AVATAR [42]. Por esta razón se utilizó el mismo modelo ontológico descrito en [8] [42] y [1] , que se explica a continuación:

El perfil-ontológico se crea en OWL (estándar explicado en la sección 2.6.2), el cual, es representado por el dominio ontológico mostrado en la figura 4.2.

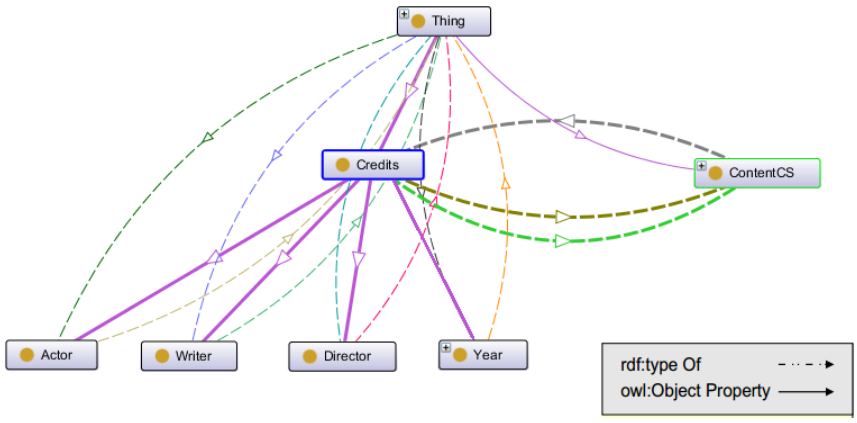


Figura 4.2: *Ontología utilizada en el proyecto.*

En esta figura (4.2) se puede observar cada "Cosa" ("Thing" en la ontología), que es la instancia de un programa televisivo, cuya propiedad es créditos ("Credits"), tiene como sub-propiedades Actor, Director, Writer y Year. Por otro lado está *ContentCS* que es la clase que contiene los Géneros, tal cual se muestra en la figura 4.3.

Para entender mejor esta sección y las relaciones que se adquieren con un perfil-ontológico se presenta el ejemplo en la figura 4.4, donde se puede apreciar cada Clase (representada por un cuadrado de color amarillo). Cada clase tiene varias instancias (representadas con un círculo), las cuales están relacionadas entre varias similares.

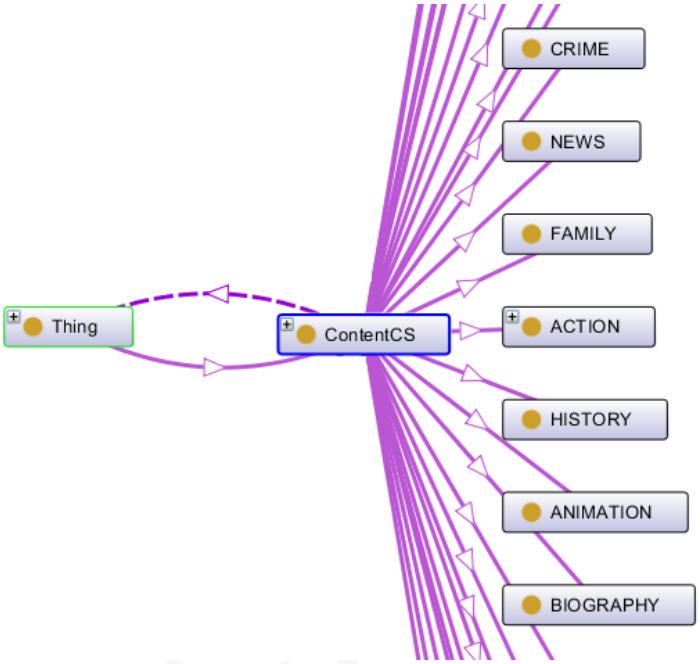


Figura 4.3: *Clase géneros en la ontología.*

Esta relación puede darse por cualquier propiedad que se tenga en común o por su clase padre (en nuestro caso género).

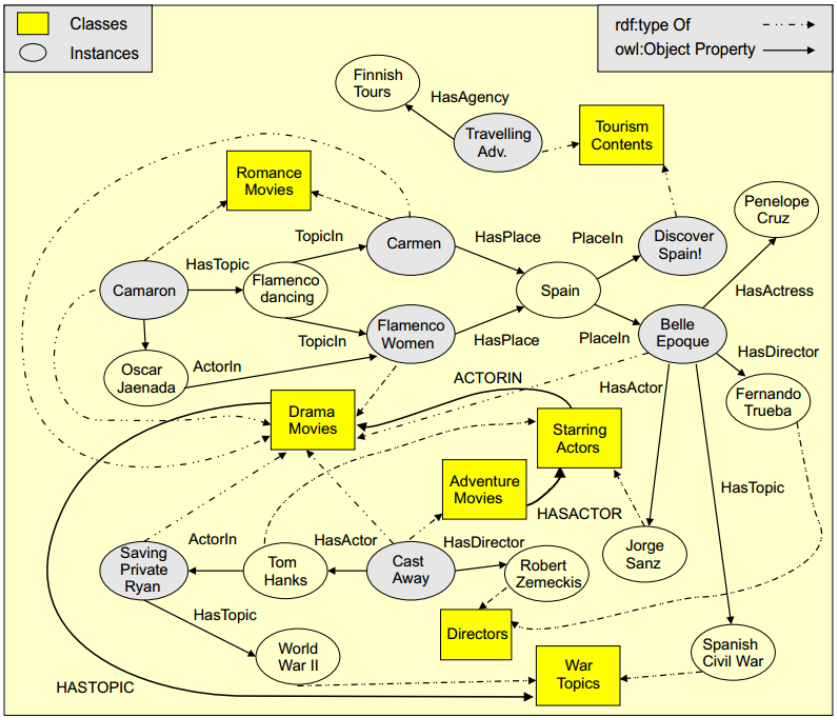


Figura 4.4: *Conjunto de instancias, propiedades y clases en la ontología OWL.[8]*

4.2.7. Creación y enriquecimiento del perfil-ontológico.

Un perfil-ontológico es creado por cada usuario basándose en las películas que haya calificado, es decir, cada película que califique se ingresará en formato de tripleta OWL en su perfil. La creación se realiza de la siguiente manera:

1. Se obtiene los ratings realizados por el usuario de la base de datos de MovieLens (mencionada en la sección 4.2.5).

2. Con el título de cada película se obtiene el ID desde IMDB.

3. El enriquecimiento semántico se realiza a través del servicio web de OMDB (mencionado en la sección 4.2.2). La petición se realiza utilizando como parámetro el ID de IMDB, obteniendo como valor de retorno del servicio web la información relacionada a: título, géneros, elenco completo, lenguajes, etc. Un ejemplo puede apreciarse en la figura 4.5.

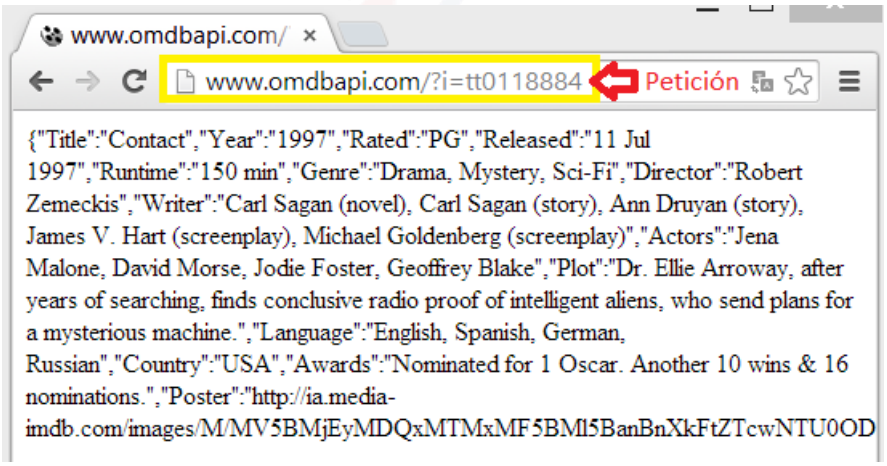


Figura 4.5: *Ejemplo de Api OMDB*

4. Una vez todos obtenidos los datos de la película, se ingresa las propiedades deseadas a la ontología.

5. Para finalizar, el DOI[10] que se ve representado como la calificación numérica otorgada por el usuario, se transmite hacia todas las propiedades asociadas mediante un mecanismo de propagación, que finalmente permite otorgar un DOI a cada una de ellas.

[10]DOI o grado de interés del usuario sobre la película

4.3. Modificaciones realizadas.

Como se mencionó en la sección 3.4.4, el código originalmente realizado por Juayek y Scuoteguazza en [5], se modificó en este proyecto por motivos de mejora de rendimiento y evaluación. Estas modificaciones serán explicadas a continuación:

4.3.1. Cambio Diagrama de Base de datos.

Tras un breve análisis, se determinó que la aplicación inicial de la base de datos (utilizando 4.2.2), era muy básica, ineficiente y limitada para el manejo de grandes volúmenes de datos. En la figura 4.6 se aprecia el modelo inicial, en el que puede observarse que el sistema crea una tabla por cada perfil de usuario que exista con su respectivo ID, y conteniendo únicamente los géneros con su respectivo DOI.

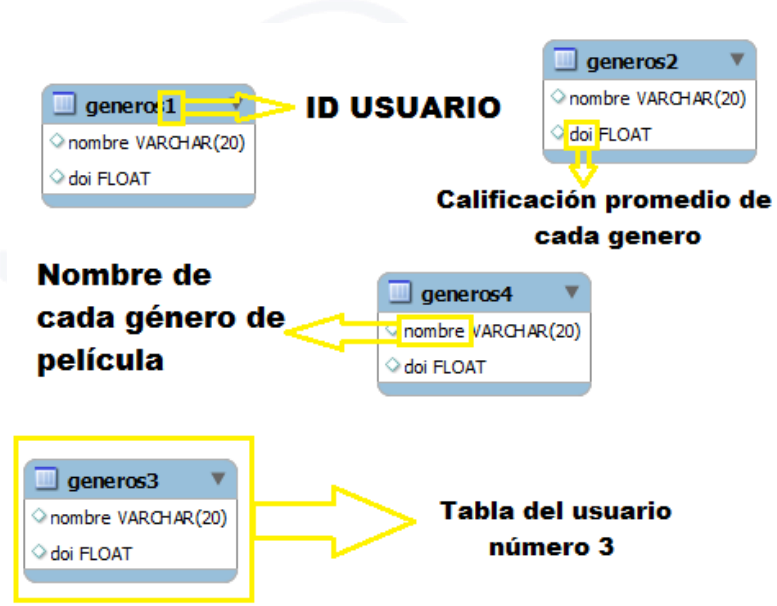


Figura 4.6: *Diagrama Entidad Relación utilizado en [5].*

Ocurría comúnmente un error al usar el diagrama de la figura 4.6, este se daba al tener una gran cantidad de usuarios, ya que se creaba una gran cantidad de tablas, lo cual, entorpecía el rendimiento del Sistema, entre otros problemas.
En este modelo tampoco existe ninguna relación entre tablas ni entidades, y los datos del usuario no se toman en cuenta.

Para evitar todos estos problemas, mejorar la eficacia y rendimiento se creó el Modelo de base de datos que se presenta en la figura 4.7, que contiene en la tabla "USER" a todos los usuarios del sistema con su respectivos campos(ID, Género, edad y ocupación), y que se relaciona con cada tabla de género que exista, siendo ésta una relación *una a muchos*; Cada tabla de género (simbolizadas por un cuadrado amarillo) tiene un ID de usuario con el DOI y número de películas vistas por este.

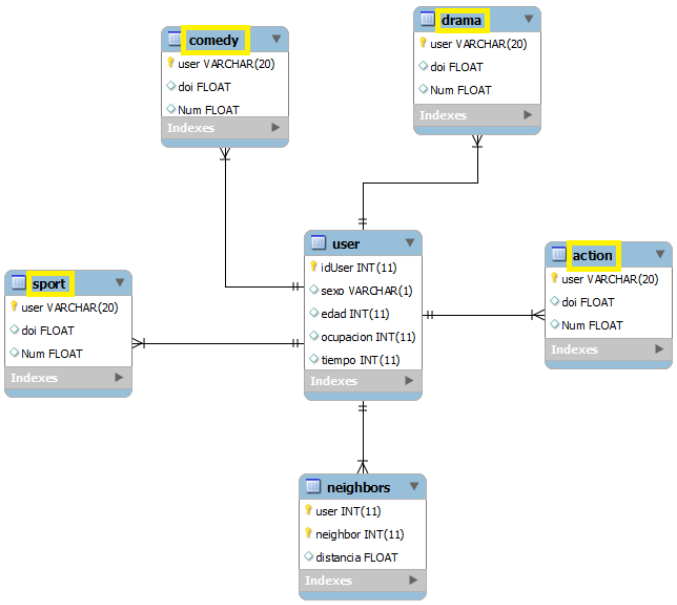


Figura 4.7: *Diagrama Entidad Relación creado en el proyecto actual.*

Por otro lado se creó una tabla llamada "Neighborgs" que se relaciona con "User". Esta tabla fue creada para la aplicación del módulo con el algoritmo KNN (k-nearest neighbors: Algoritmo de K-vecinos cercanos) que será explicada más adelante en la sección 4.6.3.

4.3.2. Restricción de datos por rendimiento.

Al ejecutar el sistema, tanto en la creación de los perfiles como en la predicción, existía un error de *bucle infinito*, el grupo de trabajo percibió que esto ocurría con ciertos programas televisivos que tenían un excesivo número de miembros en su elenco, por ejemplo la figura 4.8, presenta a la película "Pocahontas" con aparentemente más de veinticinco escritores, sin embargo, tras una observación minuciosa, se puede notar que los escritores son únicamente los tres primeros (denotados por el texto *written by*). Esta singularidad suele ocurrir en casos tales como escritores o

Pocahontas (1995)

Full Cast & Crew

Writing Credits

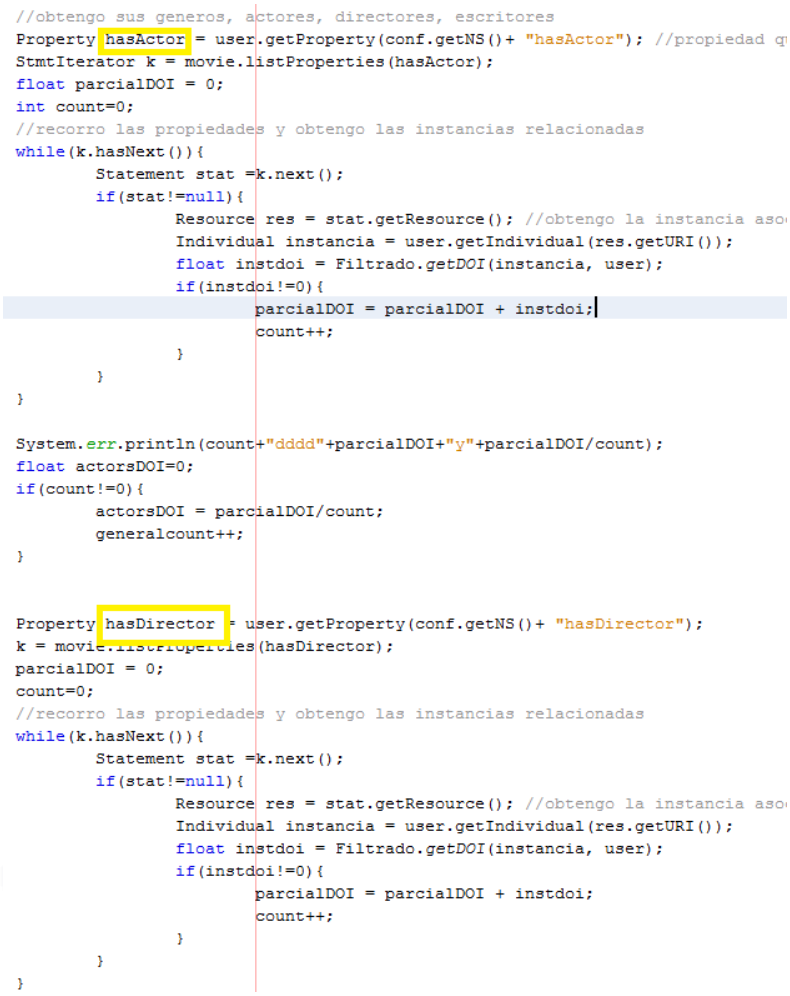
Carl Binder	...	(written by) &
Susannah Grant	...	(written by) &
Philip LaZebnik	...	(written by)
Tom Sito	...	(story supervisor: artistic)
Glen Keane	...	(story) &
Joe Grant	...	(story) &
Ralph Zondag	...	(story) &
Burny Mattinson	...	(story) &
Ed Gombert	...	(story) &
Kaan Kalyon	...	(story) &
Francis Glebas	...	(story) &
Rob Gibbs	...	(story) (as Robert Gibbs) &
Bruce Morris	...	(story) &
Todd Kurosawa	...	(story) &
Duncan Marjoribanks	...	(story) &
Chris Buck	...	(story)
Andrew Chapman	...	(additional story development) &
Randy Cartwright	...	(additional story development) &
Will Finn	...	(additional story development) &
Broose Johnson	...	(additional story development) &
T. Daniel Hofstedt	...	(additional story development) &
David Pruiksma	...	(additional story development) (as Dave Pruiksma) &
Nik Ranieri	...	(additional story development) &
Vincent DeFrances	...	(additional story development) &
Tom Mazzocco	...	(additional story development) &
Don Dougherty	...	(additional story development) &
Jorgen Klubien	...	(additional story development)

Figura 4.8: *Ejemplo de exceso de escritores. Fuente: [9].*

directores. Para evitar esto, se ha decidido limitar a un número máximo de instancias de una propiedad, que en nuestro caso es tres. Esta limitación se implementa en la utilización de la API-OMDB que se explica con detalle en la sección 4.2.2.

4.3.3. Funciones Genéricas.

En gran parte del código inicial existía código obsoleto y repetido, en algunas partes se repetía con pequeños cambios como modificaciones sutiles de ciertos parámetros. Con fines de optimizar el código, se creó funciones genéricas que cumplen funciones similares y pueden ser reutilizadas en diferentes contextos.



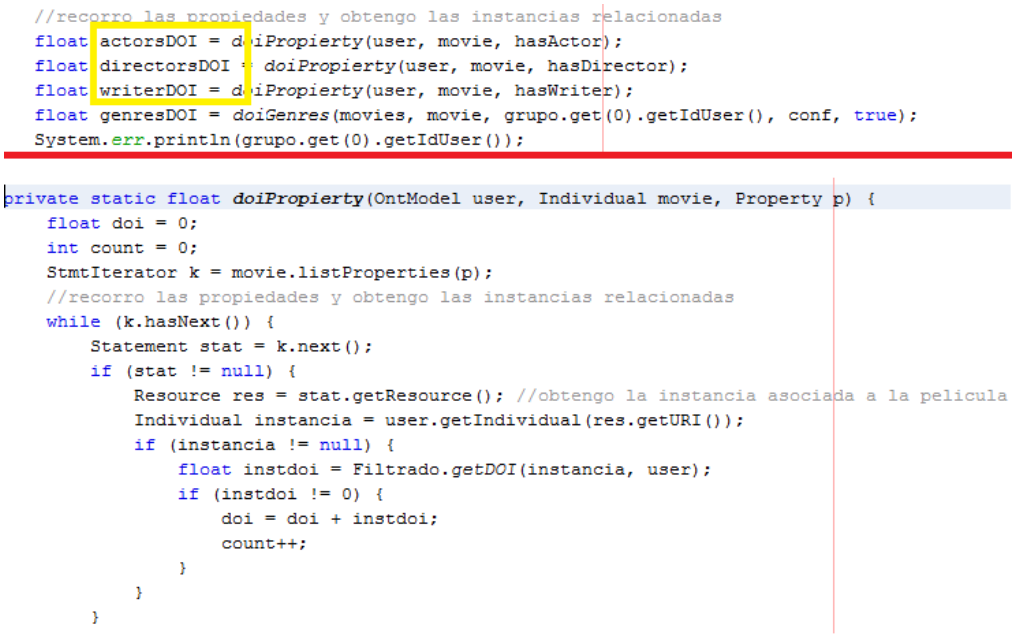
```
//obtengo sus generos, actores, directores, escritores
Property hasActor = user.getProperty(conf.getNS()+ "hasActor"); //propiedad q
StmtIterator k = movie.listProperties(hasActor);
float parcialDOI = 0;
int count=0;
//recorro las propiedades y obtengo las instancias relacionadas
while(k.hasNext()){
        Statement stat =k.next();
        if(stat!=null){
                Resource res = stat.getResource(); //obtengo la instancia aso
                Individual instancia = user.getIndividual(res.getURI());
                float instdoi = Filtrado.getDOI(instancia, user);
                if(instdoi!=0){
                        parcialDOI = parcialDOI + instdoi;
                        count++;
                }
        }
}

System.err.println(count+"dddd"+parcialDOI+"y"+parcialDOI/count);
float actorsDOI=0;
if(count!=0){
        actorsDOI = parcialDOI/count;
        generalcount++;
}


Property hasDirector = user.getProperty(conf.getNS()+ "hasDirector");
k = movie.listProperties(hasDirector);
parcialDOI = 0;
count=0;
//recorro las propiedades y obtengo las instancias relacionadas
while(k.hasNext()){
        Statement stat =k.next();
        if(stat!=null){
                Resource res = stat.getResource(); //obtengo la instancia aso
                Individual instancia = user.getIndividual(res.getURI());
                float instdoi = Filtrado.getDOI(instancia, user);
                if(instdoi!=0){
                        parcialDOI = parcialDOI + instdoi;
                        count++;
                }
        }
}
```

Figura 4.9: *Implementación del código original por [5].*

Un ejemplo de ello se manifiesta en las figuras 4.9 y 4.10. Observando que en el código original, mostrado en la figura 4.9, se utiliza una extensa cantidad de líneas de código solamente para obtener el DOI de dos propiedades (recuadro amarillo, actor y director), mientras que, en la figura 4.10, la función genérica "doiPropierty" adquiere el mismo resultado con un número muy reducido de líneas de código. De esta manera se puede agregar fácilmente cualquier propiedad extra que sea necesaria, como en el ejemplo mostrado, donde se obtiene el DOI de tres propiedades (recuadro amarillo; actor, director y escritor).



```
//recorro las propiedades y obtengo las instancias relacionadas
float actorsDOI = doiPropierty(user, movie, hasActor);
float directorsDOI = doiPropierty(user, movie, hasDirector);
float writerDOI = doiPropierty(user, movie, hasWriter);
float genresDOI = doiGenres(movies, movie, grupo.get(0).getIdUser(), conf, true);
System.err.println(grupo.get(0).getIdUser());
```

```
private static float doiPropierty(OntModel user, Individual movie, Property p) {
    float doi = 0;
    int count = 0;
    StmtIterator k = movie.listProperties(p);
    //recorro las propiedades y obtengo las instancias relacionadas
    while (k.hasNext()) {
        Statement stat = k.next();
        if (stat != null) {
            Resource res = stat.getResource(); //obtengo la instancia asociada a la pelicula
            Individual instancia = user.getIndividual(res.getURI());
            if (instancia != null) {
                float instdoi = Filtrado.getDOI(instancia, user);
                if (instdoi != 0) {
                    doi = doi + instdoi;
                    count++;
                }
            }
        }
```

Figura 4.10: *Función genérica creada en el proyecto.*

Con motivo de la realizaron de pruebas, se ha creado algoritmos de recomendación genéricos que reciben parámetros exclusivos con el fin de obtener resultados diferentes para poder cumplir con los objetivos planteados. Esto se explicará más detalladamente en el capítulo 5.

