UNIVERSIDAD DE MURCIA

FACULTAD DE INFORMÁTICA

Tecnologías para la Recomendación Semántica y Filtrado Colaborativo de Contenidos y Servicios

D. Luis Omar Colombo Mendoza

2017

UNIVERSIDAD DE MURCIA
FACULTAD DE INFORMÁTICA

Tecnologías para la Recomendación Semántica y Filtrado Colaborativo de Contenidos y Servicios

D. Luis Omar Colombo Mendoza

2017

UNIVERSIDAD DE MURCIA
FACULTAD DE INFORMÁTICA

TESIS DOCTORAL

Tecnologías para la Recomendación Semántica y Filtrado Colaborativo de Contenidos y Servicios

Luis Omar Colombo Mendoza

Septiembre 2017

Directores:

Rafael Valencia García

Alejandro Rodríguez González

Agradecimientos

A mi familia (mi mamá, mi abuelita, mi madrina, mis hermanos y mi prima Liliana, mi tío Francisco, mi tío José y su esposa Bella, mi Papá, y los sobrinos y primos pequeños que hacían los días más alegres) por el apoyo de toda mi vida.

A mis compañeros de piso y amigos, Pilar y Mario, por haber sido mi segunda familia durante este camino nada fácil.

A mis amigos del laboratorio de la Fac. de Informática con quienes he compartido estos cuatro años (Ginés, María José, Manuel, Maricarmen, Miguel, Francisco, Philip, Astrid y Ángel).

A mis amigos de la preparatoria y la universidad (Cecilia, Raquel, Alberto, Ivonne, Salomé y Aldo) por sus palabras de aliento durante el camino y por su amistad de hace tantos años.

A otros amigos mexicanos de corazón y de nacionalidad por su compañía de este otro lado del océano Atlántico (Marcelino, Asunción, Patricia, Carmen y Severino).

A mis directores, Rafael y Alejandro, por la oportunidad de llevar a cabo esta investigación, y por el apoyo en el ámbito académico.

A CONACYT por el apoyo económico sin el cual nada de esto habría sido posible.

Índice de Contenido

Índice de Tablas

Índice de Figuras

Lista de Acrónimos

ACB	Ancestro Común más Bajo
API	Application Programming Interface
CARS	Context-aware Recommender Systems
CBR	Case-based Reasoning
CDMA	Code Division Multiple Access
CF	Collaborative Filtering
CFRS	Collaborative Filtering Recommender Systems
DL	Description Logics
DLP	Description Logic Programs
DSS	Dynamic Support System
DTV	Digital Television
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IRI	Internationalized Resource Identifier
JAX-RS	Java API for RESTful Web Services
JAX-WS	Java API for XML Web Services
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
KBRS	Knowledge-based Recommender Systems
LCA	Latent Class Analysis
LDA	Latent Dirichlet Allocation
LSA	Latent Semántica Analysis
MAC	Media Access Control
MAE	Mean Absolute Error
MAUT	Multi-attribute Utility Theory
MSE	Mean Squared Error
NDPM	Normalized Distance-based Performance Measure
NFC	Normalization Form C
NMAE	Normalized Mean Absolute Error
NRA	National Restaurant Association
OLAP	On-Line Analytical Processing
OWL	OWL Web Ontology Language
PCA	Principal Component Analysis
PIB	Producto interno bruto
PLSA	Probabilistic Latent Semántica Analysis
PMF	Probability Matrix Factorization
POI	Point of interest
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
ResQue	Recommender systems' Quality of user experience
REST	Representational State Transfer
RFID	Radio Frequency Identification
RIA	Rich Internet Application
RIF	Rule Interchange Format
RIF-BLD	RIF Basic Logic Dialect
RMSE	Root Mean Squared Error
Rulen	Rule Markup Language

SKOS	Simple Knowledge Organization System
SNOMED-CT	SNOMED Clinical Terms
SOA	Services Oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SPIN	SPARQL Inferencing Notation
SQuaRE	Systems and Software Quality Requirements and Evaluation
STONE	Semantic Tree-based Object Navigator and Editor
SVD	Singular Value Decomposition
SWRL	Semántica Web Rule Language
TF-IDF	Term Frequency-Inverse Document Frequency
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
URP	User Rating Profile
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Capítulo 1 . Introducción

Los sistemas de recomendación personalizada (o sistemas de recomendación, para abreviar) son técnicas y herramientas de software cuyo objetivo es la realización de predicciones y la provisión de sugerencias para usuarios acerca de los ítems en un dominio determinado (Ricci, Rokach, & Shapira, 2011). Formalmente, estos pueden considerarse un tipo híbrido de sistemas de recuperación de información y Sistemas de Soporte a las Decisiones (*DSS*, por sus siglas en inglés) cuyo objetivo es proveer recomendaciones personalizadas de información.

Desde la aparición del sistema experimental de filtrado de correo electrónico, Tapestry, en 1992, el cual es considerado el primer sistema de recomendación de la historia, los sistemas de recomendación han demostrado su efectividad para vencer los retos del problema de la sobrecarga de información en la Web. Además, este tipo de técnicas y herramientas de software han demostrado ser particularmente valiosas para usuarios no experimentados en procesos de toma de decisiones, tal y como lo evidencia su aplicación cada vez más común en dominios aparentemente dispares, como comercio electrónico y comercio móvil (Salamó, McCarthy, & Smyth, 2012); (Li, Wu, & Lai, 2013), noticias (Gao, Chen, Wang, Mensah, & Fu, 2014); (Lin, Xie, Guan, Li, & Li, 2014); (Gu, Dong, & Chen, 2016); (Shi, Ifrim, & Hurley, 2016), turismo y ocio (Borràs, Moreno, & Valls, 2014); (Christensen, Schiaffino, & Armentano, 2016); (Jiang, Qian, Mei, & Fu, 2016) y contenido multimedia y entretenimiento (Lai, Liu, & Liu, 2015); (Mao, Lu, Li, & Yi, 2016).

La creciente popularidad de las redes sociales basadas en localización y los sitios Web de opiniones de usuarios a lo largo de distintos dominios está haciendo posible a las empresas que anuncian sus servicios a través de dichos medios obtener un entendimiento cada vez más amplio y exacto acerca de las preferencias de los usuarios; al mismo tiempo, esto está permitiendo a los usuarios tomar decisiones cada vez más y mejor informadas acerca de que servicios cumplen sus requerimientos en circunstancias específicas al considerar las sugerencias de otros usuarios respecto a sus experiencias personales con los servicios. Sin embargo, para ello resulta imprescindible, para las empresas y para los usuarios, el uso de técnicas y herramientas de software que brinden algún tipo de soporte frente al problema de la sobrecarga de información acerca de las preferencias de los usuarios y de los servicios en sí, respectivamente.

Los sistemas de recomendación se pueden clasificar en cinco categorías principales dependiendo de las técnicas empleadas por estos para realizar las predicciones y proveer las sugerencias sobre los ítems (Adomavicius & Tuzhilin, 2005):

- Sistemas de recomendación basada en filtrado colaborativo: originalmente, estos sistemas recomiendan al usuario objetivo ítems preferidos en el pasado por otros usuarios con preferencias similares a las de él.
- Sistemas de recomendación de filtrado basado en contenido: estos sistemas recomiendan al usuario objetivo ítems similares a los ítems preferidos en el pasado por él.
- Sistemas de recomendación demográficos: estos sistemas recomiendan al usuario objetivo ítems preferidos por otros usuarios existentes en el mismo segmento demográfico de este.
- Sistemas de recomendación basada en conocimiento: estos sistemas realizan recomendaciones a partir de alguna forma de especificación de conocimiento acerca del dominio de los ítems.
- Sistemas de recomendación híbridos: estos sistemas se basan en cualquier combinación de las técnicas anteriores.

Además, con el objetivo de proveer recomendaciones más relevantes para el usuario, un tipo de sistema de recomendación con la habilidad de sugerir ítems de posible interés para el usuario en circunstancias determinadas: sistema de recomendación sensible al contexto (Adomavicius & Tuzhilin, 2008), surgió a inicios

de la década de los 2000. Este tipo de sistema de recomendación se basa en la premisa de que es primordial preservar información potencialmente útil sobre el contexto en el que ocurren las solicitudes de recomendación.

Al mismo tiempo, la Web Semántica fue ideada por Tim Berners-Lee, James Hendler y Ora Lassila (Berners-Lee, Hendler, & Lassila, 2001) en un intento por dotar de estructura al contenido de las páginas Web, dando lugar a un entorno tecnológico en el que existen agentes de software que son capaces, no solo de mostrar dicho contenido, sino de procesarlo y "entenderlo" automáticamente para llevar a cabo tareas útiles para los usuarios rápidamente. Actualmente, dicha idea ha dado lugar a un *framework* común para el intercambio y reutilización de datos entre aplicaciones, empresas y comunidades, a un medio para la definición y descripción de vocabularios y de mecanismos de razonamiento de conocimiento, no solo razonamiento basado en vocabularios, sino razonamiento basado en reglas similares a aquellas utilizadas por los sistemas basados en reglas (lógica de primer grado, programación lógica, reglas de producción y reglas reactivas o reglas evento-condición-acción).

Dada la idoneidad de las tecnologías de la Web Semántica para el desarrollo de sistemas de recomendación basada en conocimiento, el uso de tecnologías como los lenguaje de definición y descripción de vocabularios, *OWL* (de inglés *Web Ontology Language*) y *RDFS* (del inglés *Resource Description Framework Schema*), el framework de representación de información en la Web, *RDF* (del inglés *Resource Description Framework*), el lenguaje de consulta para *RDF*, *SPARQL*, y el lenguaje de reglas y restricciones, *SWRL* (del inglés *Semántica Web Rule Language*), en el desarrollo de este tipo de técnicas y herramientas de software se ha convertido recientemente en tendencia en el campo de la investigación en sistemas de recomendación, esto motivado por la necesidad de buscar nuevas formas de hacer frente al problema antes mencionado.

En este sentido, como uno de los bloques de construcción principales de la Web Semántica, las ontologías – vocabularios complejos y formales- son aprovechadas con dos propósitos primordiales: representación de conocimiento y razonamiento. Más allá, la creación de técnicas de recomendación basada en tecnologías de la Web Semántica, esto es, técnicas de recomendación que aprovechan los mecanismos de razonamiento basado en ontologías o en reglas de inferencia ha contribuido en gran medida a la popularidad de los sistemas de recomendación híbridos basados en conocimiento (Martín-Vicente et al., 2014); (L.-C. Chen, Kuo, & Liao, 2015); (Movahedian & Khayyambashi, 2014); (Al-Hassan, Lu, & Lu, 2015), en los cuales, las técnicas de recomendación basada en conocimiento comúnmente se utilizan en conjunto con técnicas de alguno(s) de los tipos restantes de técnicas de recomendación a fin de aprovechar las ventajas de todas las técnicas involucradas, al mismo tiempo que con la incorporación de unas se contrarrestan las desventajas de otras, y viceversa.

Las técnicas de recomendación basada en filtrado colaborativo bien pueden ser el tipo de técnicas más ampliamente utilizado hoy en día a lo largo de los sistemas de recomendación en una variedad de dominios distintos. En detalle, este tipo de técnicas de recomendación se pueden clasificar en dos grandes grupos: (1) técnicas basadas en memoria y (2) técnicas basadas en modelos. Las técnicas basadas en modelos generalmente implican la utilización de una base de datos de *ratings* para, primeramente, aprender un modelo predictivo del comportamiento o las preferencias de los usuarios que permita la generación posterior de predicciones inteligentes sobre ítems desconocidos por ellos. De hecho, el dominio de la información de las técnicas de recomendación basada en filtrado colaborativo usualmente corresponde a una base de datos de *ratings* asignados por un conjunto de usuarios a un conjunto de *ítems*, en donde dichos *ratings* se pueden considerar una forma de preferencias de usuario de bajo nivel.

Uno de los enfoques más populares en la categoría de técnicas de filtrado colaborativo basado en modelos es, junto con los enfoques basados en modelos de reducción de dimensionalidad y modelos de descomposición de matrices, el enfoque basado en modelos estadísticos. Formalmente, estos enfoques provienen de la

intersección del área de investigación de aprendizaje computacional (e inteligencia artificial en general) y la estadística como disciplina (Cacheda, Carneiro, Fernández, & Formoso, 2011). Por otro lado, distintos modelos existentes en la intersección de las áreas de investigación de aprendizaje computacional y minería de datos han sido ampliamente utilizados en la construcción de técnicas de filtrado colaborativo basado en modelos (Amatriain, Jaimes*, Oliver, & Pujol, 2011).

En el contexto de los enfoques basados en modelos estadísticos, los modelos estadísticos de clases latentes, particularmente el modelo *Latent Semántica Analysis* (*LSA*), el cual se basa en una técnica de modelado estadístico que introduce variables de clases latentes en un modelo mixto con el objetivo de descubrir comunidades de usuarios y perfiles de usuarios prototípicos, han demostrado tener mayor precisión y escalabilidad respecto a las técnicas de filtrado colaborativo basado en memoria (Hofmann, 2004). Más allá, en el contexto del descubrimiento de estructuras de tópicos latentes a partir de colecciones de documentos o datos de uso histórico, el modelo probabilístico generativo de tópicos *Latent Dirichlet Allocation* (*LDA*) ha sido considerado tradicionalmente una alternativa razonable al modelo *Probabilistic Latent Semántica Analysis* (*PLSA*), el cual se puede considerar una especialización del modelo *LSA*.

El modelado probabilístico de tópicos es un problema recurrente en las áreas de investigación de minería de texto y recuperación de información llevado recientemente al campo de la investigación en sistemas de recomendación, especialmente al campo de la investigación en sistemas de recomendación basada en filtrado colaborativo, en un intento por producir recomendaciones más diversas y flexibles a partir de matrices de *ratings*.

Partiendo de la hipótesis de que la combinación de tecnologías de la Web Semántica y modelos estadísticos de clases latentes permitirá el diseño y la implementación de técnicas más potentes, tanto de representación y modelado de información contextual, como de recomendación sensible al contexto, en esta investigación se propone un método híbrido, basado en conocimiento y en filtrado colaborativo bajo un enfoque de modelos estadísticos de clases latentes, de recomendación sensible al contexto para el dominio de la restauración en el contexto de las *APIs* de redes sociales basadas en localización y de sitios Web de opiniones de usuarios.

Las propuestas existentes en la aplicación de técnicas y herramientas de recomendación sensible al contexto al dominio de la restauración emplean, bien tecnologías de la Web semántica, comúnmente el lenguaje de definición y descripción de vocabularios, *OWL*, y el lenguaje de reglas, *SWRL*, bien modelos estadísticos de clases latentes, comúnmente el modelo generativo de tópicos, *LDA*, con dos fines distintos: (1) la definición de técnicas híbridas de minería, modelado y representación de información contextual y (2) la definición de técnicas híbridas de recomendación; no obstante, prácticamente no existen propuestas que aprovechen los posibles puntos de convergencia entre dichos tipos de tecnologías y técnicas computacionales.