4.4. Diseño conceptual y estructura modular del sistema de recomendación semántico.

El sistema de recomendación semántico propuesto en el proyecto está estructurado según se muestra en la figura 4.11. Este diseño posibilita la generación de una recomendación para un usuario partiendo de un conjunto de ítems dado, y además se incluyen varios módulos adicionales que permiten realizar evaluaciones paramétricas, que se detallan en el capítulo 5.

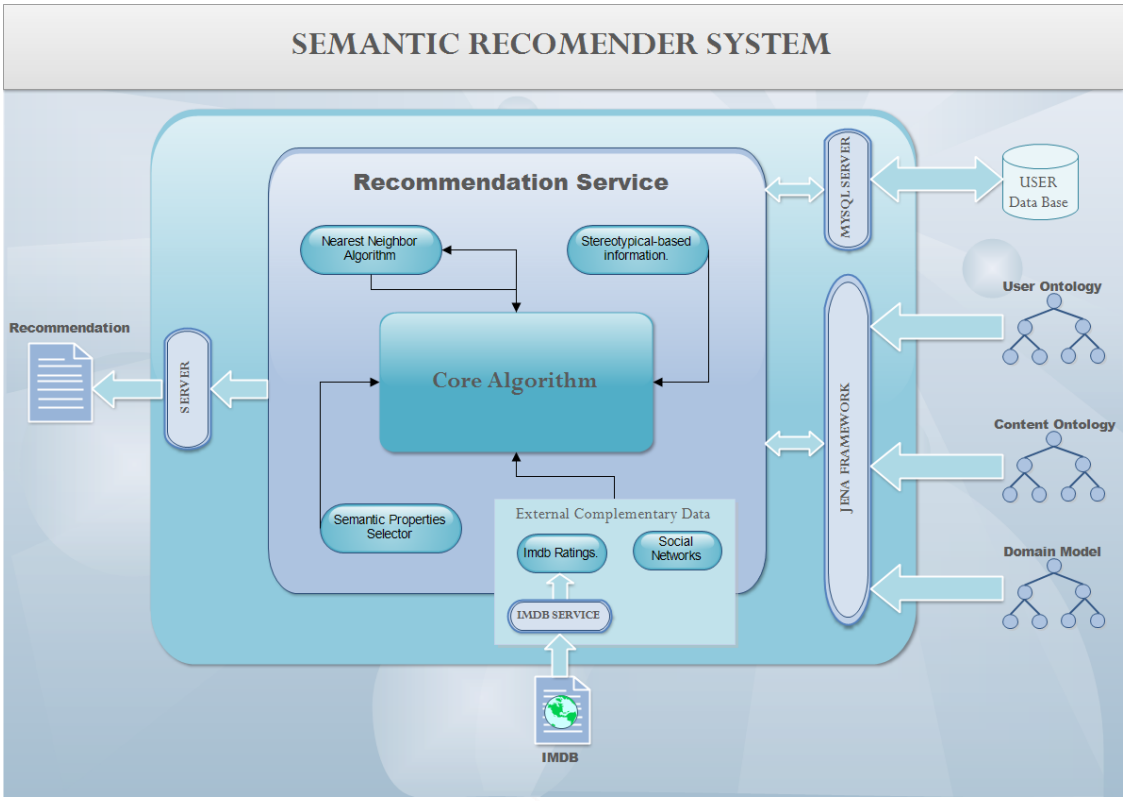


Figura 4.11: *Arquitectura del Sistema.*

4.4.1. Parámetros de entrada.

El sistema tiene como parámetros de entrada tres ontologías, las cuales son:

- **User ontology.** Consta del perfil ontológico de cada usuario, explicado en la sección 4.2.4.
- **Content ontology.** Esta ontología corresponde a la programación o contenido y consta de un conjunto de ítems expresados de manera semántica. De este conjunto se obtendrá cuáles ítems son los más "Idóneos" para cada usuario.
- **Domain ontology.** Corresponde a la estructura en la que se basan las dos ontologías mencionadas anteriormente.

Una entrada de naturaleza no-semántica es la base de datos de los usuarios, explicada detalladamente en la sección 4.3.1.

4.4.2. Servicio de recomendación.

El servicio de recomendación está dividido en dos partes principales, que constituyen componentes clave del proyecto:

- **Algoritmo núcleo.** Explicado detalladamente en la sección 4.5.
- **Módulos complementarios.** La sección 4.6 ofrece un detallado análisis de cada uno de estos elementos.

El servicio está creado para la fácil modificación y/o agregación de los componentes, es decir, si se desea excluir un módulo, simplemente se lo "desactiva", para que el sistema opere sin esta función.

4.4.3. Salida o recomendación.

La recomendación final consta de una lista ordenada de sugerencias de alternativas de entretenimiento para cada usuario. Esta lista presenta desde los ítems con mayor compatibilidad, hasta aquellos menos relacionados a los intereses del usuario.

4.5. Algoritmos de recomendación y núcleo del sistema.

El bloque central del sistema es el módulo recomendador, aquí se agrupan las clases y funciones principales para vincular los perfiles de usuario, las programaciones televisivas y generar una recomendación final. En esta sección se explicará el funcionamiento de los algoritmos utilizados.

4.5.1. Algoritmo de recomendación semántico por dispersión [5].

Este algoritmo hace uso de una semántica básica para realizar las recomendaciones sin implementar ninguna clase de inferencia ni búsqueda de conocimiento implícito. Las recomendaciones generadas por el algoritmo se realizan basándose en las propiedades del recurso y el grado de interés de usuario en esas propiedades. Para analizar la operación de este algoritmo, se planteará un ejemplo en el que

se desea obtener una valoración para la película "*Los piratas de caribe, la maldición del perla negra*" para un usuario determinado. La figura 4.12 muestra la representación de dicha película en forma de ontología, en la que resulta evidente, que no se trata de una representación formal ya que todas las propiedades de tipo recurso representadas en óvalos, deben estar descritas con una URI y no con el nombre directo de la propiedad. Para el ejemplo del funcionamiento del algoritmo, sin embargo, se encontró más oportuno representar los recursos de esta manera.

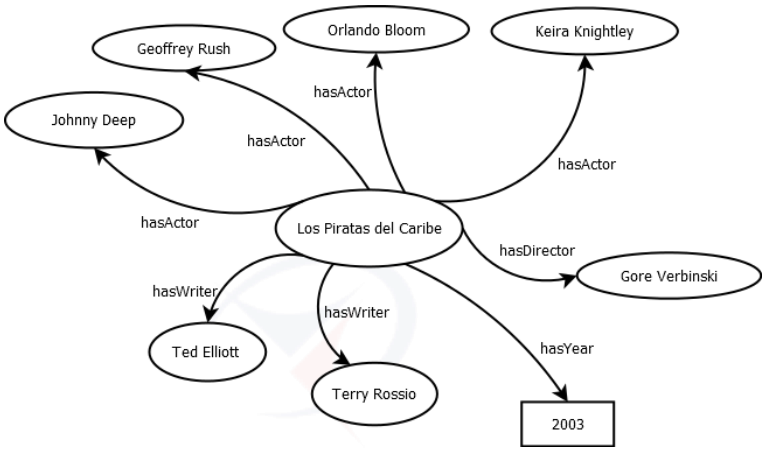


Figura 4.12: *Ontología para la película piratas del caribe*

Para obtener una calificación para esta película, el algoritmo calcula el interés del usuario para cada una de las propiedades semánticas agrupadas por su tipo, es decir, calcula un valor para todos los *actores*, otro para los *directores* y un tercer valor para los *escritores*. Adicionalmente, se calcula el valor del interés del usuario hacia el género de la película, puesto que este recurso es una instancia de la clase del *género*, en este caso de las clases *Acción*, *Aventura* y *Fantasía*. Para encontrar un valor para cada grupo de propiedades semánticas (Actor, Director, Escritor) se calcula el DOI de cada recurso según su tipo y se realiza un promedio, es decir, el valor para una propiedad es igual a la suma del DOI de cada uno de sus recursos dividida entre el número total de ellos.

Para este caso, suponiendo que el usuario tiene especificado en su perfil para el actor *Johnny Deep* un DOI de 4.5, para *Geoffrey Rush* un DOI de 3.8, para *Keira Knightley* 4.2 y ningún DOI para Orlando Bloom, entonces, el DOI de la propiedad actor será (4.5+3.8+4.2)/3=4.16. A pesar de que hayan cuatro actores, el algoritmo no promediará recursos que no tengan asignado un DOI.

Se realiza entonces un procedimiento similar para las demás propiedades semánticas, concretamente, para directores, escritores y otras propiedades que pueden incluirse en caso de que existan; cada uno de estos valores es sumado y se realiza un segundo promedio para encontrar el DOI de la película; por lo tanto, el DOI o grado de interés que calcula el algoritmo para un recurso de Programación televisiva será:

$$DOI_t = \frac{\sum_{i=1}^{n} DOI(P)}{n} \tag{4.1}$$

En donde:

- **n** representa el número de propiedades semánticas del recurso que poseen un valor de DOI
- **P** representa la propiedad semántica P.
- **DOI(P)** representa el DOI de la propiedad semántica calculada como en el ejemplo
- DOI_t representa el DOI total del recurso a calificar.

A la ecuación 4.1 se le puede incluir un mecanismo de ponderación o peso para cada propiedad semántica que la convierte en la ecuación 4.2:

$$DOI_t = \sum_{i=1}^{n} DOI(P)W(P) \tag{4.2}$$

En donde **W(P)** representa la ponderación de la propiedad semántica en cuestión. Nótece que en este caso se elimina la división para n de la ecuación 4.1 puesto que la suma de los pesos de las propiedades semánticas ponderadas debería ser $= 1$ (que representa el 100 % del valor de los pesos), es decir que:

$$\sum_{i=1}^{n} W(P) = 1 \tag{4.3}$$

En el capítulo siguiente se hace un estudio sobre el impacto del uso de las propiedades semánticas en el error de la recomendación al momento de quitarlas o incluirlas en el algoritmo. En la figura 4.13 se muestra el proceso explicado anteriormente en forma de Pseudocódigo, el mismo que se realiza para cada película de la parrilla de programación para obtener una lista ordenada de cada película con su respectiva valoración, al final del proceso se devuelve dicha lista.

```
Funcion_Recomendar
  Entradas: Id del usuario, contenido a recomendar
  Salidas: DOI del contenido
Inicio:
  Leer la descripción semántica del contenido
  Leer el perfil de usuario
  Extraer las propiedades semánticas del contenido
  Para cada propiedad hacer:
    Extraer los recursos de la propiedad
    Para cada recursos hacer:
      Buscar el DOI del recurso en la ontología del perfil
      Acumular el DOI en caso de que exista
  Acumular el DOI de la Propiedad si es Mayor a 0
  Calcular el DOI total del promedio del DOI de las Propiedades
  Devolver el DOI promedio
FIN.
```

Figura 4.13: *Pseudocódigo del Algoritmo de Recomendación por dispersión.*

4.5.2. Algoritmo de Recomendación con inferencia Semántica [5][1][2]

Con una complejidad mayor tanto computacional como lógica, éste algoritmo busca conocimiento implícito encontrando relaciones entre los recursos semánticos que comparten propiedades en común. Con estas relaciones posibilitan la formación de secuencias de recursos para al final, obtener una calificación para el contenido televisivo analizando cada una de las secuencias encontradas. Para ello, el algoritmo realiza una serie de procedimientos que se describen a continuación:

Creación de las cadenas de secuencias.

En esta etapa se tratan de encontrar relaciones entre los recursos conectando las propiedades semánticas que comparten entre sí, por ejemplo, en la figura 4.12 se mostró el modelamiento para la película *"Piratas del caribe, la maldición del perla negra"* y se hizo la suposición de que un usuario en particular mostraba interés hacia el actor *Johnny Deep* en un grado de 4.5 sobre 5. Lo que se trata de hacer a continuación, es encontrar contenidos televisivos que entre sus propiedades semánticas incluyan el recurso "Johnny Deep", para seguidamente, crear una cadena con la primera película. Esta tarea no resulta complicada debido a que en la propia descripción del recurso "Johnny Deep" se expresan explícitamente los roles que cumple este recurso en otros contenidos televisivos.

Para aclarar esta idea, se tomará como referencia la figura 4.12 que contiene el recurso "Johnny Deep", el mismo que se extiende para obtener las conexiones

mostradas en la figura 4.14. Así, se ha podido relacionar la película "Los piratas del caribe, la maldición del perla negra" con las películas "Rango" y "Alicia en el país de las maravillas" puesto que en todas aparece el recurso "Johnny Deep" como actor. Es importante mencionar que no solo se pueden conectar recursos de tipo película o del mismo tipo entre sí, sino que por ejemplo, se puede conectar la película "Rocky" con el reality show "El Retador" puesto que en ambos recursos aparece "Silvester Stallone" entre sus propiedades.

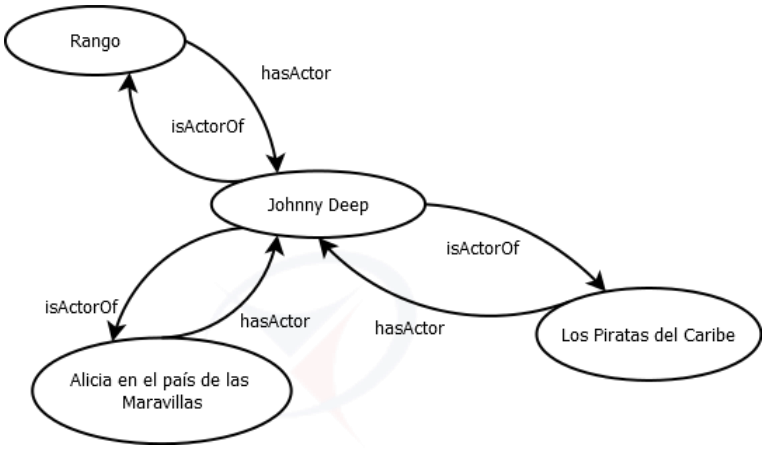


Figura 4.14: *Descripción del recurso "Johnny Deep"*

Como se mencionó anteriormente, la representación de la figura 4.14 se utiliza exclusivamente a manera de ejemplo, puesto que todos los recursos deben estar identificados con una URI. La figura 4.15 muestra un ejemplo de representación formal en formato RDF del recurso "Johnny Deep" en donde se observa claramente el uso de las URI para identificar al recurso y a todas las propiedades de tipo recurso.

```
<rdf:Description rdf:about="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#JohnnyDepp">
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0109707"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0113972"/>
  <rdf:type rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#Actor"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0106387"/>
  <hasName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Johnny Depp</hasName>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0181833"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0138304"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0120669"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0162661"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0119008"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0099487"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0112883"/>
  <isActorOf rdf:resource="file:///home/marcos/workspace/Ontology/ONTOLOGIES/OntologyIMDB.owl#tt0108550"/>
</rdf:Description>
```

Figura 4.15: *Representación RDF del recurso "Johnny Deep"*

Siguiendo este principio, el algoritmo crea cadenas de conexiones (también llamadas secuencias) para relacionar varios contenidos entre sí, aunque es importante mencionar que a pesar de que dos conceptos estén relacionados, el algoritmo sólo

incluirá en las secuencias aquellos conceptos que sean de interés para el usuario, para esto, la primera conexión considera únicamente los conceptos que tengan expresado un DOI superior a 3 en el perfil del usuario en cuestión, puesto que este número representa más de la mitad de la valoración en el rango de calificaciones (1-5); a partir de la primera conexión se incluye sólo a los recursos que tengan un DOI superior al recurso anterior en la secuencia. Para explicar esto siguiendo el ejemplo anterior enfocado a la película "Piratas del caribe, la maldición del perla negra", se tiene que al crear las primeras conexiones en la secuencia, el algoritmo contempla únicamente a los recursos que tengan un DOI mayor que 3 en el perfil del usuario, así por ejemplo, el recurso "Orlando Bloom" será descartado. En la figura 4.16 se observa la primera conexión entre el recurso y sus propiedades, donde puede notarse que para el ejemplo mencionado, solo se ha tomado en cuenta las propiedades de tipo "Actor", aunque de hecho, el algoritmo realiza esto con todos los tipos de propiedades (Escritor, Director, Etc.)

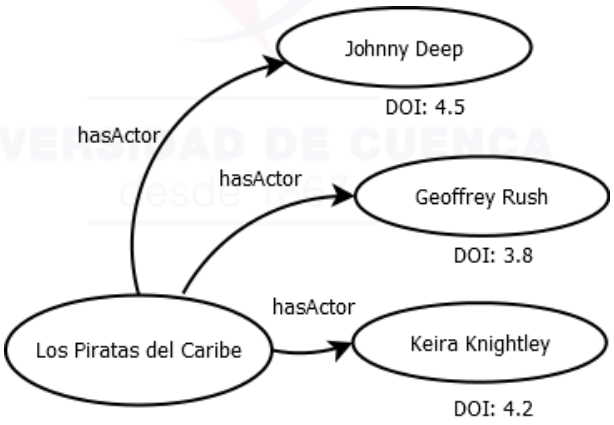


Figura 4.16: *Primera cadena de conexiones*

En una segunda etapa, el algoritmo busca recursivamente conexiones con más recursos con propiedades comunes, sin embargo, como se mencionó, éste agrega a la secuencia solo los conceptos que tengan un DOI superior al anterior. Así, suponiendo que la película "Alicia en el país de las maravillas" tiene un DOI de 5 para el usuario, mientras que la película "Rango" un DOI de 4, lo que resulta es que en la secuencia se agregará unicamente la primera película. Después de la segunda iteración la secuencia quedará como se muestra en la figura 4.17.

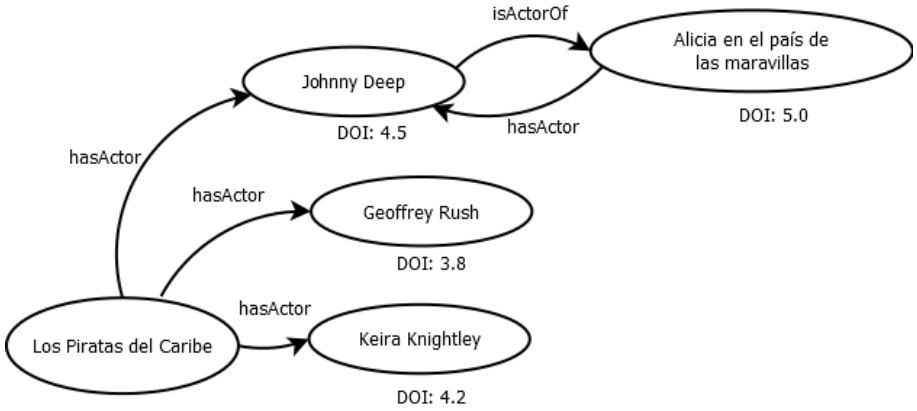


Figura 4.17: *Segunda cadena de conexiones, con el actor Johnny Deep*

Puesto que el procedimiento es recursivo, la segunda iteración completa se muestra en la figura 4.18; para el caso del ejemplo se ha creado una conexión con un solo contenido aunque en la realidad no necesariamente debe ocurrir esto, se crearán tantas conexiones como sea posible siempre y cuando el DOI del recurso sea mayor que el anterior.

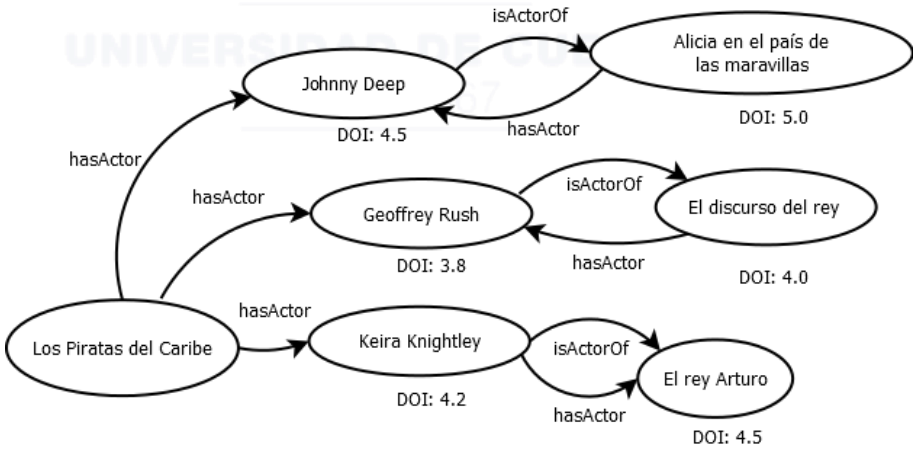


Figura 4.18: *Creación de la secuencia en la segunda iteración*

Otro aspecto de vital importancia en el algoritmo es que un recurso no puede aparecer más de una vez en la secuencia. Por ejemplo, la película *"Los piratas del caribe, el cofre de la muerte"* cuenta con los mismos tres actores mencionados anteriormente en su elenco, por lo que se podrían crear conexiones a tres instancias de la misma película que a su vez volverían a conectar a los actores, creándose de esta manera un ciclo infinito de conexiones. Para controlar que esto no suceda, antes de agregar un recurso a la secuencia, el algoritmo la recorre por completo verificando

que el recurso no haya sido agregado antes y en caso de ser así, simplemente lo descarta, de esta manera se asegura que no se creen ciclos de conexiones, por lo tanto la película "Los piratas del caribe, el cofre de la muerte" se conectará solo una vez con el recurso "Johnny Deep" (por ser el primero en ser analizado) y será descartada en las siguientes apariciones, ya sea mediante los Actores, Directores, Escritores o en general, a través de cualquier otra propiedad.

La adición de recursos a la secuencia terminará cuando ya no hayan nuevos conceptos para agregar o simplemente cuando se llegue a un límite de profundidad pre-establecido. Este límite es conocido como ***path***. Al finalizar el proceso recursivo, se tendrá una secuencia de conexiones parecida a la que se muestra en la figura 4.19. Se puede notar que las posiciones impares $(1, 3, 5, .., 2n + 1)$ siempre se encontrarán recursos de tipo contenido televisivo (películas, novelas, series), mientras que las posiciones pares$(2, 4, 6, .., 2n)$ tendrán recursos de tipo propiedad (Actores, Directores, Escritores). Este es un aspecto muy relevante para el siguiente paso del algoritmo, en donde se aislarán las secuencias individualmente para encontrar su longitud. La longitud de las secuencias será más grande mientras más conceptos nuevos aparezcan o mientras no se haya alcanzado el path pre-establecido.

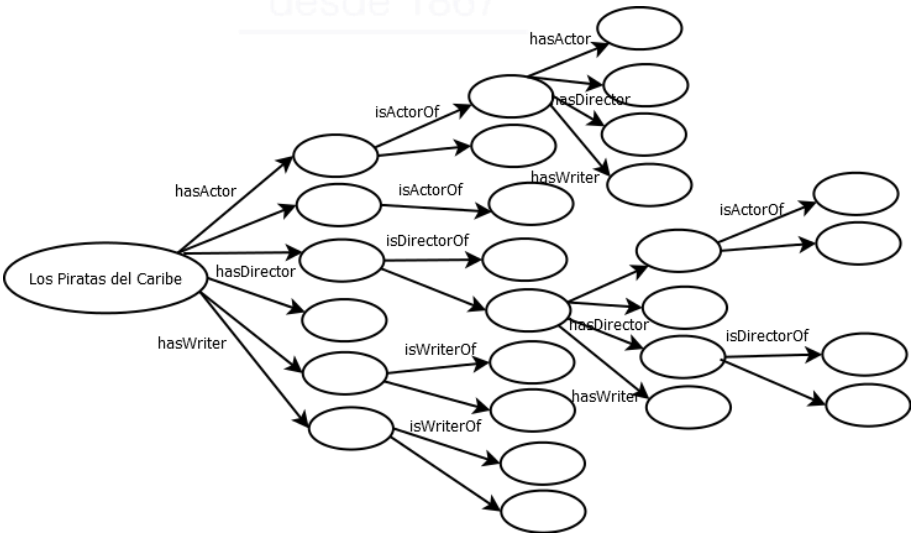


Figura 4.19: *Ejemplo de secuencias de un contenido televisivo*

Búsqueda de relaciones rho-path.

Una vez que se tenga la cadena de secuencias completa, el siguiente paso es encontrar todas las relaciones *rho-path* existentes; dos recursos semánticos tienen una relación *rho-path* cuando es posible establecer por lo menos un camino que los conecte [5]. Del ejemplo citado a lo largo de esta sección se puede establecer que la película "Los piratas del caribe, la maldición del perla negra" tiene una relación *rho-path* con la película "Alicia en el país de las maravillas" puesto que se las puede conectar mediante el actor "Johnny Deep". Para encontrar todas las relaciones *rho-path*, se descompone la cadena de secuencias en secuencias individuales como se muestra en la figura 4.20, la cual representa la descomposición de las secuencias de la figura 4.18.

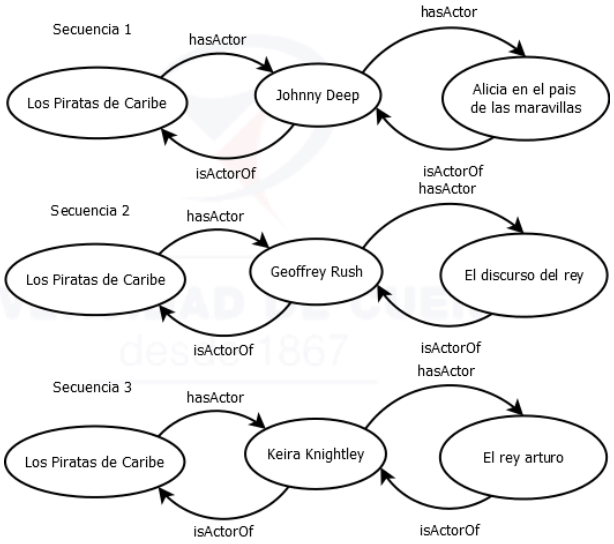


Figura 4.20: *Descomposición individual de secuencias*

A continuación se crean nuevas secuencias incluyendo únicamente a los recursos de tipo programación televisiva. En la figura 4.21 se observa las secuencias creadas, cada una de ellas es conocida como una relación *rho-path*; en el ejemplo, la longitud de la relación es de dos, pero la longitud y el número de relaciones varía dependiendo de la longitud de las secuencias originales. Suponiendo que se ha creado una relación adicional entre la película "Alicia en el país de las maravillas" y la serie televisiva "En terapia", puesto que ambos recursos comparten a la actriz "Mia Wasikowska", para esta secuencia se crearán dos relaciones *rho-path*: una entre las películas "Los piratas del Caribe" y "Alicia en el país de las maravillas" cuya longitud es de dos y una segunda relación entre la película "Los piratas del caribe" y la serie televisiva "En terapia" cuya longitud es de tres. En conclusión, se creará una relación *rho-path*

para cada par de recursos de programación televisiva conectados entre sí mediante algún camino o *path*.

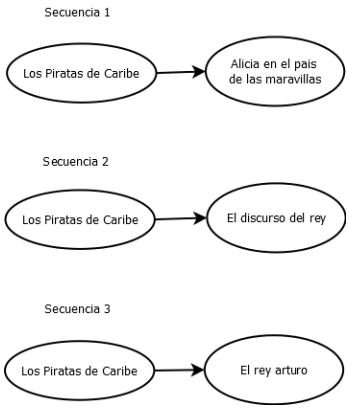


Figura 4.21: *Relaciones* Rho-Path

Búsqueda de relaciones *rho-join*

Existe una relación *rho-join* entre dos recursos siempre y cuando los dos sean instancias de la misma clase de unión, por ejemplo, existe una relación *rho-join* de longitud 1 entre todos los recursos que son de tipo película de acción, y una relación *rho-join* de longitud 2 entre todos los contenidos que son de tipo series sin tomar en cuenta su género. Según el modelo de AVATAR, si dos recursos tienen tanto una relación *rho-join* como una *rho-path* se infiere la relación *rho-path* entre los contenidos en cuestión.

Cálculo del DOI para un recurso de programación televisiva.

El paso final del algoritmo luego de obtenidas todas las relaciones rho-path y rho-join es proceder a valorar el contenido televisivo, para ello se aplica las siguiente fórmula:

$$DOI_t = \frac{\sum_{i=1}^{n} DOI(R_i)/length(R_i)}{\sum_{i=1}^{n} 1/length(R_i)}[5] \tag{4.4}$$

En donde:

N es el número total de relaciones rho-path y rho-join encontradas.

DOI(R_i***)*** representa el grado de interés del usuario sobre el último recurso de una relación.

length(R_i***)*** representa la longitud de la relación rho-path o rho-join.

De esta fórmula se puede interpretar que se realiza un promedio ponderado del DOI de los recursos en el cual, los recursos más alejados o las relaciones más largas son semánticamente menos influyentes que los recursos más próximos al recurso objetivo.

4.6. Módulos complementarios.

Como se explicó en la sección 4.4.2, el sistema incluye cuatro módulos satélites que incorporan funciones e información adicionales al algoritmo núcleo:

1. **Información de estereotipos:** Stereotypical-based Information.
2. **Selector de propiedades semánticas:** Semantic Properties selector.
3. **Algoritmo de vecinos cercanos (KNN):** Nearest Neighbor Algorithm.
4. **Componentes externos no-semánticos:** External complementary data.

4.6.1. Información de Estereotipos

Este módulo intenta resolver los problemas de arranque en frió descritos en la sección 3.2, para lo cual, se agrupa a los usuarios nuevos o con un número reducido de calificaciones de acuerdo a ciertos factores demográficos o similares a manera de "Estereotipos" de tal modo que se pueda generar recomendaciones dirigidas a todos los miembros del grupo basándose en sus características comunes y reales, como por ejemplo su género y edad. Así el algoritmo describe un comportamiento de *filtrado pasivo* como los descritos en la sección 3.2.4.

División de los estereotipos.

Para la inclusión de este módulo en el sistema, se han contemplado catorce estereotipos diferentes: un grupo para cada género, y éstos a su vez, se subdividen en siete rangos de edades (menor a 18, 18-24, 25-34, 35-44, 45-49, 50-55, 56+). Este esquema se basa en la suposición que individuos del mismo género y con edades cercanas tienen preferencias televisivas relativamente similares. La división de los grupos de edades se ha hecho basándose en las recomendaciones de agrupación de *MovieLens*[11] descritas en la sección 4.2.5.

[11] http://www.movielens.org

Construcción de los Estereotipos.

Para la construcción de los estereotipos en primer lugar se agrupa a todos los usuarios según a la clasificación a la que corresponda, para esto se recorre el conjunto total de usuarios y, dependiendo del sexo y la edad, se ubica a cada usuario en su grupo correspondiente. Con los grupos ya definidos, se almacena en un archivo los identificadores de los usuarios que pertenecen a cada estereotipo.

El siguiente paso es la creación de una ontología para cada estereotipo, para lo cual se adquieren todos los identificadores de los usuarios almacenados en el archivo mencionado anteriormente y que pertenecen al estereotipo analizado con el fin de leer y agrupar las preferencias de todos los usuarios del grupo y unificarlas como si se tratase de un único usuario. La ontología del estereotipo se crea de la misma manera que aquella de un usuario normal, lo cual se describe en la sección 4.2.7.

Recomendación mediante estereotipos.

Una recomendación generada utilizando la información de los estereotipos se presenta cuando se trata de un nuevo usuario o cuando éste no ha alcanzado un número mínimo de calificaciones necesarias para estimación de una recomendación personalizada. Con este objetivo en mente, al momento que se requiera una recomendación, el sistema inicialmente verifica que el usuario haya alcanzado el límite antes descrito, de no ser así, se elije el estereotipo en el cual está definido dicho usuario y se procede a realizar la recomendación utilizando cualquiera de los algoritmos descritos en la sección 4.5, con la diferencia que no se tomará como entrada del algoritmo la ontología del usuario propiamente, sino la del estereotipo al que pertenece. En el capítulo siguiente se presentará un análisis que permite establecer cuantitativamente el límite mínimo de calificaciones necesario para que un usuario pueda recibir recomendaciones personalizadas.

4.6.2. Selector de propiedades semánticas

Los contenidos audiovisuales pueden contener tantas propiedades como se desee, sin embargo, el utilizar todas las propiedades disponibles en un SRS no necesariamente causará que las predicciones obtenidas sean más precisas e incluso, podrían reflejar una degradación de los resultados. Con el objetivo de analizar la influencia de la inclusión de las propiedades semánticas más comunes presentes en este tipo de contenidos en el SRS, se propone un módulo que permita realizar una variación

de los algoritmos núcleo que permita que cualquier algoritmo sea capaz de recibir como paramentos de entrada un conjunto de propiedades semánticas determinado, así, al éste realizará los cálculos corresponientes considerando únicamente las propiedades ingresadas. Este módulo permite analizar el impacto que causa sobre el error en la estimación de una recomendación, la inclusión o exclusión de propiedades semánticas en el algoritmo. Este estudio será presentado en el capítulo siguiente.

4.6.3. Algoritmo de vecinos cercanos (KNN).

Este módulo se ha concebido con la finalidad de generar una predicción diferente a la del sistema principal y que permita obtener resultados que puedan ser comparados. Su funcionamiento se basa en la utilización de una técnica llamada "Vecinos Cercanos" [46] , que permite encontrar "usuarios semejantes" y en consecuencia, perfiles de usuarios con gustos similares.

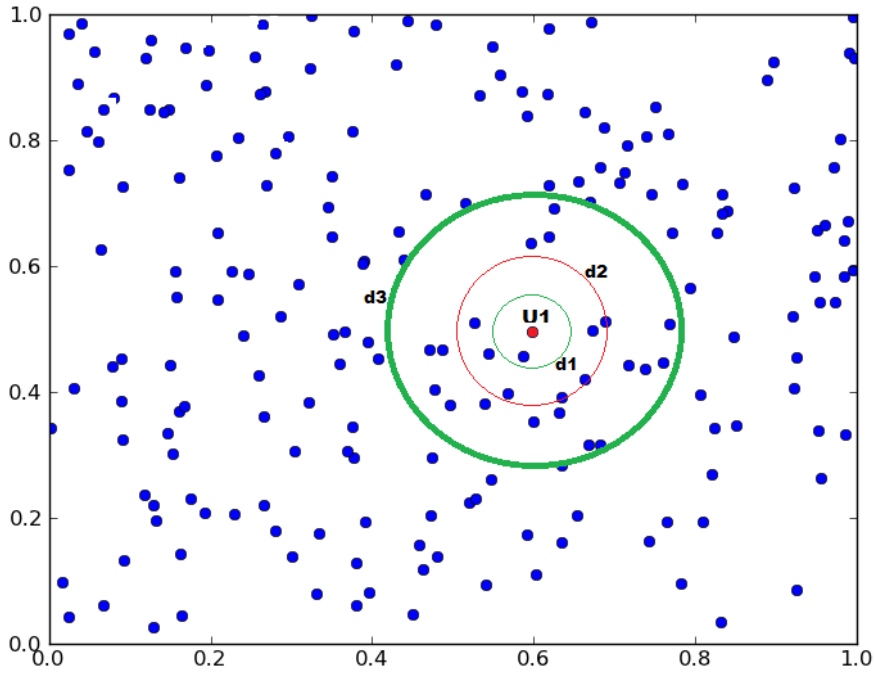


Figura 4.22: *Ejemplo de KNN.*

Este algoritmo determina un número de usuarios o "vecinos" que se encuentren dentro de un rango dado en base a una medida de similaridad. Por ejemplo, se observa que en la figura 4.22 el punto rojo "*U1*" representa el usuario sobre el cual se desea encontrar los vecinos cercanos. La posición de U_1 se fija de acuerdo a sus gustos y preferencias según dos parámetros, que se mapean al plano bi-dimensional

en los ejes x y y. Un punto cualquiera se le posiciona en el plano de manera que cada eje toma el valor un género de cada usuario, es decir, el eje x puede ser drama y el eje y comedia, entonces un usuario tiene un valor de cada uno de estos géneros, por ejemplo, el usuario U_1 tiene aproximadamente 0.6 en el eje x que es drama y un 0.5 en el eje y que es comedia.

U_1 puede tener vecinos de acuerdo a la distancia que se determine en un inicio, por ejemplo, si se observa la imagen, con una distancia dada d_1, el usuario U_1 tiene tan solo un vecino que se encuentra en un rango menor a d_1. Por otra parte, si se aumenta la distancia a d_2, el mismo usuario tiene alrededor de 8 vecinos. Mientras más grande la distancia más vecinos son posibles de encontrar, sin embargo, hay que notar, que el aumentar la distancia ocasionaría que los vecinos sean menos cercanos con respecto a sus gustos y preferencias.

Para encontrar la separación entre dos puntos, se utiliza la distancia euclidiana, cuya fórmula para $n-dimensiones$ se describe en la ecuación 4.5, donde las coordenadas de los puntos son representadas por: $P = (p_1, p_2, .., p_n)$ y $Q = (q_1, q_2, .., q_n)$.

$$d_E(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{4.5}$$

El módulo implementado permite encontrar vecinos cercanos según los siguientes parámetros:

1. **Distancia euclidiana.** La distancia máxima a buscar vecinos.
2. **Número máximo de vecinos.** Es el número máximo de vecinos que puede tener un usuario, específicamente, se obtiene los n más cercanos.
3. **Número mínimo de películas.** Se le otorga un número mínimo de películas que el vecino debe haber calificado para ser tomado en cuenta, si no cumple con este requisito se descarta.

Una vez que se han obtenido los vecinos de un usuario, la valoración de un ítem se predice en base a la ejecución del algoritmo núcleo 4.5 para cada uno de sus vecinos; posteriormente se realiza un promedio de las predicciones de los vecinos para presentarla como la predicción de dicho usuario.

4.6.4. Componentes externos no-semánticos.

Para la incorporación de este módulo, se parte de la premisa de que una predicción obtenida mediante nuestro SRS, puede enriquecerse mediante información externa, no necesariamente de naturaleza semántica. Para ello, se plantea mejorar la exactitud de las predicciones generadas por el SRS al combinar sus resultados con información externa, la cual está disponible en la web, tal como aquella extraída de IMDB[12] o de redes sociales.

Por ejemplo, una película catalogada en IMDB generalmente tiene una calificación promedio (*average rating*) obtenida de entre las valoraciones otorgadas por miles de usuarios, así, puede asumirse como un indicativo estable y preponderante que potencialmente aportaría positivamente a la precisión del sistema de recomendación, razón por la cual, ha sido considerada en la versión presente del sistema.

Se estima que una futura versión del sistema, debe incluir un enriquecimiento de perfiles de usuario por medio de las redes sociales (Facebook[13], Twitter[14], etc.), para incluir información adicional de carácter explícito en las ontologías de usuario, tal como gustos sobre los actores, escritores, etc, que puede ponderarse con un peso mayor en la estimación de las predicciones dada su condición de información explícita provista por el usuario.

[12]Internet Movie Data Base: www.imdb.com
[13]www.facebook.com
[14]www.twitter.com

Capítulo 5

Evaluación del Sistema de Recomendación

5.1. Introducción.

En este capítulo se muestran las evaluaciones y pruebas realizadas que permiten obtener los resultados que proporcionarán los mejores parámetros de ajustes para el sistema de recomendación, lo que contribuiría a que la estimación de las predicciones del sistema sean más acertadas.
En las secciones subsiguientes se detalla el entorno de prueba creado por el grupo de trabajo, se resume los procedimientos de análisis de los parámetros y módulos explicados en el capítulo 4 como parte de la Evaluación Cuantitativa (sección 5.3), mostrando resultados satisfactorios y concluyentes. Por último, se propone un procedimiento de Evaluación Cualitativa como trabajo futuro, para cuando el sistema se encuentre en uso.

5.2. Entorno de Prueba.

Con el fin de realizar pruebas o experimentaciones sobre el comportamiento de los algoritmos de recomendación se ha visto la necesidad de crear un entorno de pruebas adecuado que proporcione una serie de lineamientos para la realización de cada una de ellas. En esta sección se describe dicho entorno, los procedimientos que se seguirán para cada experimentación, las métricas de evaluación y la forma de presentación de resultados finales.