De hecho, ya no hablando del dominio del turismo, a diferencia del dominio del ocio, el cual representa a un sector económico relacionado con actividades no necesariamente soportadas por turistas, terceras industrias relacionadas con actividades parcialmente soportadas a consecuencia de la afluencia turística, por ejemplo, la industria de la restauración, no cuentan actualmente con el soporte requerido. En este sentido, esta investigación pretende servir como punto de partida para otros académicos y desarrolladores del área de investigación en sistemas de recomendación en el afán de construir un ecosistema abierto de datos y aplicaciones semánticas en el dominio de la restauración homogenizadas e integradas bajo un enfoque de *Linked Data* y, finalmente, contribuir a cerrar dicha brecha en el largo plazo.

1.1. Organización del Documento

Este documento está organizado de la siguiente manera.

El Capítulo 2 presenta, por un lado, los resultados del análisis del estado del arte en dos vertientes principales: (1) sistemas de recomendación, distinguiendo entre sistemas de recomendación basada en filtrado colaborativo, sistemas de recomendación basada en conocimiento, sistemas de recomendación híbridos y sistemas de recomendación sensibles al contexto, y (2) tecnologías y bloques de construcción principales de la llamada pila de la Web Semántica. Esto da pie a la presentación, en dicho capítulo, de un conjunto condensado de objetivos y una hipótesis a partir de la cual se pretende demostrar la tesis de investigación mediante la aplicación de una metodología basada en el método científico.

En el Capítulo 3 se describen en detalle las subtareas correspondientes a la tarea de formalización de la metodología definida, así como las contribuciones resultantes de la realización de dichas tareas, las cuales en conjunto corresponden a un método híbrido, basado en conocimiento y en filtrado colaborativo bajo un enfoque estadístico de clases latentes y un enfoque basado en memoria, de recomendación sensible al contexto de establecimientos de alimentos y bebidas.

El Capítulo 4 presenta en detalle el método de evaluación propuesto en esta tesis doctoral, el cual está destinado a la validación de la propuesta de la misma (tarea de validación de la metodología definida) y, finalmente, a la demostración de la tesis de la investigación. En detalle, el método de evaluación propuesto es un método doble (métricas tradicionales y métricas de calidad de *rankings*) en forma de análisis comparativo bajo un enfoque de recuperación de información.

En el Capítulo 5 se describen las conclusiones generales de esta tesis doctoral; esto está relacionado, evidentemente, con la demostración de la hipótesis definida. Además, en este capítulo se discuten en detalle las limitaciones principales de las contribuciones de esta investigación, así como las posibles líneas de investigación futura a las que dichas limitaciones pueden dar pie considerando el objetivo a largo plazo de la misma.

Finalmente, el Capítulo 6 presenta un resumen en inglés del contenido de los capítulos restantes.

Capítulo 2 . Estado del Arte

2.1. Introducción

Las técnicas de recomendación basada en conocimiento surgieron a finales de la década de 1990 como un enfoque complementario a las técnicas de recomendación primigenias, a saber, las técnicas de filtrado colaborativo y las técnicas basadas en contenido (R. D. Burke, Hammond, & Young, 1997). Esto significa que los sistemas de recomendación basados en dichas técnicas generalmente son sistemas que combinan más de una técnica empleando distintos métodos de hibridación -sistemas de recomendación híbridos. En este sentido, si bien el conocimiento en los sistemas de recomendación puede estar dado en distintas formas: a) funciones de similitud, b) funciones de utilidad, c) problemas de satisfacción de restricciones, d) bases de conocimiento declarativo, entre otras, son la primera categoría y la última categoría las que resultan de especial relevancia para esta investigación (Jannach & Friedrich, 2011). En la primera de ellas se pueden incluir las métricas de similitud semántica basada en ontologías (vocabularios complejos y formales) de la Web Semántica, mientras que en la última es posible incluir las técnicas de recomendación basadas en las capacidades de inferencia habilitadas por las ontologías y los lenguajes de reglas de la Web Semántica.

De hecho, con los primeros resultados en el desarrollo de la Web Semántica, específicamente después del reconocimiento como recomendación del *World Wide Web Consortium* (*W3C*) (año 2004) de la especificación de la primera versión del lenguaje de definición y descripción de vocabularios, *OWL* (Bechhofer et al., 2004), la investigación en sistemas de recomendación basada en conocimiento tomo un nuevo rumbo en la búsqueda de soluciones más eficaces y eficientes al problema de la sobrecarga de información. En la actualidad, la Web Semántica representa un *framework* común para el intercambio y reutilización de datos entre aplicaciones, empresas y comunidades, un medio para la definición y descripción de vocabularios y de mecanismos de razonamiento de conocimiento, y no solo de razonamiento basado en vocabularios, sino también de razonamiento basado en reglas.

Volviendo al tema de la naturaleza híbrida de los sistemas de recomendación basada en conocimiento, uno de los tipos de técnica de recomendación tradicionalmente más explotado por este tipo de sistemas es el de las técnicas de filtrado colaborativo y, particularmente, el de las técnicas de filtrado colaborativo basado en modelos. Las técnicas de filtrado colaborativo basadas en modelos generalmente implican la utilización de una base de datos de *ratings* para, primeramente, aprender un modelo predictivo del comportamiento o las preferencias de los usuarios que permita la generación posterior de predicciones inteligentes sobre ítems desconocidos por ellos. Más allá, las técnicas basadas en modelos estadísticos de clases latentes, específicamente, las técnicas basadas en el modelo probabilístico generativo de tópicos, *LDA*, han demostrado tener mayor precisión y escalabilidad respecto a otros tipos de técnicas de filtrado colaborativo. Además, el modelo *LDA* ha demostrado ser particularmente útil en el contexto del problema del modelado probabilístico de tópicos, un problema recurrente de los campos de investigación en minería de texto y recuperación de información.

De ahí que el uso de este y otros modelos probabilísticos de clases latentes haya sido explorado recientemente por trabajos en el campo de la investigación en sistemas de recomendación sensibles al contexto con el objetivo de definir técnicas de modelado y representación de información contextual capaces de capturar información contextual que comprende atributos observables parcialmente o atributos inobservables. Esto bajo la suposición de que las interacciones de los usuarios involucran un conjunto relativamente pequeño de “estados” contextuales que pueden explicar el comportamiento de los usuarios en distintos puntos a lo largo de las interacciones. La investigación en sensibilidad al contexto representa una de las líneas de investigación fundamentales en el campo de la investigación en sistemas de recomendación y, así también, en esta tesis doctoral.

En este capítulo se presenta un detallado análisis del estado del arte en sistemas de recomendación, haciendo una clara distinción entre sistemas de recomendación basada en filtrado colaborativo, sistemas de recomendación basada en conocimiento y sistemas de recomendación híbridos, y con una clara orientación a los antecedentes históricos y a las técnicas de recomendación empleadas por estos tipos de sistemas; asimismo, se aborda ampliamente el estado del arte de los sistemas de recomendación sensibles al contexto, enfatizando en las técnicas existentes de representación y modelado y los paradigmas de procesamiento de la información contextual. Dicho análisis se complementa con un análisis un tanto más técnico del estado actual de las tecnologías y bloques de construcción principales de la llamada pila de la Web Semántica. Finalmente, en este capítulo se presentan los objetivos y la hipótesis a partir de la cual se pretende demostrar la tesis de la investigación mediante la aplicación rigurosa del método científico; estos tienen sus bases en los resultados de una serie de análisis comparativos entre los trabajos de investigación más representativos en cada una de las áreas antes mencionadas, los cuales se presentan también a lo largo de este capítulo.

2.2. Sistemas de Recomendación Basada en Filtrado Colaborativo

2.2.1. Antecedentes

Las técnicas de recomendación basada en filtrado colaborativo y, en general, las técnicas de recomendación, tienen sus orígenes a inicios de la década de 1990, o al menos así se considera. En concreto, el acuñamiento del término *filtrado colaborativo* se atribuye al trabajo de Goldberg y colaboradores (D. Goldberg, Nichols, Oki, & Terry, 1992), en el que, motivados por el problema de la sobrecarga de información en la Web, proponen un sistema de correo electrónico experimental denominado “Tapestry” (caso de estudio).

En dicho trabajo, el concepto *filtrado colaborativo* simplemente hace referencia al mecanismo mediante el cual las personas colaboran en el proceso de filtrado de un flujo de documentos electrónicos, asociando “reacciones” a los documentos entrantes. En este contexto, una “reacción”, o anotación, en términos más genéricos, puede ser, por ejemplo, la cualidad de “interesante” o “no interesante” que una persona le atribuye a un documento. Es importante mencionar que, este mecanismo de *filtrado colaborativo* se basa en la premisa de que el proceso de filtrado de documentos basado en contenido puede hacerse más eficiente si se involucran personas en el mismo.

De acuerdo con lo mencionado inicialmente, “Tapestry” es considerado, al mismo tiempo, el primer sistema de recomendación de la historia (si bien no fue explícitamente planteado como tal); esto debido a que el enfoque de filtrado de documentos que implementa en realidad sienta las bases de las técnicas de recomendación en cualquiera de sus formas.

Otro hito sobresaliente en la investigación en sistemas de recomendación basada en filtrado colaborativo (*Collaborative Filtering Recommender Systems, CFRS*) lo representa la publicación del trabajo de Resnick y colaboradores (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994), en el que se presenta una arquitectura y sistema de software abierto para filtrado colaborativo de noticias en Internet. Dicho sistema denominado “GroupLens” comprende, por un lado, clientes para la entrada de *ratings* y la presentación de predicciones sobre noticias de posible interés para los usuarios y, por otro lado, servidores destinados a la distribución y provisión de esos *ratings*. En este trabajo, si bien aún no se formaliza el concepto de *recomendación* como proceso o sistema, se elabora más el concepto de “reacción”, para referirse, con el término de *rating*, a la valoración que un usuario (evaluador) otorga a una noticia electrónica una vez que la ha leído y que sirve de base para la predicción de *ratings* sobre noticias desconocidas.

Un aspecto a destacar del sistema “GroupLens” es que, a diferencia del sistema “Tapestry”, realiza las predicciones para un usuario objetivo basándose en la agregación de *ratings* otorgados en el pasado por distintos evaluadores, particularmente a partir del cálculo de las correlaciones entre los *ratings* otorgados por distintos evaluadores y los *ratings* otorgados por el usuario objetivo, y no en base al *rating* de un solo evaluador.

Este nuevo enfoque de filtrado colaborativo fue bautizado posteriormente con el nombre de "filtrado colaborativo basado en memoria" (Breese, Heckerman, & Kadie, 1998). Además, este trabajo es pionero en el uso de las que actualmente se conocen como métricas de similitud basada en *correlación* estadística; en este caso específico se trata de la "correlación de Pearson", que junto a la "correlación de Spearman", es el tipo de correlación más utilizado actualmente en el campo de la recomendación basada en filtrado colaborativo (Su & Khoshgoftaar, 2009); (Ekstrand, Riedl, & Konstan, 2011).

No fue hasta 1997 que se acuñó el término *sistema de recomendación*. En concreto, Resnick y Varian (Resnick & Varian, 1997) lo utilizan para referirse al tipo de sistema de software que asiste y aumenta el proceso social natural por el cual las personas confían en las recomendaciones de otras personas para tomar decisiones cuando no cuentan con experiencia personal suficiente respecto a una serie de alternativas. A diferencia el término sistema de *filtrado colaborativo*, que hasta entonces fue utilizado en la literatura para referirse a los sistemas de software de las características antes mencionadas, este término es más abstracto, por lo que no hace referencia a técnica de recomendación alguna.

Otro trabajo del mismo año (Balabanović & Shoham, 1997), el cual propone una arquitectura y sistema de software multiagente para la recomendación de páginas Web, a partir del uso combinado de técnicas de filtrado basado en contenido (perfiles de usuario) y técnicas de filtrado colaborativo, puede considerarse el primer trabajo en la literatura de sistemas de recomendación en formalizar el término *sistema de recomendación basada en filtrado colaborativo*. En este trabajo, dicho término se utiliza para referirse al tipo de sistema de recomendación encargado de identificar, respecto a un usuario objetivo, usuarios con gustos similares, para recomendar aquellos elementos que han sido del agrado de dichos usuarios. Más destacable aún es el hecho de que este trabajo esboza el concepto de *sistema de recomendación híbrido*, bajo el supuesto de que, combinando distintas técnicas de recomendación es posible aprovechar las ventajas de las distintas técnicas y, al mismo tiempo, evitar sus desventajas. Como se verá más adelante, este concepto es fundamental en el desarrollo de esta tesis.

Un año más tarde, Konstan y colaboradores presentan el trabajo (Konstan et al., 1997), con el cual dan continuidad al desarrollo de "GroupLens" como proyecto a largo plazo. En concreto, en este trabajo se publican los resultados de la investigación presentada como investigación en curso en el trabajo de 1994, la cual comprendía, principalmente, el mejoramiento de la arquitectura en general, así como de los clientes de noticias en particular, para hacer frente a los retos derivados de utilizar las técnicas de filtrado colaborativo con grandes grupos de usuarios y a gran escala en entornos distribuidos. De ahí que este trabajo sea uno de los primeros en abordar uno de los retos que a la fecha aún es objeto de investigación en el campo de la recomendación basada en filtrado colaborativo: la *dispersión* o escasez de *ratings*. Más allá, este trabajo ya esboza las distintas formas del problema en el que la *dispersión* de *ratings* puede derivar si no es abordado adecuadamente, es decir, las distintas modalidades del problema de "arranque en frío": la modalidad del "nuevo usuario" y la modalidad del "elemento nuevo".

Para hacer frente al reto de la *dispersión* de *ratings*, Konstan y colaboradores proponen el uso de un algoritmo de agrupamiento, el cual les permite fraccionar el conjunto de noticias en *clusters* de noticias que comúnmente son leídas en conjunto y, por consiguiente, fraccionar el conjunto de *ratings* de tal modo que se consigue mejorar la densidad de los mismos, aunque localmente, esto es, en cada *cluster*. Asimismo, se fracciona el conjunto de *correlaciones* para asegurar que los usuarios pueden ser agrupados en *clusters* de usuarios que han valorado las mismas noticias y, finalmente, generar predicciones más significativas. De acuerdo con esto, "GroupLens" puede considerarse hoy en día el primer sistema de recomendación (aunque no es explícitamente presentado como tal) en plantear el uso de técnicas de aprendizaje computacional en el campo de la recomendación basada en filtrado colaborativo, lo que posteriormente fue formalizado por Breese y colaboradores (Breese et al., 1998),

y a su vez extendido por Sarwar y colaboradores (BM Sarwar, Karypis, Konstan, & Riedl, 2000), bajo el nombre de “filtrado colaborativo basado en modelos”.

Casi una década después, en 2005, Adomavicius & Tuzhilin (Adomavicius & Tuzhilin, 2005) esbozaban una serie de líneas de investigación futura en el campo de la investigación en sistemas de recomendación, argumentando que, si bien a la fecha había tenido lugar un progreso significativo en el área, aún se requería más esfuerzo a fin de mejorar la eficacia de los algoritmos de recomendación existentes, así como su idoneidad para un rango de dominios de aplicación mucho más amplio. En el caso de los sistemas de recomendación basada en filtrado colaborativo, se exponía, por un lado, la necesidad de extender el espectro de técnicas utilizadas por los sistemas basados en modelos, más allá de las técnicas estadísticas y de aprendizaje computacional comúnmente explotadas, para incluir, por ejemplo, técnicas procedentes de disciplinas como las matemáticas y las ciencias de la computación. Por otro lado, se planteaba la necesidad de nuevos algoritmos de recomendación que considerasen *ratings* de múltiples criterios, para su uso en dominios de aplicación en los que, por la propia naturaleza de los elementos, las evaluaciones comprendiesen conjuntos de valoraciones.

Asimismo, en este trabajo se expone la necesidad de incorporar *perfiles de usuario* y *perfiles de elemento* en los algoritmos existentes de recomendación basada en filtrado colaborativo, a partir de la explotación de información disponible en historiales de transacciones y en cualquier otra fuente de información valiosa, en donde destacan las *redes sociales*, más allá de los historiales de *ratings*. Para ello, se hace hincapié en la utilización de técnicas avanzadas para el perfilamiento, en contraposición a las técnicas tradicionales basadas en palabras clave (elementos) y características demográficas (usuarios), a saber, técnicas de minería de datos y aprendizaje computacional como reglas de asociación y patrones secuenciales.

En este sentido, uno de los retos que mayor atención está obteniendo en la actualidad es el desarrollo de nuevas tecnologías que permitan la obtención de las preferencias de los usuarios de manera no intrusiva, lo que está directamente relacionado con el uso de tecnologías ubicuas y móviles para la obtención transparente de un tipo de información conocida como información *contextual*, esto es información acerca del contexto del usuario (Felfernig, Jeran, Ninaus, Reinfrank, & Reiterer, 2013). De hecho, la mayoría de estas líneas de investigación antes mencionadas representan, a la fecha, tendencias en el desarrollo de sistemas de recomendación basada en filtrado colaborativo (Agarwal & Bharadwaj, 2013); (Nilashi, Ibrahim, & Ithnin, 2014); (Jing, Wang, & Yang, 2015); (Krzywicki et al., 2015); (Yin, Cui, Chen, Hu, & Zhang, 2015).

2.2.1.1. Sistemas de Filtrado Colaborativo Basado en Modelos

Vale la pena ahondar en los antecedentes del enfoque de filtrado colaborativo basado en modelos. En este contexto, si bien los primeros avances en ese campo de la investigación en sistemas de recomendación se sustentaron en técnicas de agrupamiento basadas en redes bayesianas (Breese et al., 1998), redes bayesianas en general (Y. Chen & George, 1999) y redes de dependencia, el trabajo de Hofmann del año 2004 (Hofmann, 2004) en algoritmos de filtrado colaborativo y minería de datos de usuario basado en la técnica estadística *PLSA* marcó un punto de inflexión en dicho campo en un esfuerzo por mejorar la exactitud de las predicciones de los sistemas de recomendación. La diferencia respecto a los trabajos previos radica en el hecho de que el trabajo de Hofmann se basa en un modelo de variables latentes que introduce la noción de comunidades de usuarios y grupos de elementos, mientras que las técnicas de redes bayesianas y redes de dependencia construyen una estructura de dependencias directamente a partir de las variables observadas.

Con el objetivo de permitir la integración de *ratings* explícitos, los cuales eran habituales en la mayoría de los sistemas de recomendación de filtrado colaborativo existentes a la fecha, Hofmann (Hofmann, 2004) propuso una extensión a la técnica *PLSA*; en concreto, propuso aumentar dicha técnica con una variable aleatoria *v* para los *ratings* explícitos, de modo que el *rating* a predecir dependía directamente de la variable latente *z* y del elemento *y* (la variante denominada “de comunidad”) o del usuario *u* (la variante denominada “de categorías”).

Dicha extensión permitía también el descubrimiento de patrones y regularidades que describían intereses y desintereses compartidos por los usuarios, así como correlaciones entre *ratings* dados a elementos distintos. Finalmente, cabe mencionar que este trabajo se sustentó en un trabajo previo del mismo autor (Hofmann & Puzicha, 1999), el cual se puede considerar hoy en día, junto con los primeros trabajos en filtrado colaborativo basados en redes bayesianas, pionero en proponer un enfoque estadístico para la tarea de filtrado colaborativo; en detalle, se trataba de un enfoque basado en la técnica *Latent Class Analysis* (*LCA*) y la técnica de *biclustering* o *co-clustering*, ambas técnicas consideradas tipos de modelos de mezclas finitas, esto es, técnicas probabilísticas para modelado de datos en una o más dimensiones.

No obstante, la técnica *PLSA*, o más bien la técnica *Latent Semántica Analysis* (*LSA*), la cual inspiró el desarrollo de la primera, está relacionada con técnicas de descomposición de matrices y reducción de dimensionalidad utilizadas con anterioridad en el campo de la recomendación de filtrado colaborativo basado en modelos, a saber, las técnicas *Singular Value Decomposition* (*SVD*) y *Principal Component Analysis* (*PCA*). Evidentemente, la aplicación de dichas técnicas en el contexto de los sistemas de recomendación de filtrado colaborativo tuvo como objetivo mejorar la escalabilidad de los algoritmos utilizados a la fecha, conservando o incluso mejorando la exactitud de las predicciones realizadas, objetivo que aún al día de hoy es difícil de lograr teniendo en cuenta que se trata de dos requerimientos a primera vista incompatibles. Trabajos como (Billsus & Pazzani, 1998), (BM Sarwar et al., 2000) y (K. Goldberg, Roeder, Gupta, & Perkins, 2001) destacan entre los trabajos pioneros en el uso de las técnicas *SVD* y *PCA* en el contexto de los sistemas de recomendación de filtrado colaborativo basado en modelos.

Concretamente, Billsus & Pazzan (Billsus & Pazzani, 1998) propusieron un *framework* que abordó la tarea de filtrado colaborativo como una tarea de clasificación, integrando el uso de un algoritmo de aprendizaje de un modelo de red neuronal artificial pre-alimentada (para el cálculo de las recomendaciones finales) con el uso de la técnica *SVD* (tanto como medio para la extracción de características para el proceso de aprendizaje, como técnica de descomposición de matrices). En particular, con este *framework* se pretendió, por un lado, descartar información considerada no relevante para la tarea de clasificación y, por otro lado, tomar en cuenta posibles interacciones y dependencias entre las características identificadas, como prerrequisito para el cálculo de las correlaciones entre pares de usuarios (formalmente, entre los *ratings* otorgados por ellos) incluso en los casos en los que no existen elementos calificados en común.

Goldberg y colaboradores (K. Goldberg et al., 2001), por su parte, propusieron un algoritmo de filtrado colaborativo de tiempo de ejecución constante (independiente del número de usuarios) al que denominaron "Eigentaste". En la fase *offline* de dicho algoritmo, se empleaba la técnica *PCA* para reducir un subconjunto denso de la matriz de *ratings* extraídos mediante la explotación de una técnica de obtención de *ratings* explícitos a partir de lo que los autores denominaron "consultas universales". Este tipo de consultas, en contraposición a lo que sucede con las denominadas "consultas elegidas por el usuario", permite la construcción de un sub-conjunto denso de la matriz de *ratings* al implicar la calificación por parte de todos los usuarios (durante una fase de perfilamiento) de los elementos en un único sub-conjunto predefinido de dicha matriz. Una vez reducido el subconjunto denso de *ratings*, los usuarios eran agrupados en el sub-espacio de baja dimensionalidad. Por último, durante la fase "online" del algoritmo, se utilizaban los "eigenvectores" para proyectar nuevos usuarios en los *clusters* y, finalmente, se empleaba una suerte de tabla de búsqueda para recomendar a ellos los elementos más apropiados.

Con la aparición de la técnica *LDA* en el año 2003 (Blei, Ng, & Jordan, 2003), la investigación en sistemas de recomendación de filtrado colaborativo basado en modelos tomó un nuevo rumbo en el contexto del modelado de tópicos en el campo de la minería de texto. Dicha técnica es considerada una mejora sobre la técnica *PLSA*, y a su vez se puede interpretar como una generalización de esta última bajo condiciones determinadas. Cabe mencionar que, formalmente, *LDA* es un modelo probabilístico generativo de un corpus, en el que los

documentos se representan como mezclas aleatorias de tópicos latentes, cada tópico se caracteriza por una distribución sobre las palabras en los documentos, y existe una distribución a priori de Dirichlet tanto para la distribución documento-tópico como para la distribución tópico-palabra.

Particularmente, en el año 2004, Marlin (Marlin, 2004) propuso un modelo híbrido generativo y de variables latentes para perfiles de usuario basados en *ratings* explícitos en sistemas de recomendación de filtrado colaborativo. Dicho modelo denominado *User Rating Profile* (*URP*) se basaba tanto en el modelo *LDA*, como en un modelo de mezclas multinomial y en el modelo de aspectos propuesto en el trabajo de Hofmann de 1999 (modelo basado en la técnica *LCA*) (Hofmann & Puzicha, 1999). En el modelo *URP*, los perfiles de usuario se representaban como mezclas de "actitudes" en las que las proporciones se distribuían de acuerdo a una variable aleatoria de Dirichlet. Cada elemento estaba asociado a una actitud y cada actitud a un *rating* de acuerdo a un patrón de preferencias; de modo que los perfiles podían descomponerse en patrones de preferencias y en los grados en los que los perfiles se ajustaban a dichos patrones. En este punto vale la pena mencionar que, ya en el trabajo en el que se proponía el modelo *LDA* (Blei et al., 2003) se mencionaba el filtrado colaborativo como una de las posibles aplicaciones del mismo. En dicha aplicación, se utilizaba, igualmente, para modelar los perfiles de usuario: cada usuario se interpretaba como un documento y cada elemento como una palabra. No obstante, como se verá más adelante en este documento, el enfoque de filtrado colaborativo basado en la técnica *LDA* propuesto en esta tesis doctoral es distinto al planteado en los trabajos antes descritos: se ha propuesto utilizar dicho modelo para extraer la información contextual histórica asociada a los usuarios, a fin de descubrir posibles tópicos implícitos en dicha información, mismos que pueden ser relevantes para los usuarios en la recomendación de elementos (en este caso establecimientos de alimentos y bebidas) desconocidos para ellos en determinados contextos socio-temporales.

En este contexto, muchos de los trabajos más recientes en el campo de la investigación en sistemas de recomendación de filtrado colaborativo basado en modelos proponen la aplicación de técnicas estadísticas, en su mayoría las técnicas *LDA* y *PLSA*, como herramientas para la minería y modelado de información contextual (K. Yu, Zhang, Zhu, Cao, & Tian, 2012); (Xu, Chen, & Chen, 2015); (Allahyari & Kochut, 2016). La Tabla 2.1 resume estos trabajos, así como algunos otros en los que se propone el uso de técnicas estadísticas con un enfoque diferente, a saber, como fundamento de técnicas de recomendación (técnicas para la predicción de *ratings* para elementos desconocidos por los usuarios) más efectivas.

Trabajo	Dominio	Técnica Estadística	Información Contextual		
			Geográfica	Temporal	Social
(K. Yu et al., 2012)	Servicios móviles	LDA (MC y TR)	✓	✓	×
(Xu et al., 2015)	Turismo (puntos de interés)	PLSA (MC)	✓	✓	×
(Pyo, Kim, & kim, 2015)	Contenido multimedia (programas de televisión)	LDA (TR)	No aplica	No aplica	No aplica
(Liu & Xiong, 2013)	Turismo, ocio y restauración (puntos de interés)	LDA (MC y TR)	✓	×	×
(Allahyari & Kochut, 2016)	Contenido multimedia (películas)	LDA (MC)	Indefinido	Indefinido	×
(Zhao, Zhu, Jin, & Yang, 2016)	Micro-blogueo (*hashtags*)	LDA (TR)	No aplica	No aplica	No aplica

Tabla 2.1. Trabajos relacionados en el campo de la investigación en sistemas de recomendación de filtrado colaborativo basado en modelos (MC=Modelado contextual; TR=Técnica de recomendación).

2.2.2. Técnicas de Recomendación Basada en Filtrado Colaborativo

Como se dejó entrever en las subsecciones anteriores de esta sección, las técnicas de filtrado colaborativo se pueden clasificar en dos grandes grupos: 1) técnicas basadas en memoria y 2) técnicas basadas en modelos. En general, las técnicas basadas en memoria implican la utilización de una muestra o incluso de la base de datos de *ratings* completa en la generación de predicciones para ítems desconocidos por los usuarios; mientras que las técnicas basadas en modelos implican utilizar la base de datos de *ratings* para, primeramente, aprender un modelo que permita la generación posterior de predicciones. En este punto, es importante mencionar que, el dominio de la información de las técnicas de recomendación basada en filtrado colaborativo usualmente corresponde a una base de datos de *ratings* asignados por un conjunto de usuarios a un conjunto de *ítems*; dichos *ratings* pueden considerarse una forma de preferencias de usuario de bajo nivel.

2.2.2.1. Técnicas de Filtrado Colaborativo Basado en Memoria

Entre las técnicas de filtrado colaborativo basado en memoria, el tipo primigenio de técnicas de filtrado colaborativo, se pueden distinguir a su vez dos enfoques predominantes: 1) filtrado colaborativo basado en vecindario y 2) filtrado colaborativo de top-n recomendaciones. El primer enfoque se basa en el principio conocido como "boca a boca", según el cual las personas somos susceptibles de ser influenciadas por las opiniones de otras personas con pensamientos similares a los nuestros al evaluar el valor de un ítem respecto a nuestros propios intereses. De ahí que el enfoque original de filtrado colaborativo basado en memoria sea conocido como vecindario basado en usuarios. En este contexto, cabe mencionar que las técnicas de filtrado colaborativo pueden clasificarse, además, en dos categorías distintas dependiendo de si la similitud se calcula respecto a los usuarios o respecto a los elementos o, en otras palabras, dependiendo de si se construyen grupos de usuarios o ítems similares 1) filtrado colaborativo basado en usuarios y 2) filtrado colaborativo basado en ítems (Aggarwal, 2016).

En general, los enfoques de filtrado colaborativo basado en vecindario comprenden las siguientes fases: 1) calcular la similitud (en ocasiones también llamada peso) entre el usuario objetivo (en ocasiones también llamado usuario activo) y el resto de usuarios existentes o entre todos los ítems en la matriz de *ratings*, 2) calcular la predicción para un ítem desconocido por el usuario objetivo como el promedio ponderado de los *ratings* dados al correspondiente ítem por los top-k usuarios más parecidos al usuario objetivo o como el

promedio ponderado de los *ratings* dados por él a los top-k ítems más parecidos al correspondiente ítem (Su & Khoshgoftaar, 2009).