5.2.1. Descripción del escenario de pruebas.

A través de la experimentación se busca estudiar el comportamiento de los algoritmos según se varían los datos de entrada o ciertos parámetros de ajuste, para luego encontrar una combinación adecuada que permita reducir el error en la recomendación. Para cada prueba, se ejecutará el algoritmo de recomendación las veces necesarias, se almacenarán y analizarán los datos de salida para finalmente obtener conclusiones para cada experimento.

El escenario típico para la realización de los experimentos o pruebas al algoritmo se describe como sigue:

1. Se analizará y especificará los resultados a obtener de cada prueba.
2. Se preparará al algoritmo para dicha prueba.
3. Se elegirán y construirán los conjuntos (sets) de entrenamiento y pruebas adecuados.
4. Se analizarán los resultados y se obtendrán las conclusiones de cada experimento.

Preparación de los algoritmos de recomendación

Para que sea posible evaluar el comportamiento de los algoritmos de recomendación de acuerdo a la variación de ciertos parámetros, ha sido necesario realizar varias modificaciones en ellos para que puedan recibir como información de entrada los diversos parámetros de ajuste con los que se desea ejecutar las pruebas. Además de esto, se modifican las salidas de cada una de estas mutaciones para que devuelvan únicamente los resultados relevantes para cada prueba propuesta. Más adelante en este capítulo se expondrán todas las pruebas realizadas y las modificaciones al algoritmo mencionadas.

Origen de los datos de prueba

Para la realización de cada una de las pruebas de funcionamiento de los algoritmos se ha tomado como entrada el set de datos descrito en la sección 4.2.5, y, se separa el archivo que registra las calificaciones que los usuarios en dos partes: la primera que contiene los datos que servirán como entrada para la creación de las ontologías del perfil del usuario, y una segunda parte, que constituirá un archivo

que contiene datos que no se incluyen en el proceso de creación del perfil del usuario y que sirven exclusivamente para realizar las pruebas de las recomendaciones. Para lograr este objetivo, se divide el número total de calificaciones de cada usuario en un porcentaje para el primer set y el porcentaje restante para segundo, de esta manera se consigue tener un set de entrenamiento y un set de pruebas parecidos a los que se utilizan en procedimientos de pruebas para los algoritmos de inteligencia artificial o de aprendizaje de máquina.

5.2.2. Selección de Usuarios

A pesar de que todas las pruebas pueden ser ejecutadas utilizando los mismos sets de entrenamiento y pruebas, en muchas de ellas se necesita elegir un cierto número de pre-condiciones para que un usuario sea aceptable para incluirse en estos sets. Por ejemplo, no se puede incluir a un usuario que haya calificado cientos de ítems, en las pruebas del módulo de estereotipos, ya que por definición, este tipo de usuario no cumple con uno de los criterios de admisión de un estereotipo que es contar con un número reducido de calificaciones otorgadas. Adicionalmente, debido a la complejidad computacional que representa la ejecución de algunos procedimientos de prueba, no todos se ejecutan con un igual número de usuarios o calificaciones, por lo que se ha elaborado un algoritmo capaz de crear tanto un set de entrenamiento como uno de pruebas en los que se puede elegir:

- A los usuarios que cumplan con un rango de calificaciones determinado, por ejemplo, se tomará en cuenta solo a los usuarios que tengan de cien a doscientas películas calificadas.
- El número de usuarios a incluir en los set de entrenamiento y pruebas. Cabe recalcar que el número de usuarios en ambos sets se ha escogido de tal forma que sea siempre igual.
- El porcentaje de calificaciones consideradas para cada set. Por ejemplo, si un usuario tiene registradas 200 calificaciones se puede enviar el 80 % de ellas al set de entrenamiento y el 20 % restante al set de pruebas.

Como un ejemplo de la salida de este algoritmo se puede tener dos archivos que contengan, cada uno, 200 usuarios que tengan un rango de entre 200 y 220 películas calificadas, de las cuales se enviará el 85 % para el set de entrenamiento y el 15 % restante para el set de pruebas.

Otro aspecto importante de la selección de usuarios es que una vez creado el set de pruebas, el mismo se somete a un segundo filtro para eliminar las calificaciones inferiores a cuatro puntos, pues como se explica en [4], los contenidos audiovisuales con calificaciones de 4 en adelante son los que potencialmente serán recomendados a un usuario y consecuentemente otorgándole más importancia a la precisión.

5.2.3. Métricas de evaluación

La métrica más importante al momento de evaluar un algoritmo de recomendación es la precisión en las recomendaciones, la cual se mide promediando el error absoluto entre la calificación contenida en el archivo de test y la calificación obtenida por el recomendador. Dicho promedio es conocido como MAE y se expresa en la ecuación 5.1.

$$MAE = \frac{1}{N}. \sum_{u,i}^{U,I} |P_{u,i} - R_{u,i}| \tag{5.1}$$

Donde:

- U corresponde al conjunto de los usuarios examinados.
- I corresponde al conjunto de ítems evaluados.
- N corresponde al total de *ratings* comparados.
- P corresponde a la predicción obtenida por el sistema.
- R corresponde a la calificación otorgada por el usuario.

5.2.4. Presentación de Resultados.

Los resultados obtenidos de la realización de cada prueba serán presentados en forma gráfica, lo cual ofrece un mayor entendimiento del comportamiento del algoritmo ya que en la totalidad de las pruebas se obtiene como resultado una serie de tablas de resultados de gran tamaño, las cuales, al ser presentadas de esta manera, representarían por un lado una gran dificultad al momento de analizar los resultados obtenidos y por otro lado imposibilitarían una presentación de todos los resultados.

5.3. Análisis de resultados: Evaluación Cuantitativa.

La evaluación cuantitativa consta de varias pruebas realizadas a diferentes algoritmos y módulos, que permiten obtener indicadores numéricos que reflejen el desempeño de los diferentes componentes del sistema . Estas pruebas se explican con detalle a continuación.

5.3.1. Comparación de resultados de los algoritmos de recomendación.

En este experimento se hace una comparación entre las dos técnicas de recomendación mencionadas en la sección 4.5, concretamente, se compara los resultados del algoritmo de recomendación por inferencia semántica y aquellos del algoritmo de recomendación por dispersión.
El objetivo principal de esta comparación es verificar qué algoritmo tiene una mayor precisión al momento de calcular las predicciones de una calificación para un ítem determinado, para ello, se selecciona un total de cien usuarios que cuenten con un rango de 200 a 220 ítems calificados, asegurando de esta manera que hayan suficientes datos tanto para el entrenamiento del algoritmo como para el test. Además, al no existir ningún parámetro de ordenamiento en los usuarios se asegura su variedad en preferencias televisivas.

El procedimiento de experimentación de esta etapa, requiere que una vez determinado el grupo de usuarios antes mencionado, se proceda a crear los archivos de entrenamiento y de test usando un total del 80 % de las calificaciones otorgadas por el usuario para la creación de su ontología y el 20 % restante para el archivo de test. A continuación se crean las ontologías de cada usuario y al final se procede a ejecutar los dos algoritmos utilizando para cada uno las mismas ontologías de usuarios y el mismo archivo de pruebas, garantizando un entorno de prueba en igualdad de condiciones. Cada algoritmo calculará el error en la predicción obtenida con respecto a las especificadas en el set de prueba, calculará un promedio por usuario y un promedio general para el total de usuarios. Los resultados obtenidos se muestran y comentan en la siguiente sección.

Resultados

En la figura 5.1 se muestra el MAE o promedio de error por cada usuario evaluado. Puede notarse que el error tiene un comportamiento muy parecido tanto al utilizar el algoritmo de dispersión, representado en azul, como cuando se utiliza la técnica de inferencia semántica, representada en naranja. Es claro también que para todos los casos, el algoritmo de inferencia semántica presenta un menor promedio de error y en ciertos casos elimina algunos picos claramente identificables en las predicciones generadas con el primer algoritmo.

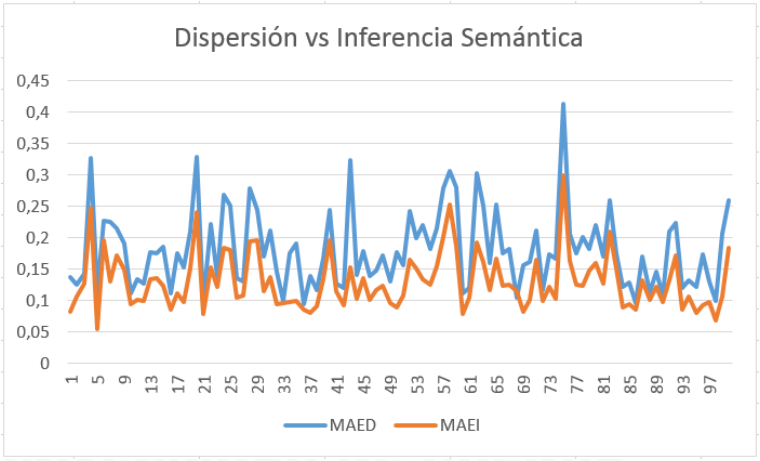


Figura 5.1: *Gráfico del MAE por usuarios, dispersión vs inferencia semántica.*

En la figura 5.2 se presenta el gráfico del MAE total de los usuarios evaluados, como es de esperar recordando el comportamiento descrito por la figura 5.1, el algoritmo de inferencia semántica presenta un menor promedio de error y en consecuencia, se puede decir que es más preciso.

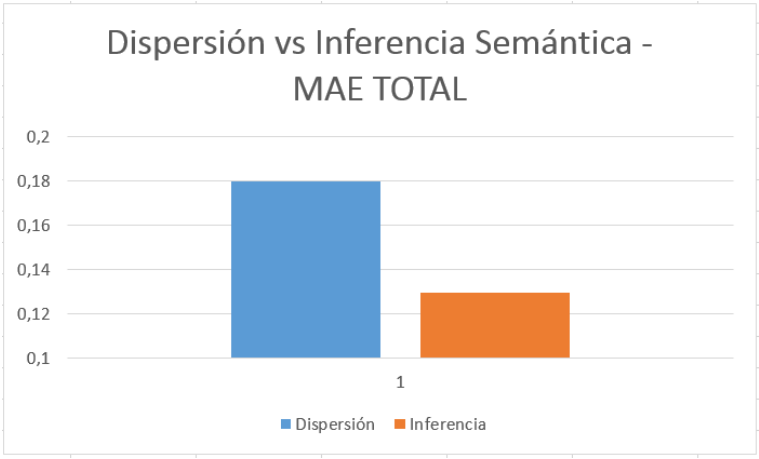


Figura 5.2: *Gráfico del MAE final, dispersión vs inferencia*

Conclusiones del experimento.

Dados los resultados obtenidos de la experimentación se puede concluir que en el algoritmo de recomendación por inferencia semántica tiene una mejor desempeño en la estimación de predicciones de la valoración para un ítem. Este resultado determina así, que en adelante en la mayoría de experimentos realizados utilicen este algoritmo como base, a menos que se indique lo contrario.

5.3.2. Impacto de las propiedades Semánticas en la Estimación de las Predicciones usando el algoritmo por dispersión

Para la realización de esta prueba, se utilizó el algoritmo por dispersión, explicado detalladamente en la sección 4.5.1. La finalidad de este experimento es demostrar que mientras más propiedades semánticas se utilicen, menor sera el MAE determinado por la ecuación: (5.1).

Conjunto de datos.

En primer lugar se seleccionó un grupo de usuarios con el mayor número de calificaciones, asegurando que al menos 100 usuarios igualen o superen ese número. El número base establecido fue un mínimo de 825 calificaciones por usuario. Así, se obtuvo un conjunto de 102 usuarios con más de 825 películas calificadas cada uno.

Se identificaron dos conjuntos disjuntos de datos: un set de entrenamiento y uno de prueba. Por cada usuario, del conjunto de películas calificadas, se eligieron las películas para cada set en base a dos criterios:

1. Las películas del set de prueba no pueden estar presentes en el set de entrenamiento.

2. Las películas del set de prueba deben tener al menos una propiedad semántica en común (a manera de una función heurística) con las películas del set de entrenamiento.

Estas dos condiciones aseguran que en el momento de realizar la prueba no ingresen datos no vinculados en lo absoluto con las preferencias del usuario representadas en la ontología, y además, que no ingresen datos que se hayan sometido

al entrenamiento y que puedan afectar de una manera u otra en los resultados.

Finalizado este proceso, se obtuvo un archivo para entrenamiento con 102 usuarios con un promedio de 580 películas calificadas por cada uno, y un archivo de test con los mismos usuarios pero con un promedio de 255 películas calificadas por cada usuario. Ninguna combinación de usuario-película-calificación están repetidas dentro de los archivos, ni incluida en ambos.

Procedimiento.

Con el conjunto de datos seleccionado, se procede a crear las ontologías, tal como se explica en la sección 4.2.7. Para la prueba, se introdujeron al sistema ocho combinaciones diferentes de propiedades semánticas en igual número de pruebas. La propiedad *Género* (Genre) se incluyó en todas las pruebas ya que es la única propiedad que tiene DOI en las ontologías de todos los usuarios. El algoritmo recibe una cadena de caracteres, los cuales son los parámetros de ejecución, es decir, se enviá a la función la combinación de propiedades con las cuales el algoritmo realizará el cálculo de las predicciones.

Se realizaron pruebas con las combinaciones de propiedades semánticas siguientes:

- Género.
- Género, Actor.
- Género, IMDB.
- Género, IMDB, Actor.
- Género, Actor,Director.
- Género, Actor, Director, Escritor.
- Género, Actor,Escritor.
- Género, Actor, Director, Escritor, IMDB

Resultados.

La figura 5.3 muestra los resultados para cada una de estas pruebas. En ella se presenta el error promedio por todos los usuarios o MAE (5.1) especificado para cada una de las combinaciones indicadas

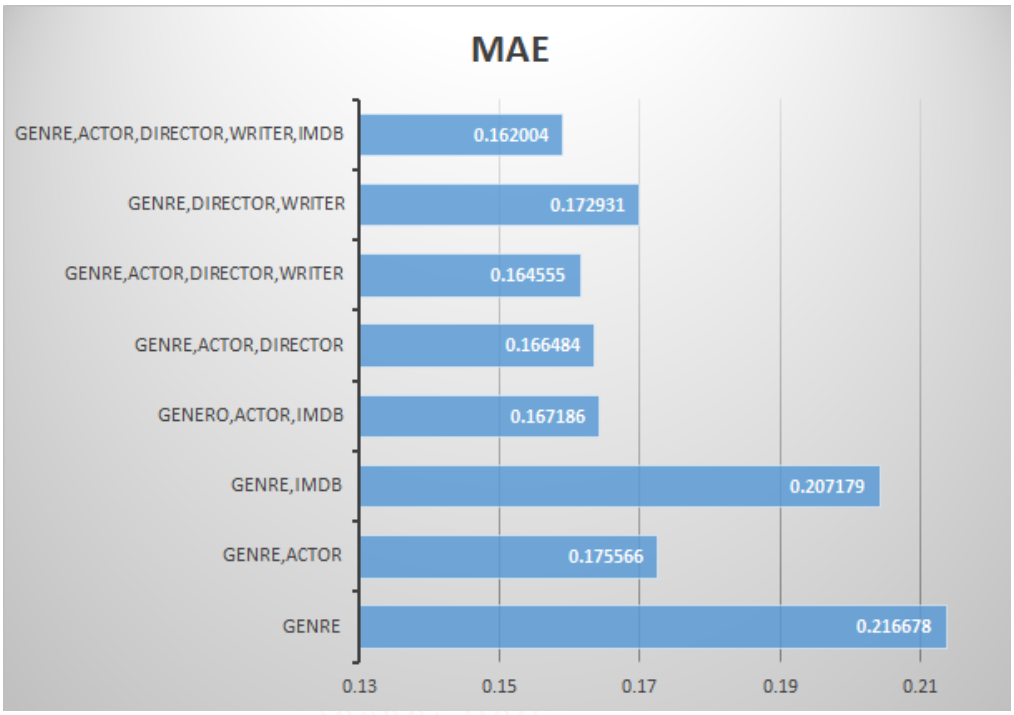


Figura 5.3: *Promedio de error de todos los usuarios en cada una de las pruebas [10].*

En la figura 5.4 presenta un gráfico del MAE por cada usuario, en cada una de las pruebas efectuadas.

Como se puede observar en la figura 5.3, la prueba que involucra únicamente la propiedad género, refleja error promedio del 21.6 % en la estimación de las predicciones, que en una escala del 1 al 5 representa un error promedio de 1.08 puntos; Al ingresar la propiedad actor, el error se reduce en un aproximado del 4 % llegando así a obtener un error del 17.5 %, que representa 0.87 puntos en la escala del 1 al 5. A medida que se siguen agregando propiedades semánticas, el error tiende a reducir aunque en un rango menor.

Se puede observar también que cuando se realiza la prueba con tres propiedades semánticas incluyendo la propiedad actor, se obtiene un error menor que ejecutando la prueba con igual número de propiedades semánticas sin incluir a la propiedad actor, lo cual sugiere que esta propiedad aporta considerablemente a la reducción del error, en relación a las otras propiedades del programa, sin tomar en cuenta

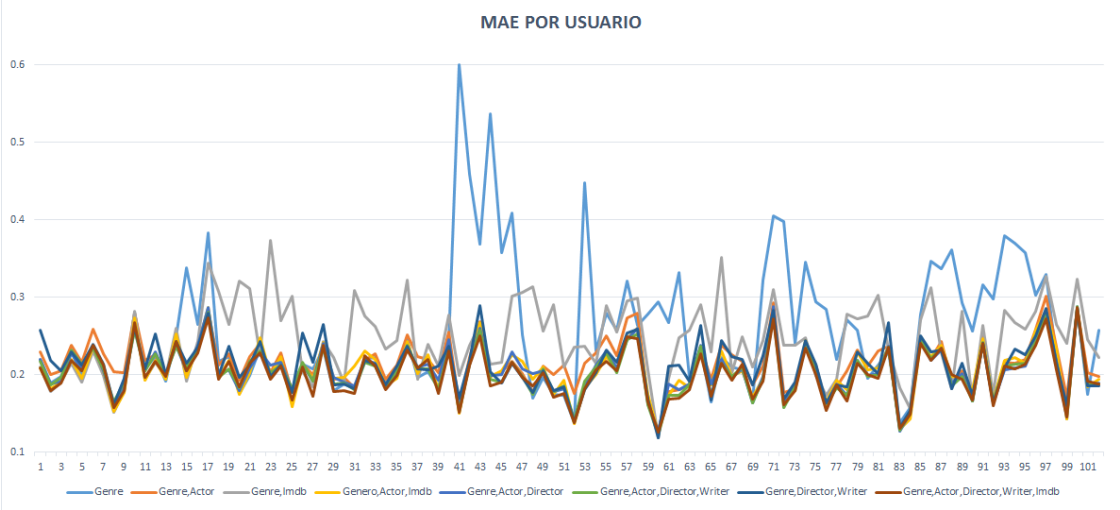


Figura 5.4: *Promedio de error de todos los usuarios en cada una de las pruebas [10].*

la propiedad género ya que como se mencionó anteriormente, ésta se utiliza para todas las pruebas.

Por otra parte cuando la incorporación de la propiedad IMDB, produce que el error tienda a reducirse ligeramente en todos los casos, sin embargo, este cambio parece ser mínimo en comparación con la reducción causada por la inclusión de las otras propiedades semánticas. De estas observaciones e puede concluir que el aporte de información de carácter no-semántico en la estimación de las predicciones, no es tan significativo como lo es el aporte de las propiedades semánticas.

Los picos y singularidades en la figura 5.4, reflejan casos particulares de usuarios con gustos dispersos (problema de la oveja negra, según se comentó en la sección 3.2.2).

Conclusiones del experimento.

Al combinar la propiedad Género con cualquier otra propiedad semántica, el error tiende a disminuir alrededor de un 3 %, llegando al 19 %, lo que indica que puede conseguirse una mejora sustancial sí, al menos, se utilizan dos propiedades semánticas en el algoritmo, como se reporta en [10].

Se puede concluir que mientras mas propiedades semánticas se utilice, el error será menor, aunque la diferencia no sea de sustancial al introducir ciertas propiedades.

5.3.3. Impacto de las propiedades Semánticas en la Estimación de las Predicciones usando el algoritmo con Inferencia Semántica

Esta prueba se realizó sobre el algoritmo de recomendación por inferencia semántica estudiado en la sección 4.5.2 En ella se trata de corroborar los resultados obtenidos en la prueba anterior, que confirmaban la hipótesis planteada de que mientras más propiedades semánticas se utilicen, menor será el MAE obtenido.

Conjunto de datos.

En este experimento, se seleccionó un total de 100 usuarios que tengan un rango de entre 200 y 220 películas calificadas, en este caso no fue necesario un proceso de selección de usuarios igual al de la sección anterior puesto que según la lógica del algoritmo, este se encarga de encontrar los vínculos que existen entre cada uno de los ítems.

De igual forma se crearon 2 conjuntos de datos: el primero para la creación de las ontologías de los usuarios y el segundo para la realización de las pruebas; se utilizó la proporción del 80 % y 20 % de las calificaciones respectivamente para la creación de cada uno de estos conjuntos.

Procedimiento.