Aquí cabe aclarar que la similitud entre usuarios o ítems comúnmente representa ángulos entre vectores, distancias entre elementos de conjuntos o coeficientes de correlación entre variables. De hecho, las métricas de similitud comúnmente empleadas en los enfoques de filtrado colaborativo basado en memoria corresponden a dos grandes grupos: 1) métricas basadas en ángulos o distancias y 2) métricas basadas en correlaciones. En el primer grupo se encuentran las populares métricas: a) similitud basada en coseno, b) distancia euclídea y c) distancia Manhattan (métrica "taxicab"). En el segundo grupo se encuentran, entre otras, las métricas: a) correlación de Pearson, b) correlación de Spearman y c) *Mean Squared Difference* (MSD), cuyo uso está ampliamente extendido (Su & Khoshgoftaar, 2009); (Desrosiers & Karypis, 2011). Asimismo, distintas variaciones e incluso combinaciones entre dichas métricas de similitud han sido propuestas en la literatura de sistemas de recomendación basada en filtrado colaborativo (enfoque de memoria) con el objetivo de mejorar la exactitud de las recomendaciones. En este documento no se profundizará en estas métricas debido a que, como se verá más adelante, su uso está fuera del alcance de esta investigación.

En lo que respecta al cálculo de la predicción para un ítem dado respecto al usuario objetivo, el método más extendido en la actualidad es el mencionado con anterioridad: tomar la media aritmética ponderada de un subconjunto (correspondiente a los top-k usuarios más similares al usuario objetivo) de todos los *ratings* asignados a dicho ítem; esto en el caso del enfoque de filtrado colaborativo de vecindario basado en usuarios (ver Fórmula 2.1). Este método fue propuesto como parte de la arquitectura y sistema de software abierto para filtrado colaborativo de noticias en Internet, "GroupLens", presentada por Resnick y colaboradores en el año 1994 (Resnick et al., 1994); y aunque originalmente se planteó como un método de ponderación mediante correlación, su planteamiento fue posteriormente extendido de acuerdo a la definición más amplia de similitud que comprende, además de coeficientes de correlación entre variables, ángulos entre vectores y distancias entre elementos de conjuntos.

$$P_{a,i} = \overline{r}_a + \frac{\sum_{u \in U}\left(r_{u,i} - \overline{r}_u\right) . w_{a,u}}{\sum_{u \in U}\left|w_{a,u}\right|} \quad (2.1)$$

Donde:

- *i* es el ítem para el cual se va a predecir el *rating*.
- *a* y *u* son el usuario objetivo y un usuario del conjunto de top-k usuarios más similares a él, *U*, respectivamente.
- $\overline{r}_a$ y $\overline{r}_u$ son los *ratings* promedio del usuario *a* y el usuario *u*, respectivamente.
- $w_{a,u}$ es el grado de similitud entre el usuario *a* y el usuario *u*.

En el caso particular del enfoque de filtrado colaborativo de vecindario basado en ítems, el cálculo de la predicción para un ítem dado respecto al usuario objetivo consiste, típicamente, en tomar la media aritmética ponderada de un subconjunto (correspondiente a los top-k ítems más similares al ítem dado) de todos los *ratings* asignados por dicho usuario a otros ítems (ver Fórmula 2.2). Este método fue originalmente propuesto por Sarwar y colaboradores en el año 2001 en (Badrul Sarwar, Karypis, Konstan, & Riedl, 2001) bajo el nombre de "suma ponderada". De hecho, en dicho trabajo se formalizó el enfoque de filtrado colaborativo basado en ítems, en un esfuerzo por proporcionar nuevas técnicas de recomendación capaces de proveer resultados más exactos en problemas de gran escala, siendo estas dos de las limitaciones del enfoque de filtrado colaborativo basado en usuarios identificadas a la fecha. Evidentemente, el método antes descrito, a diferencia del método de predicción propuesto como parte del enfoque primigenio de filtrado colaborativo basado en usuarios,

contemplaba desde sus orígenes distintas representaciones de similitud entre ítems, a saber, similitud basada en coseno, similitud basada en coseno ajustado y similitud de correlación de Pearson.

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|} \quad (2.2)$$

Donde:

- *i* es el ítem para el cual se va a predecir el *rating*.
- *u* es el usuario objetivo.
- *n* es un ítem del conjunto del de top-k ítems más similares al ítem *i*.
- $w_{i,n}$ es el grado de similitud entre el ítem *i* y el ítem *n*.

La Figura 2.1 muestra una representación gráfica de la fase de predicción de *ratings* en el enfoque de filtrado colaborativo de vecindario basado en ítems. En dicha figura, originalmente presentada en (Badrul Sarwar et al., 2001), no se incluyen los detalles de un método concreto de predicción, ya que se utiliza una "vista de caja negra"; no obstante, se representa al método de "suma ponderada" como el escenario principal, y se indica la posibilidad de utilizar cualquier otro método distinto.

En lo que respecta a los enfoques de filtrado colaborativo de top-n recomendaciones, una vez que las similitudes, entre el usuario objetivo y el resto de usuarios en la matriz de *ratings*, o entre todos los ítems en dicha matriz, se han calculado tal y como en el caso de los enfoques de filtrado colaborativo basado en vecindario, la siguiente fase consiste en la generación de una lista con las top-n recomendaciones como se explicará en detalle a continuación. De hecho, a diferencia de los enfoques basados en vecindario, la tarea de recomendación propiamente dicha no se da en este caso en los términos de la predicción de un *rating* para un ítem desconocido por el usuario objetivo a partir sus preferencias o de las preferencias de los top-k usuarios más similares a él sino en los términos de la generación de una lista con los n ítems con mayor probabilidad de ser de interés para el usuario objetivo según distintas heurísticas.

En el caso del enfoque de filtrado colaborativo de top-n recomendaciones basado en usuarios, la heurística consiste en asumir que los ítems más frecuentemente valorados por el grupo de los top-k usuarios más similares al usuario objetivo y desconocidos para él serán de su interés. En detalle, una vez que las similitudes entre el usuario objetivo y el resto de usuarios en la matriz de *ratings* se han calculado, es necesario agregar las filas de dicha matriz correspondientes a los top-k usuarios más similares al usuario objetivo, a fin de identificar el conjunto de ítems valorados por dicho grupo de usuarios con sus correspondientes frecuencias. Así, finamente los n ítems más frecuentemente valorados por dicho grupo de usuarios y desconocidos por el usuario objetivo son presentados a él como una lista de top-n recomendaciones. Si bien esta manera de llevar a cabo la tarea de recomendación fue originalmente propuesta en el contexto del enfoque de filtrado colaborativo de vecindario basado en usuarios, como parte de la investigación del grupo "GroupLens" a finales del año 2000 (BM Sarwar et al., 2000) (Badrul Sarwar, Karypis, Konstan, & Riedl, 2000) puede considerarse como una extensión o variante de dicho enfoque.

Por otro lado, la heurística en el caso del enfoque de filtrado colaborativo de top-n recomendaciones basado en ítems consiste en asumir que los top-k ítems en la matriz de *ratings* más similares al conjunto como un todo de los ítems valorados en el pasado por el usuario objetivo (excluyendo los ítems ya contenidos en dicho conjunto) serán de interés para él. En detalle, una vez que las similitudes entre todos los ítems en la matriz de *ratings* se han calculado, es necesario construir un subconjunto, *c*, a partir de la unión de los top-k ítems más similares a cada elemento en el conjunto de los ítems valorados por el usuario objetivo en el pasado, *u*; remover del conjunto *c* cualquier ítem que ya forme parte del conjunto *u*; por cada ítem en el conjunto *c*, calcular la similitud respecto al conjunto *u* como la suma de las similitudes entre dicho ítem y cada ítem en el conjunto *u*. Así, los

ítems en el conjunto *c* ordenados según el grado de similitud resultante son presentados al usuario objetivo como una lista de top-n recomendaciones.

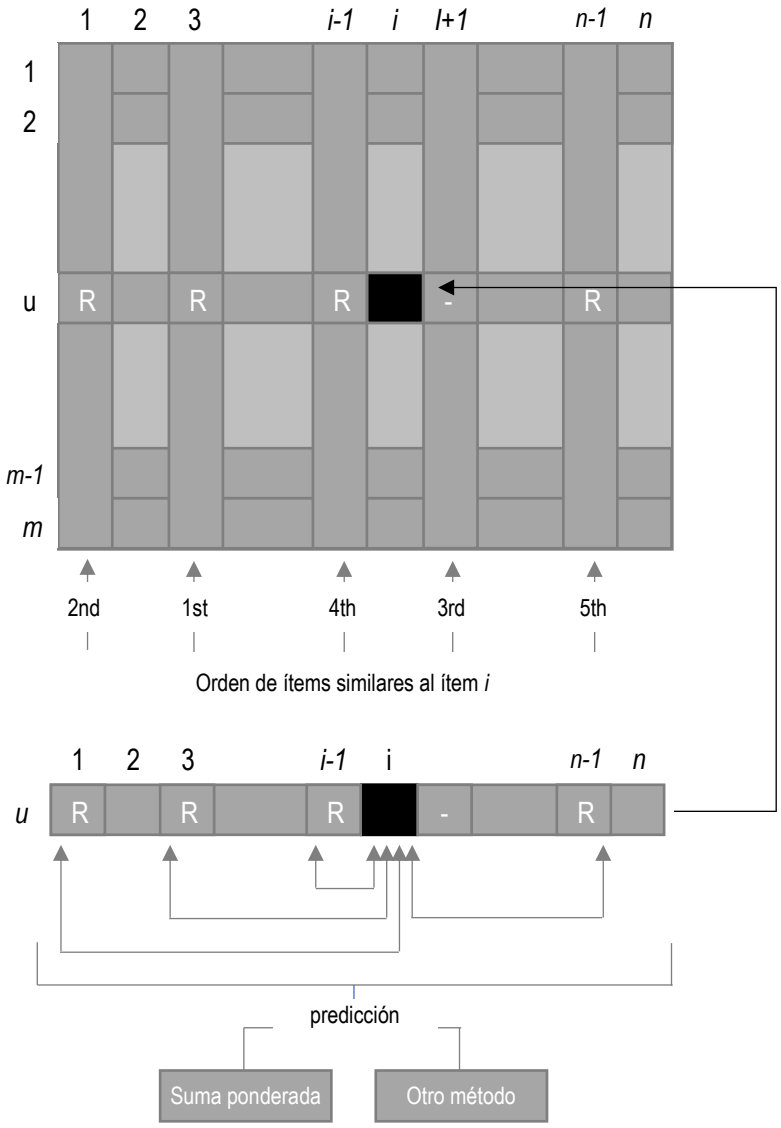


Figura 2.1. Métodos de predicción en el enfoque de filtrado colaborativo de vecindario basado en ítems.

Este método de generación de la lista de top-n recomendaciones fue propuesto originalmente por Karypis en (Karypis, 2001) como parte de la formalización del enfoque de filtrado colaborativo de top-n recomendaciones basado en ítems. Dicha formalización tuvo como motivación abordar dos de las mayores limitaciones del enfoque de top-n recomendaciones basado en usuarios y, en general, de las técnicas de filtrado colaborativo basado en usuarios, según dicho autor: la escalabilidad y el rendimiento en tiempo real. En este punto es importante mencionar que, una de las ventajas principales de las técnicas de filtrado colaborativo basado en ítems respecto a las técnicas de filtrado colaborativo basado en usuarios es que las similitudes se pueden calcular en modo *offline*, esto es, durante una fase previa a la predicción de *ratings* o a la generación de las listas de top-n recomendaciones.

2.2.2.2. Técnicas de Filtrado Colaborativo Basado en Modelos

Las técnicas de filtrado colaborativo basado en modelos surgieron como una alternativa a las técnicas basadas en memoria; surgieron, más concretamente, en un intento por solucionar los problemas de las técnicas basadas en memoria, a saber, las limitaciones de escalabilidad y rendimiento en tiempo real, así como la alta sensibilidad a los problemas de escasez de *ratings* y arranque en frío, los cuales, debido a la naturaleza propia del mecanismo de filtrado colaborativo, afectan en mayor o menor medida, a cualquier técnica basada en éste. En este sentido, la habilidad de las técnicas de filtrado colaborativo basado en modelos para obtener características subyacentes de los datos y, consecuentemente, extraer más información potencialmente relevante ha permitido a los investigadores en el campo de los sistemas de recomendación basada en filtrado colaborativo reducir los problemas antes mencionados. De hecho, si bien se ha demostrado que las técnicas basadas en modelos pueden ser más rápidas que sus predecesoras en la predicción de *ratings*, el costo extra está representado por el tiempo requerido para la construcción de los modelos (rendimiento *offline*) (Cacheda et al., 2011).

Como se mencionó previamente en la subsección 2.2.1 de la presente sección, formalizado por Breese y colaboradores en el año 1998 (Breese et al., 1998), el concepto de "filtrado colaborativo basado en modelos" hace referencia a la clase de técnicas de filtrado colaborativo que explotan algún tipo de modelo predictivo a fin de representar el comportamiento, los intereses o las preferencias de los usuarios a partir de los datos de una comunidad y, utilizando dicha representación, realizan predicciones inteligentes en el proceso de recomendación basada en filtrado colaborativo. A lo largo de la historia, distintos modelos existentes en la intersección de las áreas de investigación de aprendizaje computacional (e inteligencia artificial en general) y, por un lado, el área de investigación de minería de datos y, por otro lado, la estadística como disciplina, han sido ampliamente utilizados en la construcción de técnicas de filtrado colaborativo basado en modelos. En el primer caso es posible mencionar: a) reglas de asociación, b) patrones secuenciales, c) modelos de agrupamiento y d) redes neuronales artificiales; mientras que en el segundo caso es posible mencionar: a) modelos de reducción de dimensionalidad (por ejemplo, *PCA*), b) modelos de descomposición de matrices (por ejemplo, *SVD*), c) modelos lineales (por ejemplo, modelos de regresión lineal) y d) modelos estadísticos como modelos de redes Bayesianas y modelos estadísticos de clases latentes (Cacheda et al., 2011); (Amatriain et al., 2011).

Obsérvese que, debido a la amplitud del espectro de modelos computacionales de posible interés en la construcción de técnicas de filtrado colaborativo, los *ratings* a predecir pueden pasar, de ser los valores de una variable numérica (en la misma escala de las valoraciones dadas por los usuarios a los ítems), a ser los valores de una variable binaria, de una variable categórica, e incluso de una variable nominal. Asimismo, a diferencia de las técnicas utilizadas en los enfoques de filtrado colaborativo de vecindario y de filtrado colaborativo de top-n recomendaciones basado en usuarios, las técnicas de filtrado colaborativo basado en modelos comprenden una etapa previa a la predicción destinada a la construcción de los modelos necesarios (la estimación de los parámetros de los modelos a partir de los datos en la matriz de *ratings*), de manera similar a como las similitudes en algunos enfoques de filtrado colaborativo basado en ítems pueden ser pre-calculadas.

Es la categoría de modelos estadísticos, especialmente los modelos estadísticos de clases latentes, los que atañen especialmente a esta tesis doctoral, siendo abordados en mayor detalle a continuación.

De acuerdo con Breese y colaboradores (Breese et al., 1998), en el contexto de las técnicas de filtrado colaborativo basado en redes bayesianas, la tarea de predicción de un *rating* para un ítem desconocido por el usuario objetivo se puede interpretar, desde un punto de vista probabilístico, como el cálculo del valor esperado del *rating*, dados los datos observados acerca del usuario. Asumiendo que los *ratings* son valores numéricos en el rango [0, m] propone la siguiente fórmula para representar dicha tarea de predicción (ver Fórmula 2.3).

$$p_{a,j} = E(r_{a,j}) = \sum_{i=0}^{m} Pr(r_{a,j} = i | r_{a,k}, k \in I_a) i \quad (2.3)$$

Donde:

- $r_{a,j}$ es el *rating* a predecir para el ítem *j* respecto al usuario objetivo *a*.
- *m* es el extremo superior del rango de posibles valores para los *ratings* e *i* es cada uno de esos valores.
- $r_{a,k}, k \in I_a$ *k* es cada uno de los *ratings* previamente observados para el usuario *a*.

En lo que respecta particularmente a las técnicas de filtrado colaborativo basado en redes bayesianas es importante mencionar que una red bayesiana es un modelo gráfico que utiliza un grafo acíclico dirigido (*DAG*, por sus siglas en inglés) para representar un conjunto de variables aleatorias y sus dependencias condicionales. En dicho modelo, cada nodo $n \in N$ representa a una variable aleatoria, cada arco dirigido $a \in A$ entre nodos es una asociación probabilística entre variables, y además existe una tabla θ de probabilidad condicional que cuantifica el grado de dependencia de cada nodo respecto a sus padres.

Comúnmente, las técnicas de filtrado colaborativo basado en redes bayesianas implementan un enfoque de clasificación para la tarea de predicción; en dicho enfoque, asumiendo que las características son independientes dada la clase, la probabilidad de una cierta clase dada la totalidad de las características puede calcularse y, finalmente, la clase con la más alta probabilidad puede considerarse como la clase predicha. La suposición antes mencionada hace referencia a la propiedad de independencia condicional asumida en la familia de algoritmos de clasificación probabilística "Naive Bayes". De hecho, este algoritmo de clasificación se ha utilizado tradicionalmente para la construcción de técnicas simples de filtrado colaborativo basado en redes bayesianas (Miyahara & Pazzani, 2000); (Su & Khoshgoftaar, 2006); (de Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010).

Según Hofmann (Hofmann, 2004), las técnicas de filtrado colaborativo basado en modelos estadísticos de clases latentes, por su parte, pueden considerarse una familia separada dentro de las técnicas de filtrado colaborativo basado en modelos. En general, dichas técnicas se basan en técnicas de modelado estadístico que introducen variables de clases latentes en un modelo mixto con el objetivo de descubrir comunidades de usuarios y perfiles de usuarios prototípicos. La principal diferencia entre esta categoría de técnicas de filtrado colaborativo basado en modelos estadísticos y la categoría de técnicas basadas en redes bayesianas es que estas últimas aprenden una estructura de dependencias directamente a partir de las variables observadas, mientras que las primeras se basan en un modelo de causas latentes que introduce la noción de grupos o comunidades de usuarios y grupos de ítems.

Como se puede deducir de lo dicho en la subsección 2.2.1.1 de la presente sección, es posible considerar al modelo *PLSA* como uno de los modelos más populares dentro de la categoría de técnicas de filtrado colaborativo basado en modelos estadísticos de clases latentes. No obstante, en el contexto del descubrimiento de estructuras de tópicos latentes a partir de colecciones de documentos o datos de uso histórico, el modelo *LDA* ha sido considerado históricamente una alternativa razonable al modelo antes mencionado por dos razones. 1) Al no llevar a cabo el proceso generativo a nivel de documentos, es difícil utilizar el modelo *PLSA* para asignar probabilidades a nuevos documentos, esto es, a documentos que no están presentes en un corpus de entrenamiento. 2) Relacionado con el motivo antes mencionado, el número de parámetros a estimar en un modelo *PLSA* crece linealmente con el número de documentos de entrenamiento (Blei et al., 2003).

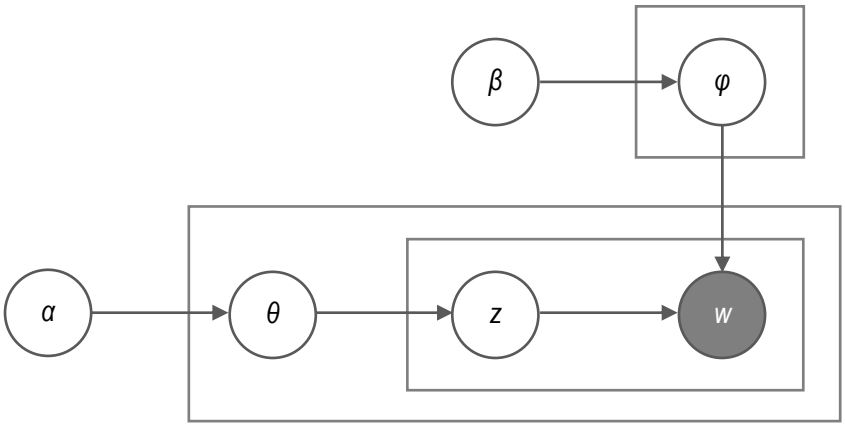
En detalle, el modelado probabilístico de tópicos es un problema recurrente en las áreas de investigación de minería de texto y recuperación de información llevado recientemente al campo de la investigación en sistemas de recomendación, especialmente al campo de la investigación en sistemas de recomendación basada en

filtrado colaborativo, en un intento por producir recomendaciones más diversas y flexibles a partir de las matrices de *ratings*. El modelo *LDA* es un modelo probabilístico generativo para la extracción de información temática (tópicos) de una colección de documentos. Este modelo asume que cada documento se compone de varios tópicos, en donde cada tópico es una distribución de probabilidades sobre palabras, y además introduce una distribución de probabilidad de Dirichlet a priori sobre los pesos de los tópicos por documento.

Sea $D = \{d_1, d_2, \ldots, d_{|D|}\}$ un corpus y $V = \{w_1, w_2, \ldots, w_{|V|}\}$ el vocabulario de ese corpus. Un tópico z_j, $1 \leq j \leq K$ se representa como una distribución de probabilidad multinomial sobre las palabras $|V|$, $p(w_i, z_j), \sum_i^{|V|} p(w_i, z_j) = 1$. El proceso de generación de palabras en el modelo *LDA* comprende dos fases: 1) la generación de las palabras a partir de los tópicos y 2) la generación de los tópicos a partir de los documentos. La distribución de palabras dado un documento se puede calcular, formalmente, utilizando la Fórmula 2.4.

$$p(w_i, d) \sum_{j=1}^{K} p(w_i|z_j)\, p(z_j|d) \quad (2.4)$$

La Figura 2.2 muestra el modelo gráfico del modelo *LDA*, así como las etapas del proceso generativo para un corpus *D*.



1. Para cada tópico $k \in \{1, 2, \ldots, K\}$, muestrear una distribución de palabras $\varphi_k \sim \text{Dir}(\beta)$
2. Para cada documento $d \in \{1, 2, \ldots, D\}$,
   a) Muestrear una distribución de tópicos $\vartheta_d \sim \text{Dir}(\alpha)$
   b) Por cada palabra w_n, donde n {1, 2, …, N} en el documento d,
      i. Muestrear un tópico $z_i \sim \text{Mult}(\vartheta_d)$
      ii. Muestrear una palabra $w_n \sim \text{Mult}(\varphi_{zi})$

Figura 2.2. Representación de Modelo Gráfico del modelo *Latent Dirichlet Allocation*.

A manera de resumen, la Tabla 2.2 presenta las ventajas y desventajas principales de los dos grandes grupos de técnicas de filtrado colaborativo analizadas en esta sección, a saber, filtrado colaborativo basado en memoria y filtrado colaborativo basado en modelos, según los distintos autores citados.

Categoría	Ventajas	Desventajas
Filtrado colaborativo basado en memoria	1. Relativamente fácil implementación. 2. Escalable con ítems coevaluados. 3. Predicciones relativamente exactas (basado en ítems). 4. Similitudes parcialmente estáticas (basado en ítems).	1. Mayor sensibilidad al problema de escasez de *ratings* y al problema de arranque en frio. 2. Escalabilidad limitada con *datasets* extensos. 3. Bajo rendimiento con *datasets* extensos. 4. Limitado grado de personalización (basado en ítems).
Filtrado colaborativo basado en modelos	1. Menor sensibilidad al problema de escasez de *ratings* y al problema de arranque en frio. 2. Relativamente más escalable. 3. Mejor rendimiento aún con *datasets* extensos.	1. Fase *offline* (construcción de modelos) demandante. 2. Perdida de información potencialmente relevante para técnicas de descomposición de matrices y de reducción de dimensionalidad.

Tabla 2.2. Análisis comparativo de los dos grandes grupos de técnicas de filtrado colaborativo existentes en la literatura.

2.3. Sistemas de Recomendación Basada en Conocimiento

2.3.1. Antecedentes

La investigación en sistemas de recomendación basada en conocimiento (*Knowledge-Based Recommender Systems, KBRS*) tiene sus orígenes a finales de la década de 1990, inspirada en la literatura sobre sistemas de razonamiento basado en casos (*Case-Based Reasoning, CBR*) -sistemas que tienen como objetivo resolver nuevos problemas a partir de problemas antiguos (es decir, casos) susceptibles de tener soluciones similares. Debido a que la fase de recuperación es esencial en el proceso *CBR*, los investigadores en este campo se han dedicado tradicionalmente a desarrollar métodos basados en conocimiento para permitir la recuperación precisa y eficiente de casos apropiadamente representados; esto es precisamente lo que dio motivo a la investigación en *KBRS* (R. D. Burke, 2000).

En particular, el trabajo de Burke, Hammond y Young (R. D. Burke et al., 1997) en el que se describe el enfoque para la exploración asistida de espacios de información multidimensionales denominado “FindMe” puede considerarse precursor en la literatura de *KBRS*. Dicho enfoque combina técnicas de búsqueda y exploración de información con técnicas de recuperación de información basada en conocimiento para hacer frente al problema de la sobrecarga de la información en la Web. En este trabajo se introduce, entre otros, el concepto de *métrica de similitud basada en conocimiento*, para referirse a la medición de la proximidad de dos productos (en función de una característica específica) en el cumplimiento de un mismo objetivo; asimismo, se introduce el concepto de *estrategias de recuperación*, esto es, algoritmos destinados a capturar una noción particular de similitud mediante la ejecución ordenada de diferentes métricas.

Es fundamental entender el carácter de orientación a objetivos de una *métrica de similitud* en el contexto del trabajo citado. Sirva de ejemplo el caso de un sistema de recomendación de películas, en el cual, si uno está interesado en obtener películas aptas para menores de edad, específicamente niños, sería conveniente dar mayor prioridad al género cinematográfico que a otras características como el reparto, en el cálculo de la similitud semántica entre las películas. También cabe resaltar que, en dicho contexto, la similitud entre productos no es calculada en base a la totalidad de las propiedades de los productos sino en base a intuiciones particulares de las características que son importantes en los productos, así como de su relevancia, las cuales son capturadas por las *estrategias de recuperación*.

No fue hasta la década del 2000, particularmente la segunda mitad, que la investigación en *KBRS* tomo un nuevo rumbo gracias a los primeros resultados en el desarrollo de la Web Semántica, específicamente después del reconocimiento de la especificación de la primera versión del lenguaje *OWL* como recomendación *W3C* en el año 2004. Las primeras propuestas se centraron en el uso de ontologías *OWL*, como medio para inferir conocimiento implícito no solo a partir de la conceptualización formal del dominio subyacente a un sistema de recomendación -ontología de dominio, sino también a partir de la conceptualización formal de las preferencias de sus usuarios (Middleton, Shadbolt, & De Roure, 2004); (Blanco-Fernandez, Pazos-Arias, Lopez-Nores, Gil-Solla, & Ramos-Cabrer, 2006); (I. Cantador, Castells, & Vallet, 2006).

De las investigaciones en *KBRS* y Web semántica antes citadas cabe resaltar el trabajo de Blanco-Fernandez y colaboradores (Blanco-Fernandez et al., 2006) en el que se propone el uso de una ontología *OWL* para representar el dominio y las preferencias de los usuarios en un sistema de recomendación de contenidos para televisión digital (*Digital Television*, *DTV*) basado en técnicas de filtrado colaborativo y filtrado basado en contenido. En dicho sistema denominado "AVATAR" las capacidades de inferencia de conocimiento implícito (en otras palabras, semántica implícita) son explotadas de tal modo que es posible determinar el nivel de coincidencia semántica entre el contenido y las preferencias de los usuarios.

En este trabajo se introduce el concepto de *métrica de similitud semántica (basada en ontología)* para referirse a la medida de la similitud semántica jerárquica entre dos contenidos de *DTV* en la *ontología de dominio*, a saber, una medida basada en el número de relaciones explicitas *subClassOf* entre las clases de las instancias que representan a los contenidos y la clase raíz en una jerarquía común a esas clases. Asimismo, dicho concepto hace referencia a la medida de la similitud semántica inferida entre dos contenidos de *DTV*, esto es, una medida del número de relaciones implícitas del tipo *objectProperty* existentes entre las instancias que representan a los contenidos: número de caminos en los que ambas instancias están relacionadas a terceras instancias (o a una misma instancia) de una misma clase.

Asimismo, vale la pena destacar el trabajo de Middleton y colaboradores (Middleton et al., 2004), ya que sienta las bases de la técnica de perfilamiento de usuarios conocida hoy en día como *perfilamiento de usuarios basado en ontologías*. Como prueba de concepto, en este trabajo se implementan dos sistemas de búsqueda y recomendación en el dominio de la recomendación de artículos de investigación electrónicos. Dichos sistemas, denominados "Quickstep" y "Foxtrot", implementan un enfoque de recomendación basada en filtrado colaborativo y filtrado basado en contenido en el que los artículos son almacenados en un repositorio centralizado que actúa como reservorio compartido de conocimiento, y clasificados en base a una ontología de dominio, a saber, una ontología para representar jerarquías de tópicos científicos. Las preferencias personales son obtenidas, por un lado, mediante la explotación de los historiales de búsqueda de los usuarios, los cuales son recopilados utilizando *proxy* Web, y, por otro lado, a partir de retroalimentación explícita por parte de ellos; una vez recopiladas, estas preferencias son representadas utilizando un modelo de usuarios basado en una ontología, que permite inferir otras preferencias no observadas explícitamente.

Un aspecto a destacar de la generalidad de las investigaciones pioneras en *KBRS* y Web semántica es que los sistemas de recomendación propuestos son sistemas híbridos que utilizan, además de técnicas de recomendación basadas en conocimiento, específicamente técnicas basadas en conocimiento semántico (tecnologías de la Web semántica), técnicas consideradas más tradicionales, como filtrado colaborativo y filtrado basado en contenido. Esto se debe a que, dada la naturaleza propia de las técnicas de recomendación basadas en conocimiento, estas se plantearon originalmente como un enfoque complementario a las técnicas de recomendación existentes, más que como un enfoque competidor.