Al igual que en los experimentos anteriores, en primer lugar se crean las ontologías de los usuarios para seguidamente, utilizando el mecanismo selector de propiedades semánticas presentado en la sección 4.6.2, introducir ocho combinaciones diferentes de propiedades semánticas en igual número de pruebas que en el experimento anterior. De igual manera, se incluyó la propiedad *Género Genre* en todas las pruebas, sin embargo, en este caso no se incluyó la propiedad IMDB puesto que no se trata de una propiedad semántica de tipo objeto lo cual imposibilita utilizarla para la realización de conexiones; las combinaciones utilizadas fueron:

1. Género.
2. Género, Escritor.
3. Género, Director.

4. Género, Escritor, Director.

5. Género, Actor.

6. Género, Actor, Escritor.

7. Género, Actor, Director.

8. Género, Actor, Director, Escritor.

Resultados.

La figura 5.5 muestra el promedio de error de todos los usuarios para cada una de las pruebas efectuadas.

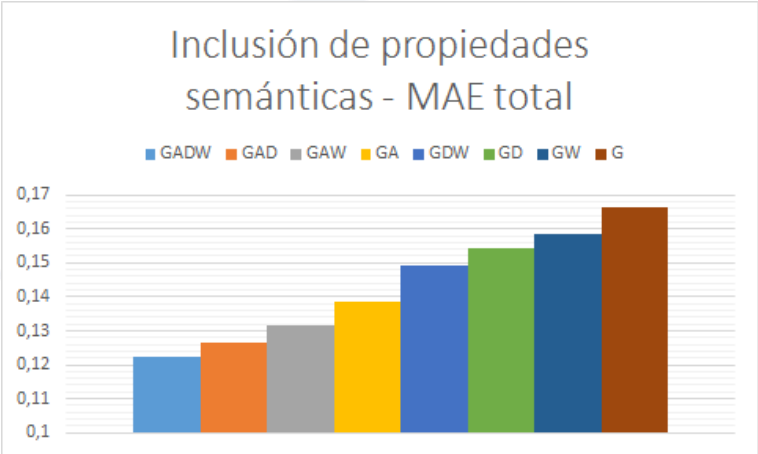


Figura 5.5: *Promedio de error de todos los usuarios en cada una de las pruebas*

Como se puede observar en la figura 5.5, en la prueba donde se utiliza únicamente la propiedad género se obtiene un error promedio de aproximadamente 16.5 % en la predicción, y, a medida que se incluyen más propiedades semánticas el error tiende a reducir, aunque en este caso la reducción depende de la propiedad; se puede observar por ejemplo, que al agregar la propiedad *Actor* a la propiedad *Género*, el error se reduce al 13.86 % lo cual representa una reducción mayor que cuando se evalúa el algoritmo con las propiedades Género, Director y Escritor lo cual produce un error de aproximadamente el 15 %. Se observa también que la propiedad *Director* en todos los casos ofrece una mayor reducción del error que la propiedad *Escritor*, finalmente al introducir todas las propiedades semánticas se obtiene el menor promedio de error el cual es el 12.2 %. En la figura 5.6 se muestra el promedio de error de cada usuario.

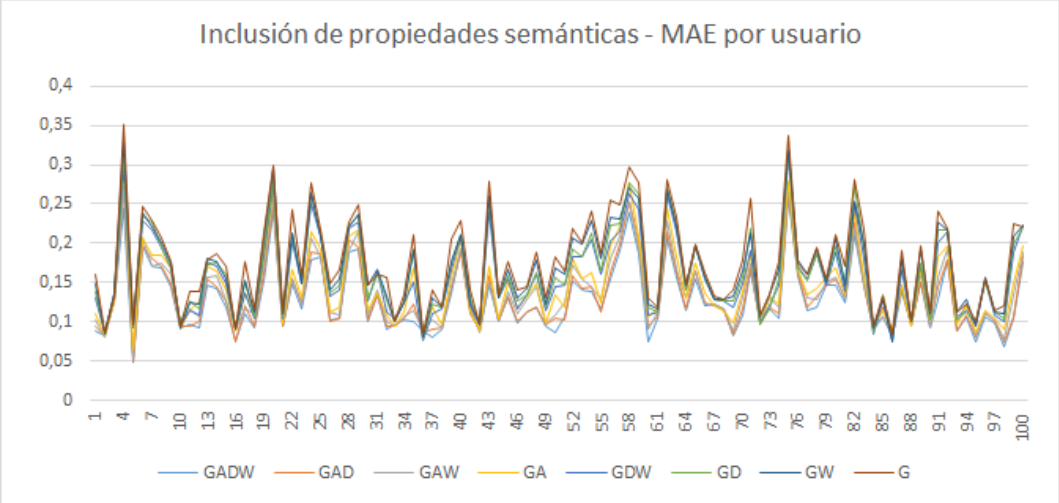


Figura 5.6: *Promedio de error por usuario en las pruebas efectuadas*

Conclusiones del experimento

Con los resultados obtenidos de la experimentación se puede concluir que la inclusión de propiedades semánticas contribuye en la reducción del error al momento de predecir una calificación, sin embargo, existen ciertas propiedades que tienen una mayor influencia, como se observa en el caso de la propiedad *Actor*, que redujo el error en mayor medida que las propiedades de *Director* y *Escritor* combinadas. Esto sugiere que la mayor reducción del error se logra con la adecuada combinación de propiedades más no necesariamente con un mayor número de ellas.

5.3.4. Impacto del uso y la retro-alimentación de los usuarios del sistema en la Estimación de las Predicciones.

En este experimento se analiza el comportamiento del error en las predicciones a medida que se incrementa el número de ítems calificados por un usuario, se intenta comprobar que el sistema mejorará la precisión de sus predicciones mientras más ítems sean valorados por dicho usuario. Este experimento posibilitará posteriormente encontrar el número mínimo de ítems que un usuario debe calificar para que se le pueda brindar una recomendación personalizada.

Conjunto de datos

Se realizaron dos pruebas con dos conjuntos diferentes de datos, en la primera se eligieron 10 usuarios que tengan más de 825 películas calificadas, mientras que para

el segundo conjunto se eligió igual número de usuarios pero se restringió su número de calificaciones a un rango de 200 a 220, en ambos casos los sets se dividen en un 80 % para la creación de las ontologías y el 20 % restante para la realización de las pruebas. En el primer caso se intentará averiguar si en algún punto, el número creciente de ítems calificados empieza a generar ruido y por ende a degradar la precisión de las calificaciones, mientras que en el segundo caso se busca una mejor apreciación de la curva de error. En este experimento se escogió un número reducido de usuario dada la complejidad computacional y la duración de las pruebas.

Procedimiento

Al igual que en las pruebas anteriores, el primer paso consiste en la creación de las ontologías de los usuarios, sin embargo, en este experimento, este proceso se diferencia de los anteriores ya que en primer lugar se crea las ontologías de los usuarios con un solo ítem calificado y se procede a la realización de las pruebas. Al finalizar las pruebas para los 10 usuarios se almacena el MAE y se vuelve a crear las ontologías pero incrementando el número de ítems calificados en 1 (uno). Nuevamente se realizan las pruebas, y se almacena el MAE. Este proceso se efectúa sucesivamente hasta alcanzar un número límite de calificaciones, las cuales en el primer caso fueron 825 y en el segundo caso 160.

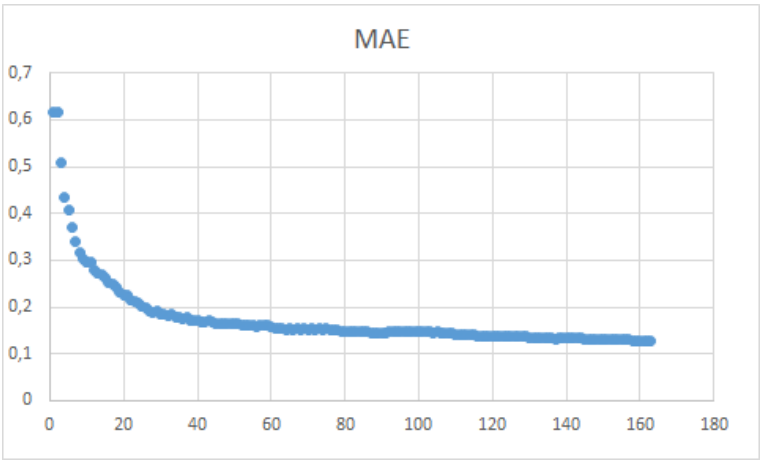


Figura 5.7: *Promedio de error a medida que se incrementan los ítems calificados 160 ítems*

Resultados.

En la figura 5.7 se presenta el promedio de error en función del incremento del número de ítems calificados con los que se crean las ontologías. En el eje de la

abscisas se tiene el número de ítems con los que se creó la ontología mientras que en el eje de las ordenadas se tiene el promedio de error o MAE. Por otra parte, en la figura 5.8 se muestra el experimento realizado a los usuarios con más de 825 ítems calificados.

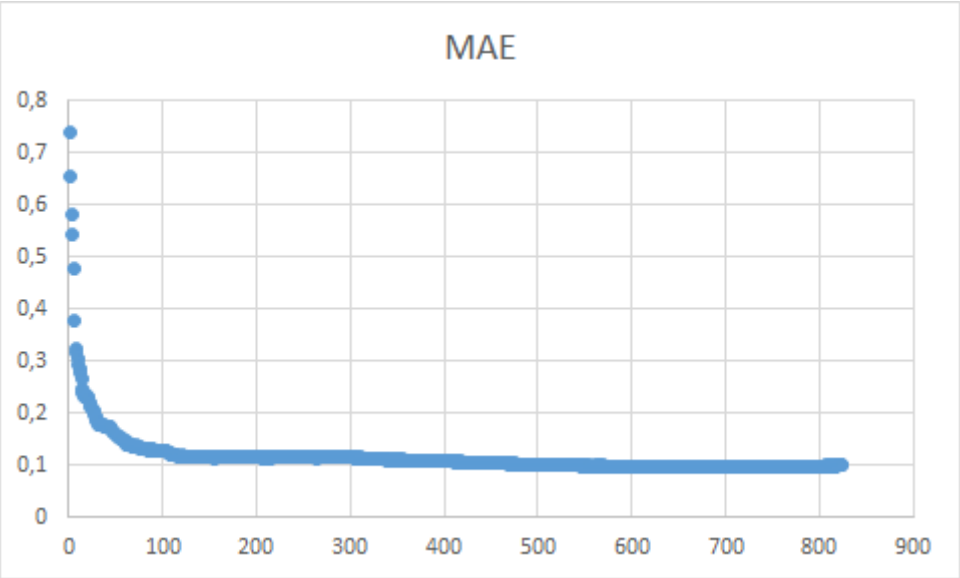


Figura 5.8: *Promedio de error a medida que se incrementan los ítems calificados, 825 ítems*

Como se puede observar, en ambos casos se describe una curva exponencial decreciente, lo cual significa que a medida que se crean las ontologías de los usuarios o en otras palabras se entrena el algoritmo con más películas calificadas el error tiende a disminuir. Por su característica exponencial esto es mucho mas evidente en los primeros casos, es decir, cuando se incrementan el número de ítems en un rango de 1 a 40, se produce una reducción de error de aproximadamente el 70 % al 20 %, para luego alcanzar cierta estabilidad. La curva en general siempre tiende a ser descendiente; se observa también en el segundo caso que el error nunca alcanza un limite mínimo de error, al menos en el intervalo analizado.

Conclusiones del Experimento

Con los resultados obtenidos de las experimentaciones se puede llegar a la conclusión que la efectividad del algoritmo siempre incrementará a medida que se incremente el número de ítems con los que se crea la ontología del usuario. Esta tendencia a la reducción se mantiene sin que la excesiva cantidad de información entorpezca el proceso de recomendación. Este comportamiento puede deberse a

que al momento de realizar las conexiones para la inferencia semántica el algoritmo escoge únicamente los datos de interés para el usuario desechando todos los datos que puedan introducir ruido.

5.3.5. Módulo de KNN

Como se explica en la sección 4.6.3, este módulo recibe como parámetros principales el número de vecinos cercanos con cuales se realizará la predicción y la distancia límite para encontrarlos. Las pruebas realizadas en este módulo, permiten estimar la distancia y número de vecinos con los cuales se logre la eficiencia máxima del algoritmo, es decir, el menor error en las predicciones.

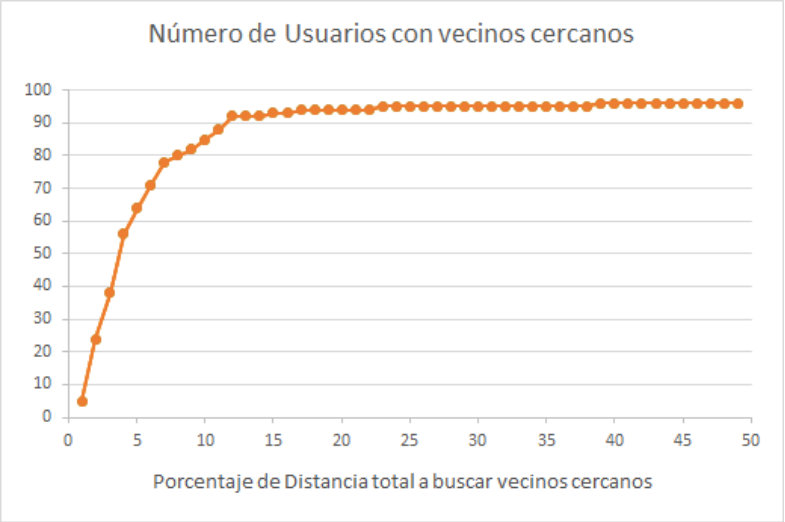


Figura 5.9: *Número de vecinos encontrados, según un porcentaje de distancia dada.*

Procedimiento

Para las pruebas se ha considerado el conjunto de datos explicado en la sección 5.2.2, con estos perfiles-ontológicos se procede a buscar vecinos para cada usuario. Específicamente, se realizaron tres pruebas, presentadas a continuación:

1. Evaluar la distancia óptima para asegurar un número de usuarios con al menos un vecino cercano.

2. Evaluar el error MAE con diferente distancia euclidiana para la obtención de vecinos cercanos.

3. Evaluar el error MAE con diferente número de vecinos para cada usuario.

Evaluar la distancia óptima para asegurar un número de usuarios con al menos un vecino cercano.

En este experimento se efectuá una búsqueda de vecinos cercanos para el conjunto de usuarios. La distancia de búsqueda comienza con el 1 % del total, hasta llegar a un nivel donde los resultados no tengan un cambio sustancial. La figura 5.9, muestra cuántos usuarios obtienen al menos un vecino según se aumenta la distancia. Como se puede observar, con el 10 % de distancia el 85 % de usuarios ya tienen al menos un vecino, por lo tanto se puede deducir que con esta distancia la mayoría de usuarios ya habrán encontrado vecinos. También se puede observar que la curva de usuarios con vecinos aumenta drásticamente en una distancia corta, y después de un 13 % aumenta muy levemente, sin presentar mayores cambios, esto indica que la distancia después de el 13 % no es relevante en el algoritmo y puede descartarse.

La figura 5.10, presenta el porcentaje de vecinos cercanos por cada usuario que obtenga al menos un vecino.

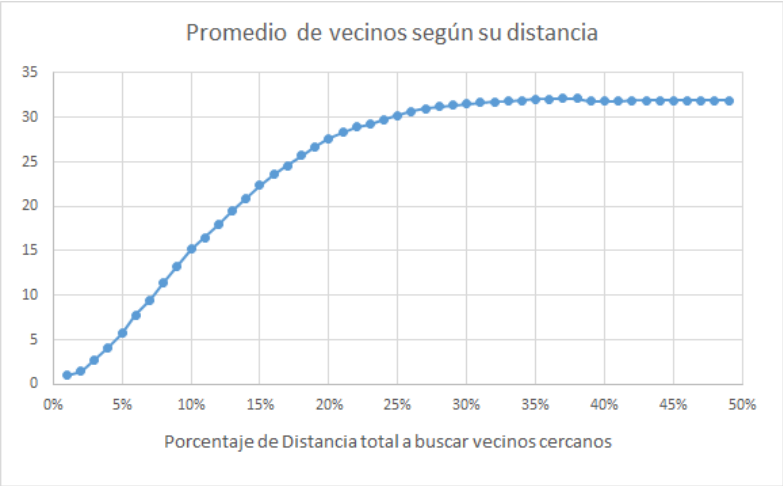


Figura 5.10: *Promedio de vecinos cercanos encontrados para aquellos usuarios con al menos un vecino.*

De la figura 5.10, se puede deducir que con en alrededor de 10 % de distancia existe un punto de inflexión en la curva, donde se puede notarse que existen aproximadamente 15 vecinos promedio por usuario. Esto confirma que la distancia obtenida en la figura 5.9 es la correcta. Después de un cierto porcentaje la curva se estabiliza, por lo que se puede decir que el parámetro de la distancia llega a ser irrelevante a partir de un punto determinado.

Evaluar el error MAE con diferente distancia euclidiana para la obtención de vecinos cercanos.

El experimento busca obtener la distancia óptima a utilizarse en el algoritmo, a partir de encontrar el error promedio (MAE) de todos los usuarios del conjunto. El MAE se obtiene para diferentes valores crecientes de la distancia, que se incrementa desde un valor base de 2 % a intervalos regulares del 2 % hasta llegar a una distancia donde el comportamiento del error tiende a estabilizarse. La figura 5.11 presenta la curva de MAE en función del incremento de la distancia euclidiana, particularmente para los 100 usuarios del conjunto. Puede notarse que una distancia equivalente a 36 % muestra un comportamiento estable que determina el intervalo máximo de observación para esta prueba.

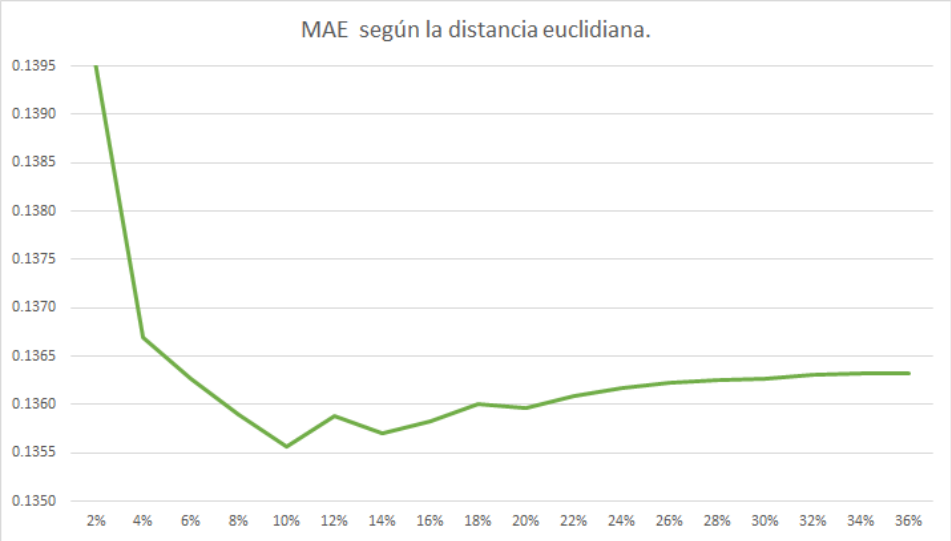


Figura 5.11: *MAE del conjunto de 100 usuarios según el aumento de la distancia euclidiana.*

La gráfica reflejada en la figura 5.11 denota que la curva del error empieza con un máximo, que disminuye a medida que aumenta la distancia. El pico más bajo se encuentra alrededor del 10 %, y a partir de ese valor, el error empieza a aumentar hasta llegar a estabilizarse.

Evaluar el error MAE con diferente número de vecinos para cada usuario.

Este experimento permite obtener el número de vecinos que refleje el mejor resultado en la salida del algoritmo. Para ello, se encuentra el MAE por cada número

de vecinos encontrados (en caso de existir más vecinos que el número especificado, se utiliza únicamente los más cercanos). Así el proceso comienza con el vecino más cercano de cada usuario, y se incrementa el número de vecinos sucesivamente hasta llegar a 30. Cabe recalcar que los usuarios para esta prueba son solamente los que tienen un mínimo de 30 vecinos en una distancia del 10 %. La figura 5.12 presenta el error promedio de todos los usuarios, en función del incremento en el número de vecinos para cada iteración.