Para finalizar, cabe mencionar las tendencias actuales en el desarrollo de sistemas *KBRS* a partir de técnicas de recomendación basadas en tecnologías de la Web semántica. Por un lado, es posible identificar trabajos

que buscan, principalmente, la explotación de las capacidades provistas por la pila completa de tecnologías de la Web semántica. En concreto, se está buscando explotar las capacidades de inferencia basada en reglas, como complemento a las capacidades de inferencia basada en ontologías (Hong, Suh, Kim, & Kim, 2009); (R.-C. Chen, Huang, Bau, & Chen, 2012); (Vesin, Ivanović, Klašnja-Milićević, & Budimac, 2012); (Moreno, Valls, Isern, Marin, & Borràs, 2013), así como las capacidades de consulta y almacenamiento de grafos *Resource Description Framework* (*RDF*), como medio para habilitar la construcción de bases de conocimiento complementarias a las ontologías de dominio (Harispe, Ranwez, Janaqi, & Montmain, 2013); (Ayala, Przyjaciel-Zablocki, Hornung, Schätzle, & Lausen, 2014).

Más allá, la construcción de sistemas de recomendación que aprovechen datos publicados bajo el enfoque de Datos Enlazados (*Linked Data*) está atrayendo la atención de los investigadores de *KBRS* y Web semántica en la actualidad (Di Noia, Mirizzi, Ostuni, Romito, & Zanker, 2012); (Peska & Vojtas, 2013); (Figueroa, Vagliano, Rocha, & Morisio, 2015, p.); (Allahyari & Kochut, 2016). La Tabla 2.3 resume los trabajos relacionados más representativos en el campo de la investigación en sistemas *KBRS*.

Trabajo	**Dominio**	**Tecnología de la Web Semántica**	**Información Contextual**		
			Geográfica	**Temporal**	**Social**
(Hong et al., 2009)	Comercio móvil	Ontologías OWL (MC y TR)	✓	✓	×
(R.-C. Chen et al., 2012)	Salud (Diabetes Mellitus)	Ontologías OWL y lenguaje de reglas SWRL (TR)	No aplica	No aplica	No aplica
(Vesin et al., 2012)	*E-learning*	Ontologías OWL y lenguaje de reglas SWRL (MC y TR)	Indefinido	Indefinido	Indefinido
(Moreno et al., 2013)	Turismo y ocio (actividades)	Ontologías OWL (TR)	✓	✓	✓

Tabla 2.3.Trabajos relacionados en el campo de la investigación en sistemas KBRS (MC=Modelado contextual; TR=Técnica de recomendación).

Como se puede apreciar en la Tabla 2.3, la mayoría de los trabajos analizados incorporan además características de sistemas de recomendación sensibles al contexto (ver sección 2.5 de este capítulo), en donde las tecnologías de la Web semántica se explotan tanto con fines de modelado (e inferencia) de información del contexto del usuario, como para los fines de la fundamentación de técnicas de recomendación más eficientes, específicamente técnicas para el cálculo de similitudes semánticas entre elementos.

2.3.2. Técnicas de Recomendación Basada en Conocimiento: Métricas de Similitud Semántica Baada en Ontologías

En el afán de distinguir las distintas clases o categorías de métricas de similitud semántica basada en ontologías propuestas en la literatura de sistemas *KBRS*, es importante considerar que es posible utilizar distintos criterios de categorización y que, de hecho, no existe una única clasificación universalmente aceptada (ver, por ejemplo (Sánchez, Batet, Isern, & Valls, 2012); (Hadj Taieb, Ben Aouicha, & Ben Hamadou, 2014); (Meymandpour & Davis, 2016)). En este contexto, cabe aclarar que, si bien existen distintos tipos de técnicas de recomendación basada en conocimiento además de las métricas de similitud semántica (específicamente las métricas basadas

en ontologías), son estas últimas las que atañen a esta investigación y, por lo tanto, son estas (además de algunas otras técnicas basadas en los bloques de construcción de la Web Semántica) las únicas que se abordan en este documento.

No obstante, a continuación se da una breve descripción del espectro completo de técnicas de recomendación basada en conocimiento según Jannach y Friedrich (Jannach & Friedrich, 2011). De acuerdo con estos autores, el conocimiento en los sistemas de recomendación puede estar dado por: a) funciones de similitud, b) funciones de utilidad, c) problemas de satisfacción de restricciones y d) bases de conocimiento declarativo. De ahí los distintos tipos de técnicas de recomendación posibles.

Las técnicas basadas en funciones de similitud buscan determinar el grado de similitud entre alguna forma de consulta de usuario y los ítems en el espacio de recomendación. Los sistemas de recomendación que emplean este tipo de técnica basada en conocimiento se pueden ver como una especie de sistema basado en casos (la relación entre los sistemas de recomendación y los sistemas basados en casos se explicó con anterioridad en la subsección 2.3.1. de esta sección). Como es posible deducir, en esta categoría se pueden considerar las métricas de similitud basada en ontologías.

Las técnicas basadas en funciones de utilidad están enfocadas en cuantificar las preferencias del usuario entre un conjunto de alternativas, a saber, los ítems. Ejemplos de funciones de utilidad son las funciones de múltiples atributos basadas en el método de análisis de decisiones conocido como *Multi-Attribute Utility Theory* (*MAUT*), en las cuales las preferencias del usuario para los distintos criterios son agregadas en una función de valor total (Huang, 2011).

Las técnicas basadas en problemas de satisfacción de restricciones, por su parte, consisten en la interpretación de las preferencias del usuario como requerimientos (restricciones), y la evaluación, mediante implicaciones lógicas o funciones objetivo con restricciones de obligación o de preferencia explícitas, de la satisfacción de las mismas respecto a las características de los ítems (variables). Un ejemplo de este tipo de técnica es la técnica implementada por Burke (R. D. Burke, 2000) en el sistema de recomendación de productos y servicios en múltiples dominios denominado “Recommender Personal Shopper” (“recommender.com”), el cual representó la culminación del proyecto de investigación iniciado con la investigación en el enfoque para la exploración asistida de espacios de información multidimensionales considerado precursor en la literatura de sistemas *KBRS*, “FindMe” (ver la subsección anterior).

Por último, las técnicas basadas en bases de conocimiento declarativo consisten en la definición, por parte de expertos, de modelos de dominio, así como de reglas sobre los datos en el mismo. En esta categoría se encuentran las técnicas de recomendación basadas en las capacidades de inferencia habilitadas por las ontologías y los lenguajes de reglas de la Web Semántica. Como se verá en detalle en el siguiente capítulo de este documento, el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas propuesto en esta tesis doctoral corresponde, formalmente, a un método híbrido de recomendación basada en conocimiento y en filtrado colaborativo (tanto basado en memoria como basado en modelos), que comprende una métrica de similitud semántica que a su vez explota las capacidades de inferencia habilitadas por las ontologías.

Ahondando en el tópico de las métricas de similitud semántica basada en ontologías, una vez concluido el preámbulo relativo al espectro de técnicas basadas en conocimiento existentes en la literatura de *KBRS*, cabe aclarar que en esta investigación se ha tomado como referencia la clasificación propuesta por Sánchez y colaboradores en (Sánchez et al., 2012). Según esta clasificación, es posible distinguir entre: 1) enfoques basados en caminos (grafos), 2) enfoques basados en conjuntos de características ontológicas y 3) enfoques basados en el concepto de “contenido de información” procedente de la teoría de la información.

Las métricas basadas en caminos (grafos) asumen que una ontología se puede considerar un grafo dirigido en el que los conceptos están interrelacionados principalmente por relaciones taxonómicas (*is-a*) y, en menor medida, por relaciones no taxonómicas. En detalle, este tipo de métricas permite calcular la similitud entre dos términos mediante el mapeo de los mismos a los conceptos en una ontología y el cálculo de la longitud del camino (secuencia de arcos representados por relaciones taxonómicas) más corto entre dichos conceptos. En este sentido, es importante tener en cuenta que en la literatura existen distintas interpretaciones del concepto de camino mínimo, con las correspondientes implicaciones que esto puede tener.

A diferencia de los dos tipos de métricas restantes, la principal ventaja de las métricas basadas en caminos es su relativa simplicidad, dado que se basan únicamente en conceptos procedentes de la teoría de grafos. Por el contrario, a diferencia de dichos tipos de métricas, el rendimiento de las métricas basadas en caminos puede ser relativamente bajo al emplearse en casos en los que los conceptos están involucrados en jerarquías de "herencia múltiple", debido a que se considera solamente el camino más corto entre los conceptos, omitiendo conocimiento taxonómico potencialmente relevante.

Por su parte, las métricas basadas en conjuntos de características ontológicas surgieron de la necesidad de superar otra limitación importante de las métricas del tipo antes mencionado: la suposición de que las relaciones taxonómicas en las ontologías representan distancias uniformes. En detalle, las métricas basadas en conjuntos de características permiten calcular la similitud semántica entre dos conceptos como el grado de solapamiento entre los conjuntos resultantes de interpretar los conceptos como funciones de sus propiedades. Este enfoque está basado en el modelo de similitud de Tversky que procede de la teoría de conjuntos (Tversky, 1977), y considera tanto las características comunes como las características no comunes de los conceptos.

En este caso, resulta crucial la definición de los conjuntos de características, para lo cual existen diversos enfoques en la literatura de sistemas *KBRS*, los cuales consideran otros elementos de las ontologías aparte de relaciones taxonómicas y relaciones no taxonómicas, como sinónimos, definiciones y otros tipos de relaciones semánticas (sirva de ejemplo la base de datos léxica "Wordnet") Lo anterior a su vez puede representar una desventaja de este tipo de métricas, ya que limita su aplicabilidad a ontologías que incluyen dichos elementos, que suelen ser ontologías muy detalladas.

Por último, las métricas basadas en el concepto de "contenido de información" radican formalmente en el cálculo de las probabilidades de aparición de los conceptos de una taxonomía a partir de las ocurrencias en un corpus de entrada, y tienen como objetivo complementar la estructura taxonómica de las ontologías con la distribución de información de los conceptos en los corpus. No obstante, los enfoques más actuales en este contexto proponen derivar los valores de contenido de información intrínsecamente, esto es directamente de la ontología, para con ello dejar de depender de corpus, bajo la suposición de que la estructura taxonómica de las ontologías sigue el principio de "prominencia cognitiva", el cual sugiere que los humanos tienden a especializar conceptos cuando necesitan diferenciarlos de otros ya existentes, dando lugar a relaciones de hiponimia (generalización). Una de las desventajas principales de este tipo de métricas es que su escalabilidad y aplicabilidad se puede ver comprometida en el afán de obtener valores de probabilidad lo más exactos posibles; dada la fase manual de preprocesamiento del corpus que esto supone. Asimismo, en los casos en los que, o bien la taxonomía o bien el corpus, cambian con el tiempo, el uso de este tipo de métricas puede requerir el recalculo de los valores de probabilidad.

La Tabla 2.4 muestra un breve análisis comparativo de trabajos recientes en el campo de la investigación en sistemas *KBRS* en los que se proponen distintos tipos de métricas de similitud semántica basada en ontologías.

Trabajo	Tipo de sistema de recomendación	Tipo de métrica			
		Caminos (grafos)	**Conjuntos de caract.**	**Contenido de información**	**Otro**
(Z. Yu, Nakamura, Jang, Kajita, & Mase, 2007)	Recomendación basada en conocimiento	✓ (No considera relaciones no taxonómicas)			
(Iván Cantador, Bellogín, & Castells, 2008)	Híbrida (basada en conocimiento y en filtrado colaborativo) *				Métrica basada en técnica de agrupamiento.
(Blanco-Fernández et al., 2008a)	Recomendación basada en conocimiento	✓ (No considera relaciones no taxonómicas)			
(Blanco-Fernández et al., 2008b)	Híbrido (basada en conocimiento y en filtrado colaborativo) *	✓	✓		
(Sánchez et al., 2012)	N/A		✓		
(Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012)	Híbrido (basada en conocimiento y en filtrado colaborativo)		✓ (No considera las caract. no comunes)		
(Meymandpour & Davis, 2016)	Híbrido (basada en conocimiento y en filtrado colaborativo)			✓	

Tabla 2.4. Métricas de similitud semántica basada en ontologías propuestas en la literatura de sistemas KBRS.

Como se explicará detalladamente en el siguiente capítulo de este documento, la métrica de similitud semántica basada en ontologías propuesta como parte de esta investigación es similar en naturaleza a la métrica propuesta en (Blanco-Fernández et al., 2008b), dado que pretende capturar los dos tipos distintos de conocimiento comúnmente disponibles en las ontologías: conocimiento taxonómico y conocimiento no taxonómico explicito e implícito (ver Fórmula 2.5). Dicha métrica comprende dos componentes correspondientes a dos tipos de conocimiento antes mencionados, los cuales, a su vez, corresponden a los enfoques de similitud basada en caminos (grafos) y similitud basada en conjuntos de características, respectivamente.

$$SemSim(a,b) = \alpha . SemSim_{Inf}(a,b) + (1-\alpha) . SemSim_{Hie}(a,b) \quad (2.5)$$

Donde:

- *a* y *b* son dos individuos a comparar.
- *α* es el factor de combinación de los dos componentes de la métrica, donde $\alpha \in [0,1]$, del cual se derivan los pesos relativos de los mismos.

- *SemSim$_{Inf}$* es el componente que permite capturar el conocimiento inferencial; se calcula mediante el uso de la Fórmula 2.6.
- *SemSim$_{IHie}$* es el componente que permite capturar el conocimiento taxonómico; se calcula utilizando la Fórmula 2.7.

$$SemSim_{Hie}(a,b) = \frac{depth\big(LCA(a,b)\big)}{max\big(depth(a), depth(b)\big)} \quad (2.6)$$

El componente de esta métrica correspondiente al conocimiento taxonómico (SemSim$_{Hie}$) se basa en los conceptos de profundidad y Ancestro Común más Bajo (ACB) procedentes de la teoría de grafos, y permite calcular la similitud semántica entre dos individuos de una ontología a partir de la posición de las clases a las que estos pertenecen en la jerarquía de clases, esto es, la similitud semántica taxonómica. En detalle, en la Fórmula 2.6 se tiene que:

- *LCA(a,b)* representa el ACB de los individuos *a* y *b*; se define como la clase más profunda que es ancestro tanto de la clase a la que pertenece *a* como de la clase a la que pertenece *b*.
- *depth(x)* representa la profundidad de una clase o una instancia; se define como la distancia (número de relaciones taxonómicas) entre la clase raíz y la clase correspondiente o la clase a la que la instancia correspondiente pertenece.

$$SemSim_{Inf}(a,b) = \frac{1}{\#CI_{MAX}(a,b)} \sum_{k=1}^{\#CI(a,b)} DOI(i_k) \quad (2.7)$$

Por su parte, el componente de la métrica correspondiente al conocimiento inferido (SemSim$_{Inf}$) se basa en el descubrimiento de relaciones implícitas entre los individuos comparados, de modo que dos individuos *a* y *b* se consideran similares si están asociados mediante propiedades a otros individuos de la misma clase hoja. A dicha asociación se le denomina “clase de unión”; mientras que en el caso de que los individuos comparados se encuentren asociados a un mismo tercer individuo la asociación se denomina “individuo de unión”. Como se puede deducir, estos conceptos son equivalentes, respectivamente, a los conceptos de característica no común y característica común del modelo de similitud de Tversky, mencionado con anterioridad en esta subsección.