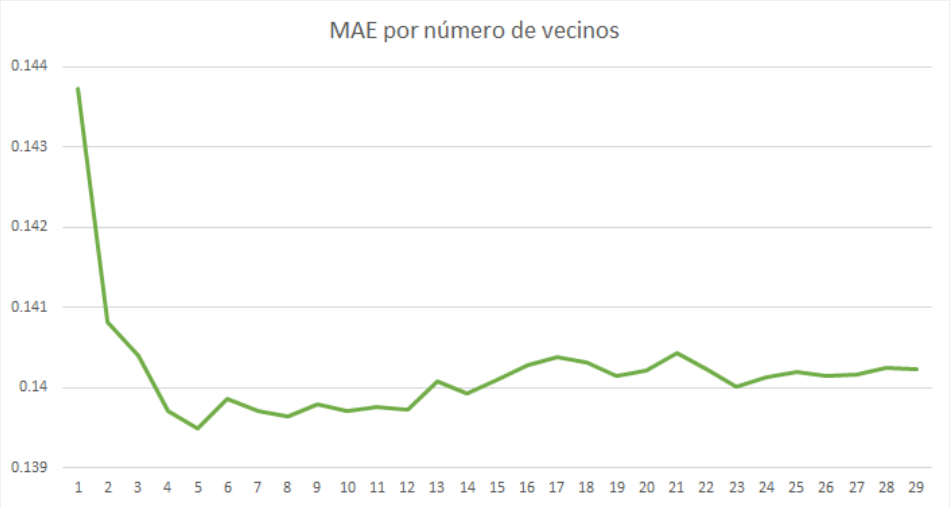


Figura 5.12: *Error promedio de predicción por cada número de vecinos por usuarios*

La figura 5.12 demuestra que el número óptimo de vecinos está determinado por la ubicación de mínimo global de la curva, específicamente: 5 vecinos. Por otra parte, puede notarse que el error con el vecino más cercano es el más alto, con lo que se puede concluir que un solo vecino no es recomendable, y que los resultados pueden mejora si se incluyen otros vecinos en el análisis. También es importante mencionar que después de 5 vecinos el rango de variación del error se estabiliza, denotando mínimas oscilaciones.

Conclusiones del las pruebas del algoritmo KNN

Con los experimentos se puede concluir, los parámetros finales con mayor eficiencia para el algoritmo son:

- Buscar vecinos al 10 % de la distancia euclidiana.
- Buscar los 5 vecinos más cercanos del usuario.

Adicionalmente, se ha demostrado que no todos los usuarios tendrán vecinos así la distancia se aumenta, con lo que se puede inferir, que existen usuarios atípicos, es decir, que simplemente no se pueden encajar en ningún estereotipo, ya que sus gustos son diferentes a los que se puede catalogar por grupos.

5.3.6. Módulo de Estereotipos.

Se realiza este experimento para determinar el número de ítems que un usuario debe calificar para que pueda recibir una recomendación personalizada, es decir para que abandone el esquema de recomendación por estereotipos. Para ello se utilizarán los resultados obtenidos en el experimento de la sección 5.3.4 y se tratará de encontrar un punto de cruce entre la línea del MAE de la recomendación por estereotipos y la curva de decrecimiento del error a medida que se incrementan las películas (figura 5.7).

Conjunto de datos

Para esta prueba se han elegido 100 usuarios que tengan un rango de entre 200 y 220 ítems calificados, en este caso no se ha creado un conjunto de datos de entrenamiento puesto que las recomendaciones serán basadas en su estereotipo y no en sus preferencias. Del total de ítems calificados se ha tomado el 20 % para la realización de las pruebas.

Procedimiento

Al tratarse de recomendaciones por estereotipos no es necesario crear las ontologías de los usuarios para evaluar el sistema, sin embargo, se han creado las ontologías de los 14 estereotipos descritos en la sección 4.6.1. Cuando se realiza una predicción, el módulo de estereotipos verifica a cuál de ellos pertenece el usuario y ejecuta el algoritmo recomendador enviando como dato de entrada la ontología del estereotipo correspondiente para que finalmente el sistema realice las predicciones y se evalúe el MAE obtenido.

Resultados del Experimento

En la figura 5.13 se presenta el resultado del MAE de la recomendación por estereotipos, mientras que la figura 5.14 muestra el cruce de este MAE con la curva de decrecimiento del error mostrada en el experimento anterior.

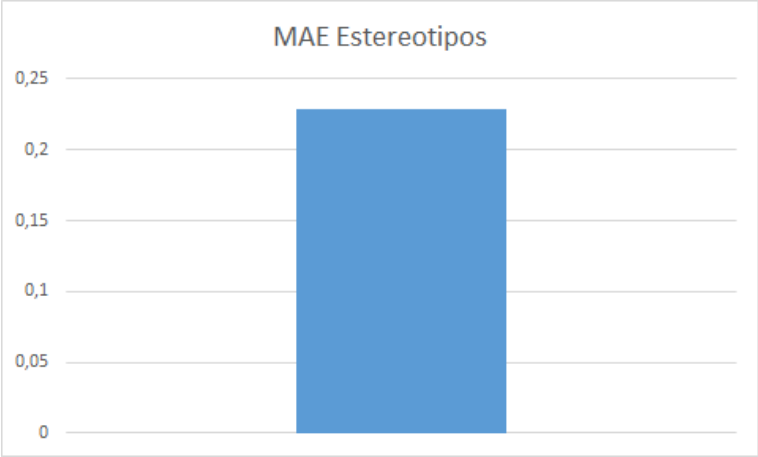


Figura 5.13: *MAE de los estereotipos*

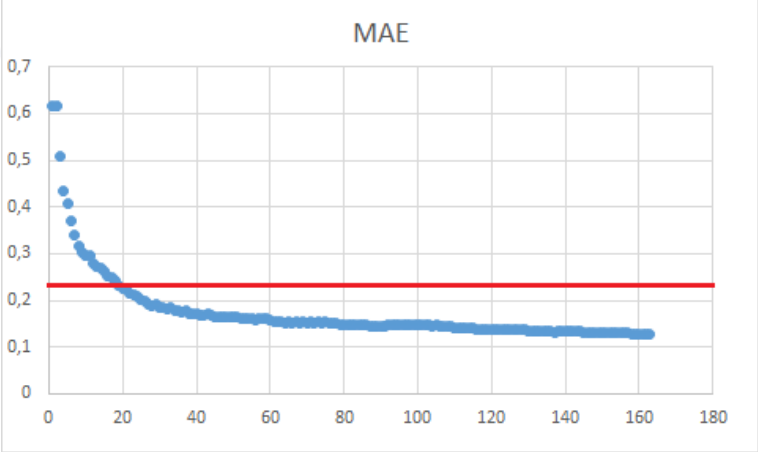


Figura 5.14: *Cruce de las gráficas*

Como se puede observar en las gráficas, el MAE de la recomendación por estereotipos es de alrededor del 22 %. Esta recta intersecta la curva de reducción del error en aproximadamente 20 películas. Se puede observar que superado este número, el error en las recomendaciones personalizadas será menor que el obtenido por los estereotipos, por lo tanto, se puede concluir que superado ese número de ítems calificados, se puede retirar al usuario de la calificación por estereotipos y brindarle calificaciones personalizadas.

5.3.7. Comparación del algoritmo de recomendación por inferencia semántica con respecto a KNN

Este experimento trata de una comparación entre las dos técnicas de recomendación mencionadas, con el objetivo de verificar cuál de ellas tiene una mayor efectividad al momento de realizar la predicción de la calificación para un ítem dado. Con este fin en mente, del conjunto de datos se han elegido cien usuarios que tengan un rango de 200 a 220 ítems calificados, asegurando de esta manera que hayan suficientes datos tanto para el entrenamiento del algoritmo como para el test, además, al no existir ningún parámetro de ordenamiento en los usuarios se asegura su variedad en preferencias.

El conjunto se subdividió en entrenamiento y test, usando un total del 80 % de las calificaciones otorgadas por el usuario para la creación de su ontología y el 20 % restante para el archivo de test. Se crean entonces, las ontologías de cada usuario y al final se procede a ejecutar los dos algoritmos con el mismo archivo de pruebas, es decir, que se ejecutan en igualdad de condiciones. Cada algoritmo calculará el error en la predicción, calculará un promedio por usuario y un promedio general para el total de usuarios.

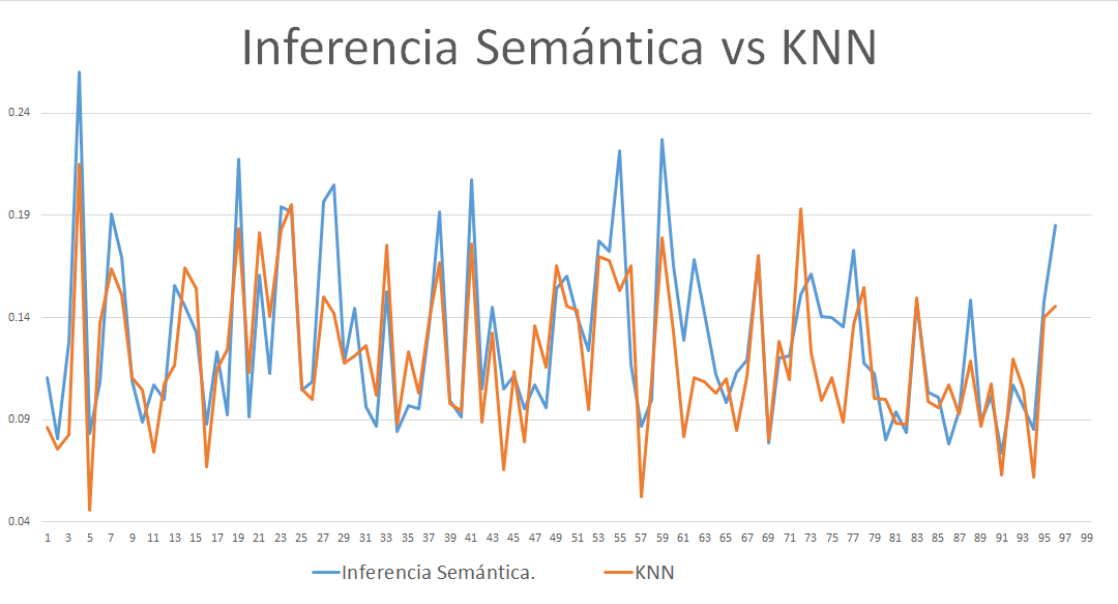


Figura 5.15: *Gráfico del MAE por usuarios, Inferencia semántica vs* KNN

Resultados del experimento

En la figura 5.15 se presenta el MAE por cada usuario evaluado. Se puede observar que el error describe un comportamiento parecido tanto al utilizar el algoritmo de inferencia semántica, representada en azul, como con el modulo KNN, representada en naranja. En estas curvas se puede observar que en algunos casos tiene mejor rendimiento el algoritmo de inferencia (ejemplo: Usuario No.20), pero en otras situaciones es mejor el KNN (ejemplo: Usuario No.5).

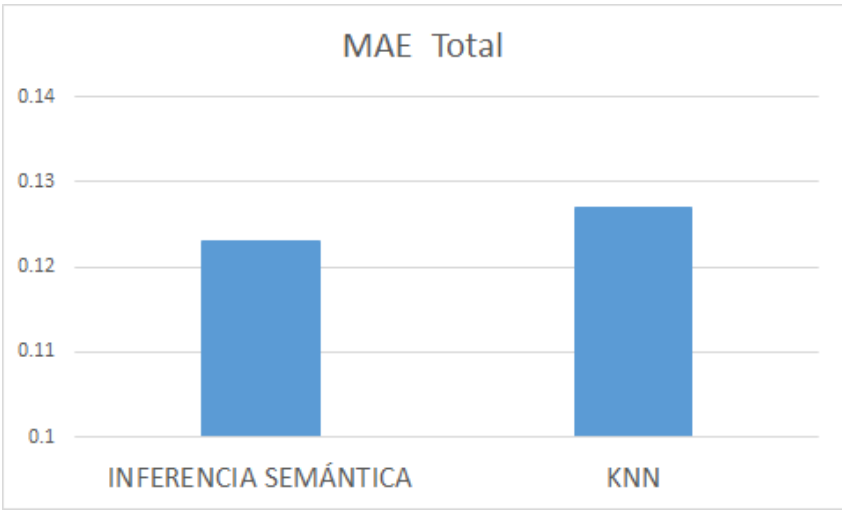


Figura 5.16: *Gráfico del MAE Total del conjunto de usuarios, Inferencia semántica vs* KNN

En la figura 5.16, mientras tanto, se puede deducir que los dos algoritmos tienen aproximadamente el mismo resultado, presentando una sutil diferencia, en la que ligeramente el algoritmo de inferencia semántica tiene un mejor desempeño.

Conclusiones del experimento

Con los resultados presentados, se puede deducir que ambos algoritmos son altamente eficaces, y que con cada usuario varía su desempeño. De estos resultados, ha surgido la idea de crear un algoritmo mixto, que sea capaz de elegir el mejor algoritmo a utilizarse para cada usuario, según su propia conveniencia.

5.4. Análisis de resultados: Evaluación cualitativa.

En la actualidad la mayoría de los algoritmos de recomendación se centran en mejorar la exactitud del sistema de recomendación. Para evaluar su precisión es esencial que el sistema de recomendación cubra diferentes facetas con el fin de hacer que el sistema sea más diverso, integral y universal. Para ello, se puede realizar pruebas de precisión, cobertura, diversidad, escalabilidad, adaptabilidad, riesgo, etc, como se describe en [47].

En el caso de este proyecto, este tipo de pruebas no pueden realizarse en su totalidad, ya que se parte de una base de datos de usuarios de prueba, y para extraer la parte cualitativa de la evaluación se requiere de etapas prolongadas de exposición y uso del sistema ante usuarios reales, por ejemplo, funcionando en el entorno de la TV Digital. Por los motivos expuestos, esta sección plantea los mecanismos de evaluación cualitativa a nivel teórico, dejando la puerta abierta a procesos de análisis en futuros proyectos.

5.4.1. Tipos de evaluación cualitativa.

Para realizar las evaluaciones se debe probar varias situaciones en distintas circunstancias, por este motivo existen un sin número de pruebas que se pueden efectuar, categorizandose fundamentalmente en dos grandes grupos:

- Evaluación desde la perspectiva del sistema.
- Evaluación desde la perspectiva del usuario.

Evaluación desde la perspectiva del sistema

Este tipo de pruebas se realiza con el fin de encontrar fallas en el algoritmo, en condiciones normales de funcionamiento, los enfoques principales para estas evaluaciones son [48]:

- **Confianza.** Se define como la exactitud que tiene en cada predicción, entendiéndose que si un sistema otorga recomendaciones exactas entonces se convierte en un sistema confiable.

- **Escalabilidad.** Al tener conjuntos de datos sumamente extensos, un sistema puede consumir grandes recursos computacionales. El nivel de escalabilidad depende de qué tan útil es el algoritmo de acuerdo a los recursos disponibles.
- **Adaptabilidad.** La adaptabilidad es una evaluación que trata de medir si el sistema funciona en condiciones reales, esto depende según el ámbito que se encuentre aquel sistema.

Evaluación desde la perspectiva del usuario.

Las evaluaciones de este tipo tratan básicamente de diagnosticar el sistema de acuerdo a la opinión explícita del usuario, estas pruebas son las más efectivas. Para estas pruebas se toman en cuenta [48]:

- **Preferencias.** Es un indicador que describe esencialmente cuánto cada usuario está satisfecho con el sistema, es decir, una medida de satisfacción de las recomendaciones que realiza el sistema.
- **Confianza.** El usuario brinda una opinión de si las recomendaciones que se están otorgando son correctas y qué nivel de confianza tiene en las mismas.

5.4.2. Planteamiento a futuro.

Cuando el sistema esté funcionando con condiciones reales, la evaluación cualitativa podrá ser realizada. Sin embargo, desde la *perspectiva del sistema*, según se indicó en la sección 5.4.1, el sistema puede ser probado en uso, evaluando resultados como: Tiempos de predicción, estabilidad, resultados de eficacia, etc. De esta manera se pretende tener una idea de cómo se comporta el sistema, con el fin de optimizarlo.

Para la *perspectiva del usuario*, sección 5.4.1, se pretende realizar sondeos a los usuarios actuales para que brinden opiniones verdaderas sobre las ventajas de uso del sistema y su funcionamiento.

Capítulo 6

Conclusiones y Futuras lineas de Investigación.

6.1. Conclusiones.

A lo largo de este proyecto se han analizado y desarrollado los componentes de un sistema de recomendación basado en tecnologías semánticas orientado a la recomendación de los contenidos audiovisuales enfocados al entorno de la Televisión Digital. Se ha estudiado el comportamiento del sistema al variar ciertos parámetros en sus entradas, en el proceso de recomendación, y en la utilización de módulos complementarios. Por otra parte, se ha realizado un estudio del estado del arte de los sistemas de recomendación semánticos exponiendo la orientación y las técnicas de recomendación de algunos de éstos sistemas encontrados en la literatura y que resumen el esfuerzo en la comunidad científica por buscar alternativas que permitan reducir la sobrecarga de información en los usuarios. Finalizado el proyecto actual se ha podido comprobar que los objetivos específicos planteados al principio del mismo se han cumplido en su totalidad, como se presenta a continuación:

1. Reconocerlas diferentes tecnologías utilizadas en los Sistemas de recomendación Semánticos - SRS:
   Se ha encontrado que los sistemas de recomendación semánticos utilizan las tecnologías propuestas por los estándares de la W3C para la representación del conocimiento es decir RDF, RDFS y OWL, por otro lado, la gran mayoría utiliza la librería de Apache Jena para la manipulación de las ontologías e inferencia de conocimiento. En este contexto, el análisis de esta tecnología ha permitido ganar una visión más amplia de los sistemas de recomendación y su aporte en el filtrado de grandes volúmenes de información.

2. Determinar el estado del arte de los Sistema de Recomendación SR y comparar los diferentes enfoques presentados hasta la fecha:
   Se ha realizado una revisión del estado de arte de los sistemas de recomendación encontrando que en la literatura principalmente se los divide en dos enfoques, siendo estos los sistemas de recomendación colaborativos y los basados en el contenido, además, se mencionó el dominio y las técnicas de algunos sistemas de recomendación semánticos implementados en distintos entornos académicos, en los que se observó que, entre las principales técnicas de recomendación se encuentran: la inferencia semántica, los árboles de decisiones, y similaridad semántica. Por otro lado se nombraron varios dominios a los cuales están orientados estos sistemas, entre ellos se tienen: el turismo, los contenidos audiovisuales, sistemas recomendadores de sitios web o sistemas orientados a grupos de usuarios, comercio electrónico, etc.

3. Analizar y utilizar las ontologías del perfil de usuario y guías de programación como entradas que alimenten al SRS a desarrollarse:
   Se ha analizado el proceso de construcción de las ontologías y se han utilizado para alimentar las entradas del sistema. En este proyecto, se manejó una ontología que modela las preferencias del usuario, y una segunda, para la representación de contenidos audiovisuales.

4. Evaluar la respuesta y los diferentes comportamientos del algoritmo de recomendación para diversos parámetros de ajuste:
   Se ha realizado una serie de pruebas en las cuales se determinan y ajustan parámetros específicos tanto en el algoritmo de recomendación, así como en los módulos complementarios diseñados, para establecer un entorno de evaluación adecuado para el sistema. Así también, se ha analizado el comportamiento del error o MAE al realizar dichas modificaciones como métrica de desempeño en la determinación de los parámetros que minimicen el error.

5. Comparar los resultados obtenidos con el objeto de encontrar el algoritmo de recomendación (o combinación de ellos) con mejor desempeño:
   Con los resultados obtenidos en cada una de las pruebas de los algoritmos estudiados se ha logrado determinar el algoritmo de recomendación con el mejor desempeño, conjuntamente con la mejor combinación de parámetros de ajuste que posibilitan obtener un menor promedio de error o MAE en la predicción de la valoración de un ítem determinado. Se entonces, la utilización del algoritmo de recomendación por inferencia semántica detallado

en la sección 4.5.2, el cual genera recomendaciones utilizando el módulo de estereotipos hasta que el usuario alcance un mínimo de 20 ítems calificados. Paralelamente se sugiere la utilización del algoritmo de recomendación por vecinos cercanos o KNN para los usuarios en los cuales el MAE en la predicción por esta técnica es inferior al MAE en la predicción por recomendación personalizada, en lo que se denomina un algoritmo híbrido conmutado.

6.2. Futuras lineas de investigación.

Existen algunas líneas de trabajo que pueden ser abordadas a futuro por este proyecto, que se presentan a continuación:

- Creación de un módulo o mecanismo de renovación de las preferencias del usuario. Cuya importancia y necesidad surge con el paso del tiempo y la natural tendencia del usuario de modificar sus gustos, con lo cual se evitaría la generación de recomendaciones antiguas u obsoletas. Este mecanismo, reduciría paulatinamente el valor de las preferencias más antiguas, para que de esta manera el recomendador dé más importancia a los contenidos sobre los que recientemente el usuario ha demostrado interés.
- Estudio de un mecanismo de recomendación para grupos de usuarios, donde se enfoque el escenario en el que más de un usuario se sienta frente a la DTV y el sistema debería ser capaz de recomendar una programación apropiada para dicho grupo.
- Implementación de un mecanismo de diferenciación entre preferencias implícitas y explicitas con las cuales el sistema dará una mayor preferencia a las programaciones en las cuales el usuario ha mostrado su interés explícitamente.

ANEXOS

Anexo A

Tablas de Resultados

A.1. Inferencia Semántica vs Dispersión

Valores numéricos de la figura: 5.1

Usuario	MAED	MAEI	Usuario	MAED	MAEI
1	0,1360836	0,08143418	51	0,15541804	0,10719387
2	0,12430376	0,10472713	52	0,24278016	0,16346998
3	0,14154979	0,12562299	53	0,19953325	0,15056653
4	0,32648982	0,2469958	54	0,22010328	0,13277423
5	0,11275789	0,05409994	55	0,18144156	0,12427311
6	0,22684709	0,19543598	56	0,21371649	0,15473115
7	0,22415928	0,13016543	57	0,27864505	0,20093147
8	0,2149443	0,17178741	58	0,30526133	0,25266875
9	0,19106002	0,14798982	59	0,27935339	0,18687758
10	0,11112542	0,09359523	60	0,11105798	0,07840611
11	0,13338701	0,10115645	61	0,11873713	0,10297533
12	0,12578206	0,09945713	62	0,30232015	0,19270495
13	0,17691586	0,1340375	63	0,24996667	0,16007101
14	0,17395829	0,13508439	64	0,15873509	0,11540658
15	0,18526788	0,12321238	65	0,25231684	0,16652703
16	0,11082552	0,08441248	66	0,1753917	0,12316952
17	0,17539364	0,11085481	67	0,18184625	0,12501139
18	0,15194697	0,09715775	68	0,10304963	0,11499875
19	0,20967398	0,15043785	69	0,15513986	0,08120313
20	0,32869291	0,24055018	70	0,16052101	0,09964409

Sigue en la página siguiente.