En detalle, en la Fórmula 2.7 se tiene que:

- *a* y *b* son los individuos a comparar.
- *#CI(a,b)* es el número de individuos comunes a los individuos *a* y *b* (tanto instancias de unión como instancias de una clase de unión).
- i_k representa a cada uno de los individuos comunes.
- *#CI$_{MAX}$(a,b)* es el número máximo de individuos que los individuos *a* y *b* pueden compartir.
- *DOI(i_k)* es el índice que representa el grado de interés del usuario por cada uno de los individuos comunes.

Como se verá en detalle en el siguiente capítulo de este documento, a fin de capturar tanto conocimiento taxonómico como conocimiento no taxonómico explícito e inferido de manera similar a como lo hace la métrica antes descrita (Blanco-Fernández et al., 2008b), la métrica de similitud semántica basada en ontologías propuesta está fundamentada en la combinación de dos métricas mencionadas en la Tabla 3 y descritas a continuación.

Por un lado, la métrica propuesta por Sánchez y colaboradores en (Sánchez et al., 2012), una métrica basada en el enfoque de conjuntos de características que, a diferencia de la gran mayoría de las métricas en esta

categoría explota las estructuras taxonómicas de las ontologías bajo el argumento de que, en muchas ocasiones, no se dispone de otros tipo de relaciones semánticas. En este sentido, una ventaja de esta métrica respecto a otras que utilizan también el enfoque basado en conjuntos de características es que, al considerar solo relaciones taxonómicas, no requiere la configuración de parámetros de ponderación para distintos tipos de relaciones semánticas.

La métrica citada (ver Fórmula 2.8), está destinada, realmente, al cálculo de la disimilitud o distancia semántica entre pares de clases. En concreto, esta se basa en la hipótesis de que, un concepto se puede distinguir semánticamente de otros mediante la comparación del conjunto de los conceptos que lo subsumen.

$$dis_{norm}(a,b) = \log_2\left(1 + \frac{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)|}{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)| + |\emptyset(A) \cap \emptyset(B)|}\right) \quad (2.8)$$

En donde:

- *a*, *b* son un par de clases procedentes de una jerarquía de clases de una ontología.
- *Ø(a)* representa al conjunto de características taxonómicas que describen a la clase *a* en términos de subsunción de conceptos.
- *Ø(b)* representa al conjunto de características taxonómicas que describen a la clase *b* en términos de subsunción de conceptos.
- *Ø(x)\ Ø(y)* es la cardinalidad del conjunto de características taxonómicas diferenciales de una clase *x* respecto a otra clase *y* (características no comunes).
- *Ø(x) ∩ Ø(y)* es la cardinalidad del conjunto de características taxonómicas comunes a una clase *x* y una clase *y*.

Por otro lado está la métrica propuesta en (Carrer-Neto et al., 2012) como resultado del trabajo previo en sistemas de recomendación basada en conocimiento del grupo de investigación. Esta métrica (ver Fórmula 2.9), a su vez se basa en la métrica presentada por Blanco-Fernández y colaboradores en (Blanco-Fernández et al., 2008b), aunque está enfocada solamente al descubrimiento de conocimiento semántico no taxonómico explícito, por lo que corresponde (parcialmente) solo al componente $SemSim_{Inf}$ de dicha métrica. En detalle, se trata también de una métrica basada parcialmente en el enfoque de conjuntos de características, en concreto, en el concepto de característica común del modelo de similitud de Tversky.

$$Similarity(a,b) = \sum_{i=1}^{\#P}\left(\frac{common(a,b,p)}{max\big(deg(a,p), deg(a,p)\big)}\right) \quad (2.9)$$

Donde:

- *a, b* es la pareja de individuos a comparar.
- *P* es un array de las relaciones no taxonómicas (propiedades de objeto) de la clase a la que pertenecen los individuos *a* y *b* que deben tenerse en cuenta en el cálculo de la similitud.
- *p* representa a cada una de las propiedades contenidas en el array *P*.
- *deg(x,p)* representa el número de individuos asociados a un individuo *x* a través de la propiedad de objeto *p*.
- *common(a,b,p)* es el número de instancias comunes asociadas a los individuos *a* y *b*.

2.4. Sistemas de Recomendación Híbridos

2.4.1. Antecedentes

Como se mencionó anteriormente en este documento, la investigación de Balabanović & Shoham que propuso una arquitectura y sistema de software multiagente para la recomendación de páginas Web (Balabanović & Shoham, 1997) puede considerarse, actualmente, pionera en hibridación de técnicas de recomendación, específicamente técnicas de filtrado colaborativo y filtrado basado en contenido. A dicha investigación le siguió la investigación de 1998 de Basu, Hirsh, & Cohen (Basu, Hirsh, & Cohen, 1998) en la cual se propuso un enfoque de aprendizaje computacional, específicamente de clasificación, para la recomendación de películas a partir tanto de los *ratings* proporcionados por los usuarios como de la información del contenido de las películas. Cabe aclarar que este enfoque consiste en predecir si una película será o no del gusto de un usuario, más que en predecir el *rating* concreto que un usuario otorgará a una película. Esta investigación se basa en la suposición de que las predicciones en las técnicas de filtrado colaborativo, al depender solamente de los *ratings* otorgados en el pasado por otros usuarios y descartar la información que muchas veces está disponible sobre la naturaleza de los elementos a recomendar, son susceptibles al problema de "arranque en frío", específicamente a la modalidad de "elemento nuevo", derivado de la dispersión o escasez de *ratings*.

Por otro lado, aunque no fue descrita explícitamente como tal, en una investigación del mismo año de Sarwar y colaboradores (B. M. Sarwar et al., 1998) se propuso una extensión mediante técnicas de recomendación basada en conocimiento (en su sentido más general) a la arquitectura y sistema de software abierto para filtrado colaborativo de noticias en Internet, "GroupLens". De hecho, dicha extensión se planteaba como un modelo de integración de técnicas de filtrado basado en contenido y filtrado colaborativo destinado a hacer frente al reto de la dispersión o escasez de *ratings* y consistía básicamente en la incorporación a la arquitectura original de agentes semi-inteligentes para el filtrado automático basado en contenido (*filterbots*). Más concretamente, dichos agentes, programados con algoritmos sencillos de emparejamiento de las características del contenido de las noticias con las necesidades generalizadas de los usuarios, permitían la generación automática de *ratings* para noticias recién publicadas y, por tanto, carentes de *ratings* por parte de los usuarios. De ahí que el sistema de recomendación resultante pueda considerarse un sistema híbrido basado más bien en técnicas de filtrado colaborativo y de recomendación basada en conocimiento.

De acuerdo con la investigación de 2002 de Burke (R. Burke, 2002) en la cual se formalizó el concepto de *sistema de recomendación híbrido*, así como los distintos enfoques de hibridación de técnicas de recomendación utilizados por los sistemas híbridos existentes a la fecha, la extensión a la arquitectura y sistema de software "GroupLens" propuesta por Sarwar y colaboradores (B. M. Sarwar et al., 1998) representa de hecho el primer sistema de recomendación híbrido parcialmente basado en conocimiento. En (R. Burke, 2002), además de presentarse un análisis del estado del arte en sistemas de recomendación híbridos, se propuso un sistema híbrido basado en técnicas de filtrado colaborativo y de recomendación basada en conocimiento. Dicho sistema denominado "EntreeC", era realmente una extensión mediante técnicas de filtrado colaborativo al sistema de razonamiento basado en casos para la recomendación de restaurantes, "Entree". "Entree" fue uno de los primeros sistemas de recomendación basada en conocimiento, dado que implementaba el enfoque para la exploración asistida de espacios de información multidimensionales, "FindMe" (ver la sección anterior para más información). En dicho sistema, las acciones de navegación de los usuarios son consideradas *ratings* implícitos. En detalle, el conocimiento sobre los restaurantes, el cual está representado por los casos en la base de casos, es utilizado inicialmente para producir recomendaciones considerando solo las necesidades expresadas por los usuarios; posteriormente, los *ratings* implícitos son empleados para ordenar las recomendaciones por orden de preferencia. Este enfoque de hibridación es formalmente definido como hibridación en cascada.

Mención aparte merece el enfoque de filtrado colaborativo extendido propuesto por Mobasher, Jin, & Zhou en el año 2004 (Mobasher, Jin, & Zhou, 2004), ya que hoy en día puede ser considerado como el primer enfoque híbrido de recomendación basada en conocimiento, específicamente conocimiento semántico (tecnologías de la Web semántica). En dicho enfoque de recomendación, conocimiento semántico estructurado sobre los elementos a recomendar se extrae automáticamente de la Web usando como referencia ontologías de dominio; dicho conocimiento es posteriormente utilizado en combinación con los *ratings* otorgados en el pasado por los usuarios (filtrado colaborativo basado en memoria) para predecir *ratings* para elementos desconocidos por los mismos. En particular, las predicciones son realizadas a partir del cálculo de las similitudes basadas en coseno entre los vectores que representan a los elementos en términos de los valores que toman sus características en las ontologías de dominio (similitud semántica) y los vectores que representan a los mismos elementos en términos de *ratings*, resultando en una métrica de similitud combinada. Es importante mencionar que la técnica de recomendación basada en conocimiento propuesta en este trabajo puede ser considerada al mismo tiempo un tipo de técnica de filtrado basado en contenido dado que radica en una métrica de similitud comúnmente utilizada en dicho campo de la investigación en sistemas de recomendación, la similitud basada en coseno.