Usuario	MAED	MAEI	Usuario	MAED	MAEI
21	0,09344399	0,07836917	71	0,21062926	0,16388496
22	0,22183149	0,15206459	72	0,11556994	0,09791202
23	0,13743294	0,12111919	73	0,17344951	0,12072109
24	0,26778703	0,18255135	74	0,16662343	0,10271062
25	0,24999305	0,17965493	75	0,41334993	0,29941531
26	0,13474388	0,10333626	76	0,20639092	0,16215565
27	0,13015479	0,10709484	77	0,17554058	0,12500398
28	0,2780684	0,19365869	78	0,20078028	0,12344556
29	0,24446258	0,19570031	79	0,18218579	0,1463873
30	0,16974417	0,11368599	80	0,219046	0,15996585
31	0,2104281	0,13726748	81	0,16980342	0,12577036
32	0,14448785	0,09337476	82	0,25944059	0,20979326
33	0,09864939	0,09527953	83	0,17437104	0,15008645
34	0,17386518	0,09774566	84	0,12091546	0,08908773
35	0,18971345	0,09801843	85	0,1288493	0,09426079
36	0,0926641	0,08559452	86	0,0954815	0,08499812
37	0,13774282	0,0801043	87	0,16888165	0,1315908
38	0,11627023	0,0907754	88	0,11194586	0,09971918
39	0,16664517	0,13243879	89	0,14568074	0,12086877
40	0,24320617	0,19592519	90	0,10680657	0,09692877
41	0,1265877	0,1148894	91	0,2088705	0,13201407
42	0,11899162	0,09242622	92	0,22353955	0,17073134
43	0,32321284	0,15242685	93	0,1198646	0,08519773
44	0,14088947	0,10216205	94	0,13219941	0,10471954
45	0,17858797	0,13480883	95	0,12073495	0,07966854
46	0,13825888	0,10064857	96	0,17257363	0,09161735
47	0,14787382	0,11617489	97	0,12985364	0,0969552
48	0,17090841	0,12298583	98	0,09865393	0,06673356
49	0,1302825	0,09508238	99	0,20531841	0,1062545
50	0,17693348	0,08845667	100	0,25989791	0,18344899

Tabla A.1: *Comparativa del algoritmos de inferencia semántica vs dispersión.*

A.2. Impacto de las propiedades Semánticas en la Estimación de las Predicciones usando el algoritmo por dispersión.

Valores numéricos de la figura: 5.3

Usuarios	G	GA	GI	GAD	GADW	GDW	GADWI
1	0,2324	0,0694	0,0981	0,0637	0,0631	0,0663	0,0603
2	0,3229	0,0971	0,1500	0,0767	0,0706	0,0712	0,0617
3	0,2404	0,0689	0,1097	0,0583	0,0933	0,0594	0,0516
4	0,3194	0,1289	0,1449	0,1186	0,1130	0,1151	0,1125
5	0,2528	0,1058	0,1747	0,0845	0,0698	0,0827	0,0750
6	0,1432	0,0711	0,1196	0,0576	0,0478	0,0522	0,0583
7	0,1510	0,0643	0,1041	0,0528	0,0435	0,0507	0,0470
8	0,1532	0,1002	0,1480	0,0803	0,0612	0,0736	0,0634
9	0,1396	0,0748	0,1178	0,0601	0,0490	0,0556	0,0486
10	0,1311	0,0900	0,1324	0,0765	0,0615	0,0696	0,0675
11	0,2122	0,1253	0,2089	0,1035	0,0867	0,0977	0,0910
12	0,1717	0,1308	0,1511	0,1172	0,1104	0,1172	0,0937
13	0,1136	0,0754	0,1105	0,0612	0,0536	0,0587	0,0545
14	0,2233	0,1705	0,2111	0,1529	0,1440	0,1529	0,1343
15	0,1335	0,0901	0,1322	0,0742	0,0611	0,0654	0,0644
16	0,1544	0,0989	0,1457	0,0796	0,0705	0,0798	0,0852
17	0,1225	0,0759	0,1169	0,0604	0,0497	0,0566	0,0497
18	0,2005	0,1212	0,1900	0,1069	0,0889	0,1010	0,0930
19	0,2778	0,1042	0,1640	0,0891	0,0703	0,0848	0,0779
20	0,2069	0,1673	0,1932	0,1579	0,2614	0,1711	0,2127
21	0,0875	0,0639	0,0836	0,0498	0,0560	0,0600	0,0625
22	0,1694	0,1029	0,1556	0,0870	0,0691	0,0776	0,0726
23	0,1607	0,0978	0,1553	0,0878	0,0707	0,0848	0,0734
24	0,1068	0,0743	0,1063	0,0698	0,0595	0,0641	0,0607
25	0,1802	0,1192	0,1737	0,1021	0,0859	0,0978	0,0922
26	0,1486	0,1016	0,1468	0,0921	0,0754	0,0814	0,0836
27	0,2001	0,1051	0,1601	0,0776	0,0660	0,0755	0,0681
28	0,3039	0,0938	0,1309	0,0650	0,0524	0,0644	0,0619

Sigue en la página siguiente.

Usuarios	G	GA	GI	GAD	GADW	GDW	GADWI
29	0,3102	0,1079	0,1667	0,0890	0,0660	0,0878	0,0779
30	0,1217	0,1746	0,1238	0,0693	0,0635	0,0654	0,0688
31	0,2139	0,0981	0,1698	0,0908	0,0776	0,0887	0,0822
32	0,3135	0,1052	0,1407	0,0835	0,0747	0,0793	0,0790
33	0,3652	0,1100	0,4678	0,0949	0,0805	0,0856	0,0771
34	0,2406	0,1114	0,3842	0,0916	0,0776	0,0866	0,0763
35	0,2597	0,0940	0,1812	0,0751	0,0561	0,0700	0,0638
36	0,1613	0,0783	0,2459	0,0674	0,0550	0,0614	0,0611
37	0,4136	0,1150	0,1913	0,0916	0,0770	0,0891	0,0877
38	0,2785	0,0886	0,1226	0,0615	0,0486	0,0529	0,0502
39	0,1985	0,1317	0,1881	0,1056	0,0916	0,0979	0,0908
40	0,2813	0,1168	0,1705	0,0914	0,0793	0,0835	0,0808
41	0,2176	0,1510	0,2200	0,1311	0,1209	0,1385	0,1339
42	0,3331	0,0811	0,1180	0,0663	0,0567	0,0617	0,0625
43	0,1231	0,0794	0,1174	0,0727	0,0658	0,0782	0,0734
44	0,1513	0,0940	0,1547	0,0767	0,0619	0,0711	0,0689
45	0,1876	0,1127	0,1826	0,0974	0,0799	0,0890	0,0895
46	0,0800	0,0519	0,1850	0,0427	0,0381	0,0432	0,1421
47	0,1611	0,1338	0,1542	0,1247	0,1201	0,1247	0,1235
48	0,1467	0,0801	0,1101	0,0705	0,0591	0,0626	0,0646
49	0,1356	0,0856	0,1233	0,0670	0,0635	0,0628	0,0719
50	0,1297	0,0893	0,1275	0,0746	0,0646	0,0763	0,0696
51	0,1498	0,0923	0,1366	0,0789	0,0787	0,1072	0,0725
52	0,1671	0,1081	0,1680	0,0892	0,0726	0,0858	0,0747
53	0,1499	0,1020	0,2602	0,0852	0,0693	0,0752	0,0722
54	0,1129	0,0745	0,1084	0,0666	0,0611	0,0653	0,0565
55	0,1548	0,2053	0,1548	0,1121	0,1065	0,1139	0,1095
56	0,1538	0,1123	0,1564	0,0806	0,0655	0,0713	0,0704
57	0,2161	0,1561	0,2196	0,1257	0,1117	0,1178	0,1280
58	0,1399	0,0940	0,2269	0,0810	0,0707	0,0723	0,0723
59	0,1219	0,0710	0,2734	0,0604	0,0542	0,0595	0,0524
60	0,2145	0,1348	0,3270	0,1045	0,0968	0,0993	0,0932
61	0,1660	0,1474	0,1534	0,1396	0,1358	0,1385	0,1351
62	0,2070	0,0919	0,2921	0,0785	0,0704	0,0770	0,0691

Sigue en la página siguiente.

Usuarios	G	GA	GI	GAD	GADW	GDW	GADWI
63	0,3100	0,1255	0,2538	0,0928	0,0840	0,0891	0,0834
64	0,1687	0,1427	0,1722	0,1371	0,1223	0,1227	0,1274
65	0,1307	0,0769	0,1218	0,0642	0,0526	0,0590	0,0607
66	0,1455	0,1165	0,1336	0,1065	0,1001	0,1050	0,0909
67	0,1098	0,0733	0,1094	0,0670	0,0575	0,0639	0,0573
68	0,1837	0,1115	0,1837	0,0853	0,0699	0,0849	0,0699
69	0,2064	0,0934	0,1486	0,0717	0,0610	0,0698	0,0609
70	0,1717	0,1059	0,1667	0,0805	0,0679	0,0799	0,0758
71	0,1073	0,0842	0,1057	0,0749	0,0710	0,0712	0,0751
72	0,1819	0,1177	0,1819	0,0941	0,0847	0,0989	0,0847
73	0,1687	0,1102	0,1584	0,0955	0,0793	0,0882	0,0852
74	0,1308	0,0940	0,1318	0,0784	0,0711	0,0725	0,0728
75	0,1394	0,0837	0,1625	0,0609	0,0575	0,0662	0,0952
76	0,2435	0,0716	0,1005	0,0595	0,0500	0,0563	0,0470
77	0,1768	0,1296	0,1681	0,1086	0,0981	0,0966	0,1043
78	0,1532	0,0857	0,1537	0,0647	0,0532	0,0654	0,0539
79	0,1691	0,1408	0,1662	0,1343	0,1290	0,1390	0,1281
80	0,1082	0,0707	0,1061	0,0644	0,0531	0,0600	0,0537
81	0,1245	0,0750	0,1216	0,0655	0,0533	0,0619	0,0524
82	0,1137	0,0802	0,1093	0,0681	0,0577	0,0617	0,0599
83	0,2184	0,1309	0,2172	0,0988	0,0795	0,0888	0,0884
84	0,2038	0,1646	0,2003	0,1553	0,1453	0,1535	0,1461
85	0,2203	0,1588	0,2155	0,1374	0,1267	0,1365	0,1312
86	0,1365	0,0877	0,1578	0,0735	0,0645	0,0738	0,0932
87	0,2898	0,2062	0,2926	0,1676	0,1507	0,1646	0,1589
88	0,1369	0,0854	0,1356	0,0803	0,0695	0,0860	0,0676
89	0,1612	0,1106	0,1612	0,0837	0,0747	0,0911	0,0747
90	0,1350	0,0837	0,1292	0,0708	0,0565	0,0654	0,0550
91	0,0813	0,0544	0,0782	0,0433	0,0376	0,0386	0,0387
92	0,2172	0,1394	0,2156	0,1191	0,0964	0,1080	0,1082
93	0,2020	0,1282	0,1910	0,1102	0,0914	0,1024	0,1006
94	0,1897	0,1606	0,1897	0,1499	0,1467	0,1521	0,1467
95	0,5659	0,1308	0,1539	0,0764	0,0636	0,0705	0,0674
96	0,1140	0,1386	0,1125	0,0631	0,0570	0,0603	0,0879

Sigue en la página siguiente.

Usuarios	G	GA	GI	GAD	GADW	GDW	GADWI
97	0,4279	0,0435	0,1356	0,0573	0,0456	0,0544	0,0477
98	0,1792	0,2534	0,1654	0,1013	0,0946	0,1013	0,0977
99	0,2205	0,1060	0,2895	0,1124	0,0933	0,1098	0,0938
100	0,1124	0,0568	0,2891	0,0649	0,0579	0,0654	0,0633
MAE:	0,1926	0,0996	0,1858	0,0831	0,0703	0,0779	0,0748

Tabla A.2: *Comparación de MAE de los 100 usuarios para las diferentes combinaciones en el algoritmo por dispersión.*

A.3. Impacto de las propiedades semánticas en el algoritmo de inferencia semántica.

Valores numéricos de la figura: 5.5.

Usuario	GADW	GAD	GAW	GA	GDW	GD	GW	G
1	0,0879	0,0945	0,1027	0,1104	0,1301	0,1377	0,1484	0,1609
2	0,0834	0,0826	0,0809	0,0832	0,0857	0,0858	0,0843	0,0861
3	0,1261	0,1263	0,1277	0,1285	0,1339	0,1318	0,1363	0,1325
4	0,2441	0,2725	0,2690	0,2962	0,2974	0,3153	0,3299	0,3513
5	0,0478	0,0498	0,0545	0,0645	0,0925	0,0951	0,0959	0,0973
6	0,1970	0,1960	0,2073	0,2072	0,2283	0,2358	0,2389	0,2463
7	0,1713	0,1806	0,1707	0,1842	0,2166	0,2264	0,2209	0,2292
8	0,1683	0,1701	0,1744	0,1840	0,1944	0,1958	0,2018	0,2060
9	0,1443	0,1477	0,1608	0,1697	0,1680	0,1699	0,1753	0,1773
10	0,0928	0,0938	0,0908	0,0929	0,0962	0,0994	0,0930	0,0954
11	0,0964	0,0947	0,1174	0,1139	0,1145	0,1267	0,1243	0,1396
12	0,0929	0,1006	0,1061	0,1273	0,1087	0,1173	0,1217	0,1379
13	0,1457	0,1552	0,1560	0,1710	0,1758	0,1735	0,1800	0,1783
14	0,1419	0,1450	0,1588	0,1647	0,1705	0,1741	0,1775	0,1873
15	0,1196	0,1258	0,1274	0,1403	0,1492	0,1600	0,1550	0,1702
16	0,0768	0,0742	0,0899	0,0883	0,0906	0,0901	0,0928	0,0915
17	0,1104	0,1202	0,1172	0,1361	0,1359	0,1514	0,1521	0,1763
18	0,0929	0,0944	0,1098	0,1127	0,1054	0,1064	0,1131	0,1170
19	0,1531	0,1618	0,1733	0,1808	0,1874	0,1933	0,2077	0,2135
20	0,2407	0,2488	0,2564	0,2683	0,2752	0,2827	0,2946	0,2988
21	0,0951	0,0944	0,0983	0,0976	0,1044	0,1062	0,1116	0,1122
22	0,1488	0,1556	0,1544	0,1676	0,2033	0,2136	0,2120	0,2423
23	0,1158	0,1222	0,1239	0,1300	0,1477	0,1527	0,1478	0,1592
24	0,1788	0,1896	0,2076	0,2148	0,2504	0,2638	0,2656	0,2779
25	0,1824	0,1859	0,1838	0,1921	0,2087	0,2116	0,2094	0,2164
26	0,1024	0,1006	0,1119	0,1127	0,1331	0,1367	0,1409	0,1481
27	0,1049	0,1056	0,1095	0,1177	0,1398	0,1461	0,1556	0,1670
28	0,1889	0,2042	0,1901	0,2085	0,2183	0,2238	0,2215	0,2279
29	0,1924	0,1951	0,2141	0,2161	0,2260	0,2357	0,2377	0,2492
30	0,0998	0,1029	0,1081	0,1141	0,1272	0,1290	0,1459	0,1463

Sigue en la página siguiente.

Usuario	GADW	GAD	GAW	GA	GDW	GD	GW	G
31	0,1352	0,1340	0,1403	0,1374	0,1656	0,1609	0,1658	0,1609
32	0,0908	0,0939	0,1013	0,1084	0,1120	0,1287	0,1331	0,1559
33	0,0972	0,0948	0,0937	0,0946	0,1041	0,1001	0,1013	0,0996
34	0,1033	0,1051	0,1070	0,1130	0,1243	0,1266	0,1241	0,1337
35	0,1004	0,1235	0,1137	0,1695	0,1504	0,1921	0,1902	0,2109
36	0,0884	0,0860	0,0901	0,0881	0,0763	0,0776	0,0812	0,0841
37	0,0806	0,0915	0,1018	0,1185	0,1114	0,1225	0,1302	0,1413
38	0,0897	0,0919	0,0929	0,0966	0,1176	0,1193	0,1185	0,1189
39	0,1369	0,1429	0,1450	0,1577	0,1671	0,1738	0,1765	0,2039
40	0,1922	0,1897	0,1919	0,2041	0,2096	0,2117	0,2099	0,2293
41	0,1145	0,1116	0,1194	0,1143	0,1211	0,1314	0,1256	0,1402
42	0,0883	0,0901	0,0866	0,0887	0,0966	0,0964	0,0972	0,0965
43	0,1463	0,1592	0,1511	0,1705	0,2418	0,2575	0,2618	0,2787
44	0,1018	0,0996	0,0997	0,1044	0,1299	0,1303	0,1305	0,1356
45	0,1300	0,1325	0,1390	0,1511	0,1538	0,1595	0,1676	0,1768
46	0,0985	0,1005	0,1102	0,1189	0,1164	0,1268	0,1312	0,1407
47	0,1124	0,1135	0,1286	0,1322	0,1352	0,1347	0,1433	0,1447
48	0,1190	0,1192	0,1444	0,1493	0,1606	0,1632	0,1785	0,1891
49	0,0947	0,0970	0,0976	0,0991	0,1143	0,1177	0,1235	0,1301
50	0,0863	0,1055	0,1086	0,1353	0,1442	0,1564	0,1689	0,1833
51	0,1064	0,1029	0,1251	0,1189	0,1458	0,1473	0,1602	0,1645
52	0,1518	0,1581	0,1720	0,1813	0,1836	0,1936	0,2068	0,2179
53	0,1414	0,1432	0,1554	0,1544	0,1834	0,1826	0,1998	0,2017
54	0,1378	0,1453	0,1456	0,1635	0,2055	0,2131	0,2281	0,2401
55	0,1158	0,1134	0,1289	0,1245	0,1608	0,1656	0,1817	0,1861
56	0,1547	0,1619	0,1816	0,1974	0,2038	0,2237	0,2320	0,2554
57	0,1911	0,1996	0,2089	0,2215	0,2149	0,2244	0,2310	0,2487
58	0,2397	0,2532	0,2508	0,2760	0,2620	0,2775	0,2712	0,2972
59	0,1865	0,2023	0,1924	0,2078	0,2435	0,2624	0,2580	0,2773
60	0,0736	0,0953	0,0907	0,1060	0,1080	0,1177	0,1220	0,1298
61	0,1040	0,1062	0,1121	0,1166	0,1118	0,1130	0,1157	0,1195
62	0,2062	0,2157	0,2286	0,2441	0,2621	0,2713	0,2681	0,2804
63	0,1562	0,1621	0,1674	0,1822	0,2141	0,2241	0,2211	0,2328
64	0,1145	0,1167	0,1312	0,1297	0,1397	0,1414	0,1458	0,1468

Sigue en la página siguiente.

Usuario	GADW	GAD	GAW	GA	GDW	GD	GW	G
65	0,1553	0,1642	0,1651	0,1755	0,1971	0,1970	0,1972	0,1989
66	0,1209	0,1253	0,1227	0,1365	0,1547	0,1575	0,1569	0,1618
67	0,1225	0,1206	0,1236	0,1212	0,1284	0,1288	0,1284	0,1336
68	0,1161	0,1150	0,1170	0,1144	0,1257	0,1260	0,1291	0,1294
69	0,0823	0,0847	0,0891	0,0977	0,1192	0,1274	0,1324	0,1406
70	0,1076	0,1163	0,1322	0,1394	0,1491	0,1569	0,1644	0,1784
71	0,1640	0,1694	0,1743	0,1877	0,1901	0,2189	0,2121	0,2561
72	0,1027	0,1015	0,0984	0,0983	0,0995	0,0975	0,1075	0,1077
73	0,1175	0,1185	0,1314	0,1322	0,1211	0,1163	0,1356	0,1351
74	0,1054	0,1099	0,1183	0,1223	0,1469	0,1545	0,1672	0,1770
75	0,2598	0,2648	0,2708	0,2782	0,3135	0,3152	0,3187	0,3363
76	0,1594	0,1621	0,1645	0,1660	0,1705	0,1684	0,1791	0,1746
77	0,1153	0,1195	0,1306	0,1354	0,1544	0,1535	0,1612	0,1605
78	0,1192	0,1333	0,1280	0,1423	0,1873	0,1874	0,1924	0,1941
79	0,1459	0,1472	0,1514	0,1565	0,1470	0,1501	0,1501	0,1541
80	0,1460	0,1537	0,1572	0,1683	0,1882	0,1961	0,2069	0,2115
81	0,1252	0,1301	0,1278	0,1347	0,1379	0,1490	0,1464	0,1705
82	0,2113	0,2278	0,2185	0,2389	0,2510	0,2734	0,2532	0,2804
83	0,1462	0,1463	0,1485	0,1561	0,1806	0,1939	0,1980	0,2168
84	0,0903	0,0902	0,0975	0,0963	0,0855	0,0861	0,0924	0,0947
85	0,1057	0,1180	0,1072	0,1128	0,1288	0,1355	0,1279	0,1333
86	0,0818	0,0862	0,0834	0,0903	0,0757	0,0807	0,0752	0,0849
87	0,1395	0,1426	0,1476	0,1476	0,1663	0,1807	0,1811	0,1906
88	0,0975	0,0971	0,0944	0,0944	0,1009	0,1006	0,1001	0,0999
89	0,1502	0,1516	0,1629	0,1667	0,1732	0,1752	0,1924	0,1971
90	0,0926	0,0985	0,0934	0,1004	0,1052	0,1033	0,1106	0,1130
91	0,1286	0,1443	0,1573	0,1834	0,2008	0,2166	0,2276	0,2411
92	0,1782	0,1794	0,1935	0,1962	0,2139	0,2176	0,2159	0,2192
93	0,0889	0,0891	0,1000	0,0967	0,1067	0,1027	0,1128	0,1128
94	0,1062	0,1083	0,1216	0,1208	0,1143	0,1122	0,1278	0,1234
95	0,0748	0,0811	0,0823	0,0861	0,0967	0,1000	0,0945	0,1015
96	0,1069	0,1118	0,1097	0,1137	0,1534	0,1531	0,1576	0,1550
97	0,0992	0,1002	0,1030	0,1050	0,1086	0,1130	0,1124	0,1142
98	0,0676	0,0743	0,0788	0,0904	0,1004	0,1050	0,1115	0,1204

Sigue en la página siguiente.

Usuario	GADW	GAD	GAW	GA	GDW	GD	GW	G
99	0,1050	0,1073	0,1348	0,1477	0,1893	0,1984	0,2088	0,2251
100	0,1760	0,1854	0,1881	0,1978	0,2220	0,2205	0,2226	0,2213
MAE	0,1284	0,1332	0,1383	0,1462	0,1581	0,1640	0,1680	0,1766

Tabla A.3: *Impacto de las propiedades semánticas en el algoritmo de recomendación por inferencia semántica*

A.4. Impacto del uso y la retro-alimentación en la estimación de pedicciones.

Valores numéricos de la figura: 5.8.

Num Calif.	MAE	Num Calif.	MAE	Num Calif.	MAE
1	0,6172859	55	0,16117249	109	0,14322104
2	0,6172859	56	0,15978495	110	0,14222026
3	0,50977602	57	0,16024131	111	0,1403595
4	0,43479578	58	0,16093082	112	0,14009777
5	0,40898223	59	0,16138686	113	0,14122704
6	0,37089992	60	0,15924048	114	0,14012064
7	0,34114216	61	0,15530785	115	0,13963965
8	0,31563492	62	0,15391038	116	0,13932559
9	0,30286914	63	0,15449881	117	0,13909917
10	0,29498119	64	0,15267152	118	0,1394765
11	0,29708512	65	0,15332826	119	0,13919891
12	0,27806026	66	0,15272799	120	0,13891117
13	0,27207853	67	0,15559395	121	0,13852603
14	0,26830713	68	0,15283211	122	0,13844593
15	0,26106546	69	0,1537837	123	0,13795113
16	0,25297407	70	0,15250815	124	0,13802158
17	0,24936185	71	0,15422522	125	0,13833588
18	0,24232536	72	0,1524559	126	0,13785967
19	0,23287017	73	0,15362582	127	0,13689502
20	0,22446412	74	0,15240143	128	0,1367193
21	0,22458968	75	0,15320846	129	0,13631562
22	0,21501573	76	0,15109338	130	0,13549349
23	0,21352803	77	0,15101664	131	0,13421784
24	0,20739265	78	0,15058611	132	0,13336529
25	0,20343055	79	0,14707978	133	0,13348163
26	0,19956602	80	0,14691324	134	0,13392195
27	0,1931012	81	0,14689451	135	0,13352041
28	0,18962299	82	0,14773432	136	0,13297814
29	0,19033716	83	0,14664697	137	0,13244918
30	0,18409481	84	0,14784866	138	0,13420752

Sigue en la página siguiente.

Num Calif.	MAE	Num Calif.	MAE	Num Calif.	MAE
31	0,1853564	85	0,14669604	139	0,13384043
32	0,18275224	86	0,14674952	140	0,13484416
33	0,18346976	87	0,14627428	141	0,13386403
34	0,17997579	88	0,14562878	142	0,13408733
35	0,17982412	89	0,14605525	143	0,13377318
36	0,17631415	90	0,14608254	144	0,13408524
37	0,17888516	91	0,14599169	145	0,13216729
38	0,17301145	92	0,14662763	146	0,13258
39	0,17122583	93	0,14682507	147	0,13259986
40	0,17053743	94	0,14709255	148	0,1321016
41	0,16981818	95	0,14713999	149	0,13188577
42	0,16889311	96	0,14787806	150	0,13159607
43	0,17041854	97	0,14916332	151	0,13104063
44	0,1693778	98	0,14956292	152	0,13147262
45	0,16561128	99	0,14720421	153	0,13068309
46	0,16372771	100	0,14646733	154	0,13050298
47	0,16508277	101	0,14663178	155	0,13061951
48	0,16499558	102	0,14733933	156	0,13031603
49	0,16611749	103	0,14657667	157	0,12959335
50	0,16441386	104	0,14605997	158	0,12882025
51	0,16358688	105	0,14670731	159	0,12765636
52	0,1622614	106	0,14577356	160	0,12789082
53	0,16023461	107	0,14583185	161	0,12752281
54	0,15983509	108	0,14435445	162	0,12779825

Tabla A.4: *Reducción del error a mendida que se incrementa el número de calificaciones.*

A.5. Módulo de KNN.

A.5.1. Vecinos cercanos encontrados para aquellos usuarios con al menos un vecino.

Valores númericos de la figura: 5.9.

Porcentaje	# Usuarios con vecinos	Media de vecinos por usuarios.
1 %	5	1,000
2 %	24	1,417
3 %	38	2,763
4 %	56	4,107
5 %	64	5,734
6 %	71	7,803
7 %	78	9,436
8 %	80	11,413
9 %	82	13,305
10 %	85	15,212
11 %	88	16,489
12 %	92	17,913
13 %	92	19,522
14 %	92	20,913
15 %	93	22,387
16 %	93	23,613
17 %	94	24,606
18 %	94	25,713
19 %	94	26,702
20 %	94	27,606
21 %	94	28,330
22 %	94	28,904
23 %	95	29,211
24 %	95	29,737
25 %	95	30,242
26 %	95	30,705
27 %	95	30,979
28 %	95	31,200

Sigue en la página siguiente.

Porcentaje	A	B
29 %	95	31,368
30 %	95	31,537
31 %	95	31,642
32 %	95	31,726
33 %	95	31,832
34 %	95	31,884
35 %	95	32,032
36 %	95	32,095
37 %	95	32,126
38 %	95	32,147
39 %	96	31,844
40 %	96	31,865
41 %	96	31,865
42 %	96	31,875
43 %	96	31,896
44 %	96	31,896
45 %	96	31,906
46 %	96	31,906
47 %	96	31,906
48 %	96	31,917
49 %	96	31,917

Tabla A.5: *Porcentaje de usuarios con al menos un vecino cercano.*

A.6. Inferencia semántica vs KNN.

Valores numéricos de la figura:5.15.

User	I.S.	KNN	User	I.S.	KNN
1	0,11052957	0,08643695	**52**	0,15446474	0,1653389
2	0,08084585	0,07575756	**53**	0,16016789	0,14544486
3	0,12840713	0,08281234	**54**	0,1404924	0,14343943
4	0,26033911	0,21508071	**55**	0,1240422	0,09502127
5	0,08340806	0,04578055	**56**	0,1773464	0,1699947
6	0,1086576	0,13749027	**57**	0,17266638	0,16790642
7	0,19075848	0,16374597	**58**	0,22140101	0,15332435
8	0,16968856	0,15141188	**59**	0,11687525	0,16516767
9	0,10901173	0,11056019	**60**	0,08682206	0,05262067
10	0,08881715	0,10456133	**61**	0,10025499	0,10989722
11	0,10725734	0,07440887	**62**	0,22717498	0,17912318
12	0,09982675	0,10765816	**63**	0,16523307	0,13301475
13	0,1557639	0,11667667	**64**	0,12907046	0,08179279
14	0,14547941	0,16429694	**65**	0,16831679	0,11086439
15	0,13292036	0,15425971	**66**	0,1419643	0,10853614
16	0,08791157	0,06698649	**67**	0,11228248	0,10301521
17	0,12310922	0,11463903	**68**	0,09848581	0,1102026
18	0,0923263	0,12431307	**69**	0,11308946	0,08457536
20	0,21756076	0,1837259	**70**	0,11943618	0,11117578
21	0,09161249	0,11323041	**71**	0,16877644	0,17024031
22	0,16063585	0,18154296	**72**	0,07869931	0,08048103
23	0,11272085	0,14051321	**73**	0,12047037	0,12835474
24	0,19403385	0,1832611	**74**	0,12121	0,10968648
25	0,19171586	0,19548415	**75**	0,15116748	0,19327232
26	0,10439322	0,10519738	**76**	0,16158391	0,12302724
27	0,10867455	0,10000212	**77**	0,1404587	0,09962434
28	0,19701032	0,14995949	**78**	0,13986206	0,11044971
29	0,20494112	0,14191961	**79**	0,13550008	0,08902494
30	0,1184705	0,11796786	**80**	0,17321492	0,1365092
31	0,14451484	0,12114681	**81**	0,11761669	0,15476166
32	0,09636193	0,1263042	**83**	0,11284209	0,10063608

Sigue en la página siguiente.

User	I.S.	KNN	User	I.S.	KNN
33	0,08679795	0,10202931	**84**	0,08044363	0,10006852
35	0,15261612	0,1755405	**85**	0,09395073	0,08828788
36	0,08419728	0,08773363	**86**	0,08403013	0,08801493
37	0,09689994	0,12326351	**87**	0,14746826	0,14956465
38	0,09555137	0,10326779	**88**	0,10337755	0,09905491
39	0,1358391	0,13870269	**89**	0,10102065	0,09573079
40	0,19168179	0,16670449	**90**	0,07801658	0,10699918
41	0,09970667	0,09778737	**91**	0,09436889	0,09314631
42	0,09123414	0,09450998	**92**	0,1484395	0,11880142
43	0,20764481	0,1758316	**93**	0,08957936	0,08684585
44	0,10517901	0,08902081	**94**	0,10179135	0,10787777
45	0,14533547	0,13247107	**95**	0,07371915	0,06306996
46	0,10512197	0,06562127	**96**	0,10692576	0,11983826
47	0,11183625	0,11380447	**97**	0,09645217	0,10507087
49	0,09530229	0,07915491	**98**	0,08536008	0,06207288
50	0,10728472	0,13576981	**99**	0,14833233	0,13993001
51	0,09574544	0,11564136	**100**	0,18538589	0,145647

Tabla A.6: *Comparativa del algoritmos de inferencia semántica vs KNN*

ACRÓNIMOS

API *Application Programming Interface*. 113

DBMS *Data Base Managment System*. 39, 113

DOI *Degree of Interest*. 33–35, 46, 48, 50, 54, 55, 58, 62, 75, 113

HTML *HyperText Markup Language*. 6, 113

IDE *Integrated Development Environment*. 39, 113

IMDB *Internet Movie Data Base*. 43, 46, 67, 75, 77, 78, 113

IRI *Identificador Internacional de Recursos*. 15, 113

KNN *K Nearest Neighbors*. 38, 48, 86, 95, 109, 111–113

MAE *Mean Absolute Error*. 71, 73, 74, 76, 78, 81–83, 85, 87, 88, 90, 94, 95, 113

OMDB *Open Movie Data Base*. 40, 46, 113

OWL *Web Ontology Language*. 11, 13, 15–17, 29, 32, 34, 39, 40, 44–46, 93, 113

RDF *Resources Description Framework*. 10, 12, 14–16, 29, 31, 34, 39, 40, 57, 93, 113

RDFS *Resources Description Framework Schema*. 12, 14–16, 39, 40, 93, 113

RIF *Rule Interchange Format*. 13, 113

SOA *Service Oriented Application*. 34, 113

SR *Sistema de Recomendación*. 4, 18, 19, 29, 38, 94, 113

SRS *Sistema de Recomendación Semántico*. 3, 28, 64, 67, 93, 94, 113

URI *Uniform Resource Identifier*. 12, 54, 57, 113

W3C *World Wide Web Consortium*. 12, 39, 93, 113

XML *Extensive Markup Language*. 12, 15–17, 19, 40, 113

Bibliografía

[1] M. Lopez-Nores, Y. Blanco-Fernandez, J. J. Pazos-Arias, and R. P. Díaz-Redondo, "Property-based collaborative filtering: A new paradigm for semantics-based, health-aware recommender systems," in *Semantic Media Adaptation and Personalization (SMAP), 2010 5th International Workshop on*. IEEE, 2010, pp. 98–103.

[2] Y. Blanco-Fernández, J. J. P. Arias, M. L. Nores, A. Gil-Solla, and M. R. Cabrer, "Avatar: an improved solution for personalized tv based on semantic inference." *IEEE Trans. Consumer Electronics*, vol. 52, no. 1, pp. 223–231, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/tce/tce52.html#Blanco-FernandezANGC06

[3] I. Cantador, A. Bellogín, and P. Castells, "A multilayer ontology-based hybrid recommendation model," *AI Communications*, vol. 21, no. 2, pp. 203–210, 2008.

[4] V. Codina and L. Ceccaroni, "Taking advantage of semantics in recommendation systems." in *CCIA*, ser. Frontiers in Artificial Intelligence and Applications, R. Alquézar, A. Moreno, and J. Aguilar-Martin, Eds., vol. 210. IOS Press, 2010, pp. 163–172. [Online]. Available: http://dblp.uni-trier.de/db/conf/ccia/ccia2010.html#CodinaC10

[5] M. J. F. Pinto and A. S. Mintegui, *Evaluación de sistemas recomendadores de contenidos audiovisuales basados en técnicas inteligentes*. Montevideo - Uruguay: Universidad de Montevideo., Dec. 2012.

[6] L. Feigenbaum, *Evolution Towards web 3.0. The semantic web*, Cambridge Semantics, 2011. [Online]. Available: http://www.slideshare.net/LeeFeigenbaum/evolution-towards-web-30-the-semantic-web

[7] D. Fensel and F. Facca, *Semantic Web Architecture*, 1992. [Online]. Available: http://teaching-wiki.sti2.at/uploads/5/54/02_SW-Architecture.pdf

[8] Y. Blanco-Fernández, J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, J. García-Duque, A. Fernández-Vilas, R. P. Díaz-Redondo, and J. Bermejo-Muñoz, "A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 305–320, 2008.

[9] "Imdb búsqueda de película pocahontas," cited 2014. [Online]. Available: www.imdb.com/title/tt0114148/fullcredits?ref_=tt_ov_wr#writers

[10] J. Ávila, X. Riofrío, K. Palacio, and O. Autores, "Sistemas de recomendación semánticos: Influencia de las relaciones semánticas," Mayo 2014, descargable en: http://goo.gl/mjjJPd.

[11] V. Kashyap, C. Bussler, and M. Moran, *The Semantic Web*. Springer, 2008.

[12] G. Antoniou and F. Van Harmelen, *A semantic web primer*. MIT press, 2004.

[13] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[14] G. Antoniou, V. Christophides, and D. Plexousakis, "The semantic web: Key ideas," *Encyclopedia of Artificial Intelligence, Idea Group*, 2004.

[15] DUNAVTECH, *History and Context*, DUNAVTECH, http://www.semanticweb.rs/Article.aspx?iddoc=32&id=65&lang=2.

[16] M. Kifer, J. de Bruijn, H. Boley, and D. Fensel, "A realistic architecture for the semantic web." in *RuleML*, 2005, pp. 17–29.

[17] R. MacManus, "10 semantic apps to watch," , Nov. 2007.

[18] V. C. Busquet, "Design, development and deployment of an intelligent, personalized recommendation system," Ph.D. dissertation, Universitat Polit'ecnica de Catalunya, 2009.

[19] D. Brickley and R. Guha, *RDF Schema 1.1*, Google and W3C, http://www.w3.org/TR/rdf-schema/.

[20] D. L. McGuinness and F. van Harmelen, "OWL web ontology language overview," *W3C*, 2004. [Online]. Available: http://www.w3.org/TR/owl-features

[21] w3Schools, *Introduction to XML Schema*, w3Schools, http://www.w3schools.com/Schema/schema_intro.asp.

[22] A. Martin and o. undefined, "Older adulthood, education and social change (australia, new zealand)," Ph.D. dissertation, ResearchSpace Auckland, 2006. [Online]. Available: https://researchspace.auckland.ac.nz/handle/2292/88

[23] P. B. Kantor, L. Rokach, F. Ricci, and B. Shapira, *Recommender systems handbook*. Springer, 2011.

[24] P. Foltz and S. Dumais, "Personalized information delivery: An analysis of information filtering methods," *Communications of the ACM*, vol. 35, no. 12, p. 5160, 1992.

[25] L. Ceccaroni and X. Verdaguer, "Tv finder: una aproximación semántica a la televisión interactiva," in *Proceedings of the workshop on Ubiquitous computation and ambient intelligence of the Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2003), Donostia, Spain*, 2003.

[26] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Applications of Data Mining to Electronic Commerce*. Springer, 2001, pp. 115–153.

[27] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[28] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: two sides of the same coin?" *Communications of the ACM*, vol. 35, no. 12, pp. 29–38, 1992.

[29] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 1, pp. 56–69, 2004.

[30] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[31] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.

[32] E. P. J. M. M. del-Castillo (Universidad de Granada); J. A. Delgado-López, *Sistemas de Recomendación Semánticos. Un análisis del estado de la cuestión*, http://www.upf.edu/hipertextnet/numero-6/recomendacion.html#Sistemas-ontologias.

[33] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities," *Scientific American*, vol. 284, no. 5, pp. 1–5, 2001.

[34] C. Ziegler, L. Schmidt-Thieme, and G. Lausen, "Exploiting semantic product descriptions for recommender systems," in *Proc. of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop (SWIR '04)*, Sheffield, UK, 2004.

[35] S. Kim and J. Kwon, "Effective context-aware recommendation on the semantic web," *International Journal of Computer Science and Network Security*, vol. 7, no. 8, pp. 154–159, 2007.

[36] R.-Q. Wang and F.-S. Kong, "Semantic-enhanced personalized recommender system," in *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 7. IEEE, 2007, pp. 4069–4074.

[37] H. K. Farsani and M. Nematbakhsh, "A semantic recommendation procedure for electronic product catalog." *Enformatika*, vol. 16, 2006.

[38] K. Jung, M. Hwang, H. Kong, and P. Kim, "Rdf triple processing methodology for the recommendation system using personal information," in *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on.* IEEE, 2005, pp. 6–pp.

[39] A. Bouza, G. Reif, A. Bernstein, and H. Gall, "Semtree: Ontology-based decision tree algorithm for recommender systems." in *International Semantic Web Conference (Posters & Demos)*, ser. CEUR Workshop Proceedings, C. Bizer and A. Joshi, Eds., vol. 401. CEUR-WS.org, 2008. [Online]. Available: http://dblp.uni-trier.de/db/conf/semweb/iswc2008p.html#BouzaRBG08

[40] N. Mabroukeh, *SemAware: An Ontolog-Based Web Recommendation System*. University of Windsor, 2011.

[41] R. S. Bovino, "Recomendación de contenidos audiovisuales para familias y grupos de amigos, basado en clasificaciones tv-anytime multidimensionales," *Memoria de Trabajos de Difusión Científica y Técnica*, no. 8, pp. 7–22, 2010.

[42] Y. B. Fernández, J. J. Pazos Arias, M. L. Nores, A. G. Solla, and M. R. Cabrer, "Avatar: An improved solution for personalized tv based on semantic inference," *Consumer Electronics, IEEE Transactions on*, vol. 52, no. 1, pp. 223–231, 2006.

[43] A. J. Fundation, *Getting started with Apache Jena*, http://jena.apache.org/getting_started/index.html.

[44] w3Schools, *MySQL Introduction*, w3Schools, http://www.w3schools.com/php/php_mysql_intro.asp.

[45] B. A. Herlocker J., Konstan J., "Movielens dataset," cited 2014. [Online]. Available: http://grouplens.org/datasets/movielens/

[46] J. E. R. Rodríguez, E. A. R. Blanco, and R. O. F. Camacho, "Clasificación de datos usando el método k-nn," *Vínculos*, vol. 4, no. 1, pp. 4–18, 2013.

[47] Z. Zaier, R. Godin, and L. Faucher, "Evaluating recommender systems," in *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on.* IEEE, 2008, pp. 211–217.

[48] W. Wu, L. He, and J. Y. 0001, "Evaluating recommender systems." *IEEE*, pp. 56–61, 2012. [Online]. Available: http://dblp.uni-trier.de/db/conf/icdim/icdim2012.html#WuH012