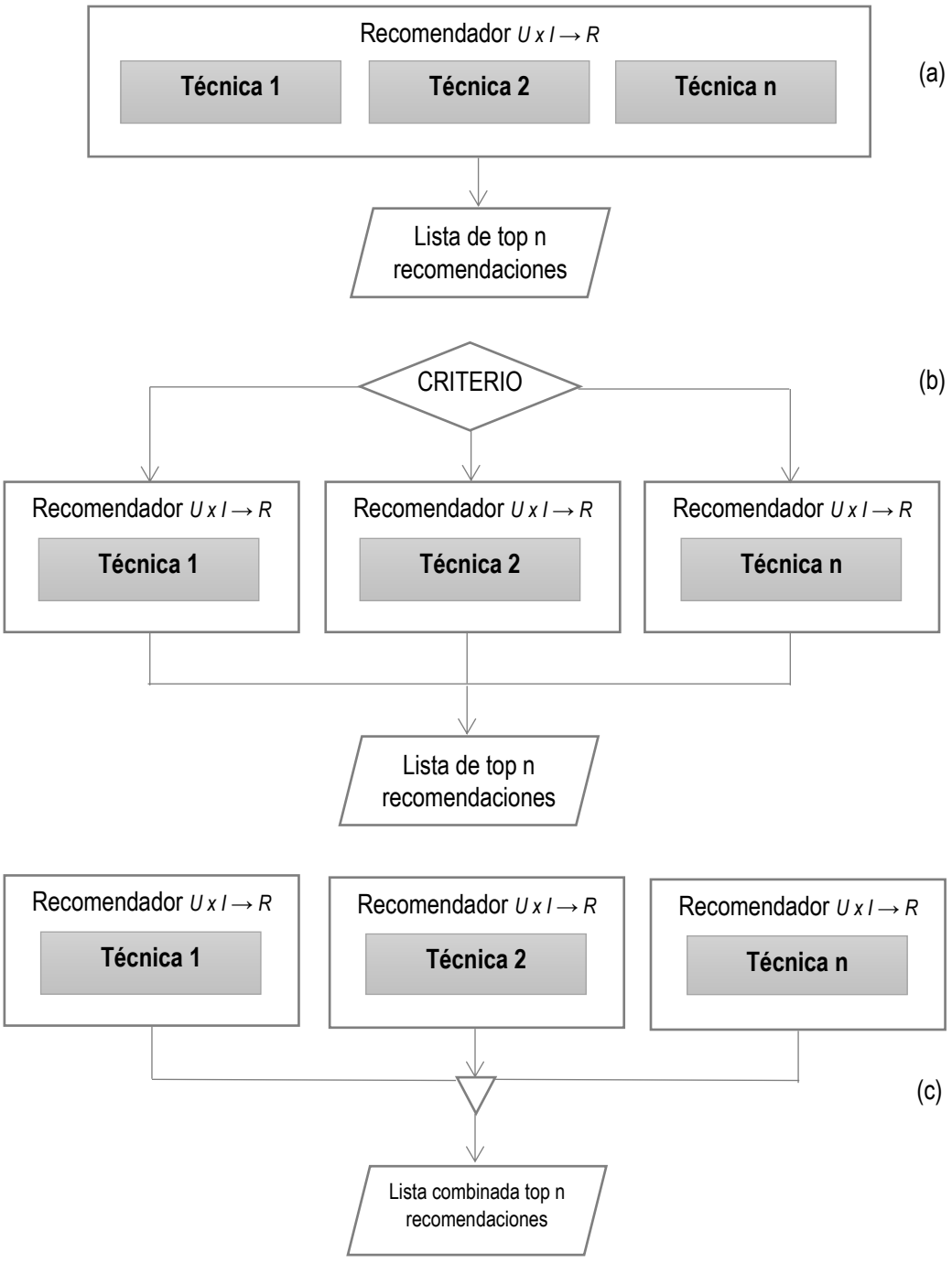
2.4.2. Métodos de Hibridación

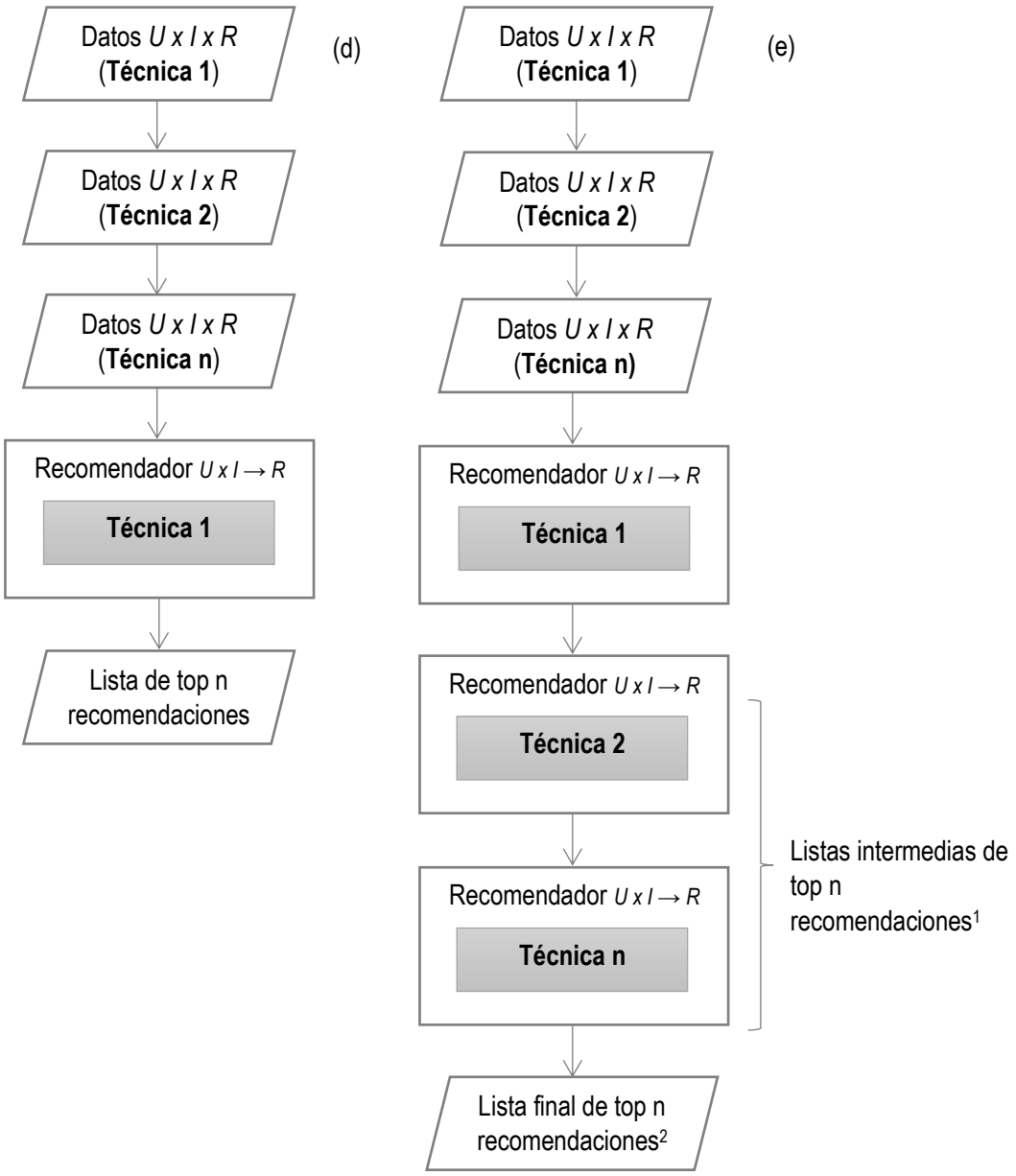
A partir del análisis de una serie de técnicas de recomendación propuestas a la fecha (año 2002), en términos de los datos de entrada y de las fases de los procesos de recomendación, esto es, los algoritmos, Burke (R. Burke, 2002) identificó y formalizó siete métodos recurrentes de hibridación en los sistemas que combinan dos o más técnicas de recomendación de naturaleza distinta, a saber: (1) ponderación, (2) conmutación, (3) mezcla, (4) combinación de características, (5) hibridación en cascada, (6) aumento de características y (7) hibridación de meta-nivel. En la Figura 2.3, que es una figura original de esta investigación, se muestran los diagramas de flujo que representan estos siete métodos de hibridación.

En detalle, la hibridación por ponderación consiste en predecir los *ratings* que los usuarios asignarían a los ítems a partir de los resultados ponderados de todas las técnicas de recomendación disponibles en el sistema de recomendación, con la ventaja de que resulta relativamente fácil realizar ajustes a la hibridación propiamente dicha, y la desventaja que representa la suposición implícita de que el valor relativo obtenido por todas y cada una de las técnicas combinadas es uniforme a lo largo del conjunto de todos los ítems.

La hibridación por conmutación, por su parte, consiste en la utilización de algún criterio que permita conmutar entre las técnicas de recomendación disponibles. Como se discutirá a continuación en mayor detalle, las ventajas de este método de hibridación derivan de la posibilidad de dotar a los sistemas de recomendación resultantes de la habilidad de ser conscientes de las fortalezas y debilidades de las diferentes técnicas que los constituyen. Al contrario, este método de hibridación supone la introducción de cierta complejidad extra relativa a la selección de los criterios de conmutación.

En lo que respecta al método de hibridación mediante mezcla, consiste, simplemente, en la predicción simultanea de *ratings* mediante distintas técnicas de recomendación, y la presentación subsecuente de los ítems correspondientes a las top-n recomendaciones en una única lista combinada, lo que con frecuencia requiere algún tipo de criterio de ordenamiento de los resultados o de selección de la mejor recomendación de entre varias procedentes de distintas técnicas. La ventaja más importante de este método, al igual que el método anteriormente descrito, es que puede ser de utilidad en la solución del problema de "arranque en frío" en su modalidad del "elemento nuevo", así como en preservar la capacidad propia de la técnica de filtrado colaborativo y de la técnica demográfica (no abordada en este documento) de descubrir nichos de usuarios.


Recomendador U x I → R
Técnica 1
Técnica 2
Técnica n
(a)
Lista de top n recomendaciones
CRITERIO
(b)
Recomendador U x I → R
Técnica 1
Recomendador U x I → R
Técnica 2
Recomendador U x I → R
Técnica n
Lista de top n recomendaciones
Recomendador U x I → R
Técnica 1
Recomendador U x I → R
Técnica 2
Recomendador U x I → R
Técnica n
(c)
Lista combinada top n recomendaciones



1. "Características generadas (ratings preliminares)" en el caso del método de aumento de características. "Modelos generados (modelos de preferencias)" en el caso del método de meta-nivel.

2. "Lista de top n recomendaciones" en los dos casos antes mencionados.

Figura 2.3. Métodos de hibridación de técnicas en sistemas de recomendación (notación basada en la notación de diagramas de flujo): a) hibridación por ponderación, b) hibridación por conmutación, c) hibridación mediante mezcla, d) hibridación por combinación de caract., e) hibridación en cascada, hibridación por aumento de caract. e hibridación de meta-nivel.

El método de combinación de características, por su parte, implica considerar cierta información de los datos de entrada de una o más técnicas de recomendación como características adicionales en los datos de entrada de una técnica principal, a fin de obtener un *dataset* aumentado. Sirva de ejemplo el caso en el que los *ratings* provistos por los usuarios en el contexto de una técnica de recomendación basada en filtrado colaborativo se consideran características adicionales en el *dataset* de un sistema de recomendación basada en contenido (características del contenido de los ítems). Evidentemente, este método de hibridación, específicamente la combinación de las técnicas antes mencionadas, permite explotar *ratings* colaborativos sin la necesidad de depender completamente de ellos, logrando que los sistemas resultantes sean menos sensibles ante la modalidad del "elemento nuevo" del problema de "arranque en frío".

Respecto al método de hibridación en cascada cabe mencionar que es un procedimiento escalonado en el que las técnicas de recomendación se emplean secuencialmente para producir listas de recomendaciones que se refinan sucesivamente al considerarse como los datos de entrada de las técnicas subsiguientes. Una de las ventajas más relevantes de este método es que permite evitar el uso de una nueva técnica en los casos en los que los *ratings* resultantes de la técnica precedente son lo bastante altos como para que los ítems correspondientes sean recomendados (aun operando las técnicas subsiguientes); como se puede deducir, los casos contrarios (*ratings* tan bajos que de ninguna manera los ítems correspondientes serían recomendados) también representan una ventaja de este método de hibridación.

El método de hibridación por aumento de características consiste en predecir *ratings* empleando una primera técnica de recomendación, y posteriormente emplear dichos *ratings* como datos de entrada en una o más técnicas subsecuentes. A diferencia del método en cascada, en este caso no son las listas de las top-n recomendaciones producidas previamente las que se refinan explotando las técnicas subsiguientes, sino que son los *ratings* resultantes los que se utilizan como parte de los datos de entrada de otras técnicas. Este método es especialmente relevante en los casos en los que es prioritario mejorar el rendimiento de una técnica de recomendación sin implicar una modificación propiamente dicha, sino solamente el "aumento" de sus datos de entrada.

Por último, el método de hibridación de meta-nivel implica utilizar un modelo completo generado mediante una técnica de recomendación como "entrada" de una segunda técnica. A diferencia del método anteriormente descrito, en este caso no son los *ratings* u otras características del modelo generado por la técnica precedente las que se utilizan como datos de entrada de una segunda técnica, sino dicho modelo en su totalidad. Aquí el concepto de "modelo" hace referencia a cualquier representación de los intereses del usuario, por ejemplo, vectores de términos en el contexto de las técnicas de recomendación basada en contenido. Uno de los posibles beneficios del uso de este método de hibridación es, precisamente, que hace posible incorporar la noción de perfil de preferencias en técnicas en las que no está naturalmente presente, como en las técnicas de filtrado colaborativo.

2.5. Sistemas de Recomendación Sensibles al Contexto

2.5.1. Antecedentes

A inicios de la década de los 2000, se formalizó el concepto de un nuevo tipo de sistema de recomendación (no relacionado con el tipo de técnica utilizada para el cálculo de las similitudes o la predicción de los *ratings*) con la habilidad de sugerir elementos de posible interés para el usuario en circunstancias determinadas, esto es, considerando información sobre el contexto en el que las recomendaciones ocurren: sistema de recomendación sensible al contexto (Adomavicius & Tuzhilin, 2008). Este tipo de sistema de recomendación surgió con el objetivo de preservar información potencialmente útil para la predicción de *ratings*, lo que finalmente podría resultar en recomendaciones más relevantes desde el punto de vista del usuario.

En este sentido, es importante mencionar que el concepto de "contexto", desde su aparición en el año 1994 en el campo de la computación móvil (Schilit, Adams, & Want, 1994), ha sido estudiado en diversas áreas de las ciencias de la información, e incluso en otras disciplinas, dando lugar a un sin número de definiciones heterogéneas. Por ejemplo, Dey (Dey, 2001) define "contexto" como: "cualquier información que se puede usar para caracterizar la situación de una entidad. Una entidad es una persona, un lugar o un objeto que se considera relevante para la interacción entre un usuario y una aplicación, incluyendo al usuario y a la aplicación mismos.". Como se verá más adelante, de la definición anterior se asume que pueden existir distintos tipos de información susceptibles de considerarse información contextual o del contexto, lo que puede dar lugar a más discrepancias en el camino hacia una definición universal de "contexto".

En (Bazire & Brézillon, 2005) se recogen y analizan algunas de estas discrepancias. En concreto, en ese trabajo se analizaron más de 150 definiciones diferentes existentes a la fecha de publicación (2005); dichas definiciones provenían no solo de áreas de las ciencias de la información, sino de otras disciplinas como filosofía, economía y negocios. Resulta interesante mencionar que para ello se utilizaron las técnicas *LSA* y *Semántica Tree-based Object Navigator and Editor* (*STONE*). Esta última es, formalmente, una herramienta de software para análisis semántico basado en cuestionarios de respuestas binarias (Poitrenaud, 2001). La conclusión fue que la definición de "contexto" depende del campo de conocimiento del que se trate, es decir, que no es posible llegar a una única definición integradora de dicho concepto.

No obstante, a partir de dicha conclusión, los autores fueron capaces de desarrollar un modelo de "contexto", en el que se representan los componentes (y las relaciones) que intervienen en la situación en la que el contexto es relevante, a saber: el usuario, el elemento (en un entorno determinado) y, eventualmente, un observador, dando lugar más bien a un modelo de "situación" en el que el contexto es un componente "externo" que interfiere en menor o mayor medida en el resto de componentes. Como resultado, se puede considerar parte del contexto cualquier cosa que sea relevante en un momento dado, y que pueda pertenecer a cualquiera de las categorías representadas por los componentes en el modelo.

De ahí que, en ocasiones, en las disciplinas de las ciencias de la computación y los sistemas de información se considere al usuario, al elemento con el que el usuario interactúa y al entorno en el que tiene lugar la interacción distintas fuentes de información contextual, dando lugar a distintas categorías o tipos de contexto. En (Adomavicius & Tuzhilin, 2008), además de formalizarse el concepto de "sistema de recomendación sensible al contexto", se analizan distintas interpretaciones del concepto de "contexto" a lo largo de diferentes áreas de investigación en las disciplinas de las ciencias de la computación y de los sistemas de información relacionadas con el área de sistemas de recomendación, por ejemplo, computación ubicua y móvil, comercio electrónico y minería de datos. En dicho trabajo se concluyó que el concepto de "contexto" es un concepto multifacético utilizado a lo largo de diferentes disciplinas, cada una de las cuales toma cierta postura y deja cierta huella en la definición del mismo.

Con el objetivo de esbozar las principales aplicaciones de los sistemas de recomendación sensibles al contexto (*CARS*, por sus siglas en ingles), Adomavicius y Tuzhilin (Adomavicius, Mobasher, Ricci, & Tuzhilin, 2011) analizaron una muestra de sistemas de recomendación existentes a la fecha (2011), clasificándolos de acuerdo al tipo de información contextual explotada; para ello definieron cuatro categorías distintas de información contextual o contexto: (1) contexto físico, el cual representa, entre otras cosas, la fecha y hora, posición geográfica (datos espacio-temporales) y actividad realizada por el usuario en el momento de la interacción con el sistema, (2) contexto social, esto es, la presencia (o ausencia) de otras personas y sus roles, independientemente de si participan en el proceso de interacción o no, (3) contexto del medio de interacción, el cual representa al dispositivo utilizado para acceder al sistema, así como al tipo de contenido al que se accede o la forma de acceder al mismo y (4) contexto modal, esto es, los pensamientos y sentimientos del usuario: el objetivo de la interacción, conocimientos y experiencia previa, su estado de ánimo, entre otros.

Esta clasificación en realidad extiende la clasificación de información contextual propuesta por Fling (Fling, 2009) en el contexto del diseño y desarrollo de aplicaciones móviles. Dicha clasificación comprende dos categorías de contexto de "alto nivel": (1) el modelo mental del usuario en el entendimiento de la circunstancia en la que se encuentra al momento de interactuar con una aplicación móvil y (2) el modo, medio y entorno en el que se lleva a cabo el proceso de interacción, el cual comprende tres categorías de contexto de "bajo nivel": (a) el contexto modal, (b) el contexto del medio de interacción y (c) el contexto físico.

2.5.2. Representación y Modelado de Información Contextual en Sistemas de Recomendación

Al incorporar información contextual al proceso de recomendación como categorías explícitas adicionales de datos, la función con la que se suele describir al proceso por el que se predicen las preferencias a largo plazo del usuario, esto es los *ratings*, pasa de ser la función mostrada en la Fórmula 2.10, a ser la función mostrada en la Fórmula 2.11 (Adomavicius & Tuzhilin, 2008), esto es, de ser un sistema de recomendación tradicional o de dos dimensiones (2D), a ser un sistema de recomendación de tres (3D) o más dimensiones en el caso de que se considere que la dimensión contextual a su vez comprende distintas dimensiones (modelo multidimensional).

$$R: User\ x\ Item \rightarrow Rating \quad (2.10)$$

$$R: User\ x\ Item\ x\ Context \rightarrow Rating \quad (2.11)$$

Donde:

- *User* y *Ítem* representan los dominios de los usuarios y los elementos, respectivamente.
- *Rating* representa el dominio de los *ratings* explícitamente dados por los usuarios o aprendidos por el sistema (implícitos)
- *Context* representa el dominio de la información contextual relevante para la predicción de *ratings* para los elementos desconocidos por el usuario.

Es importante mencionar que el proceso de predicción representado por estas funciones bien podría interpretarse como predicción genérica de preferencias (formalmente, utilidades), y no precisamente como predicción de *ratings*. No obstante, como se vio anteriormente, el modelo de *ratings* es, sin lugar a dudas, el modelo de utilidad más popular en los sistemas de recomendación, particularmente en los sistemas de filtrado colaborativo.

La manera más evidente de modelar este tipo de datos en el proceso de recomendación es mediante estructuras jerárquicas. No obstante, bajo este enfoque de modelado se asume que la información contextual comprende un conjunto de atributos observables que se conocen *a priori*, y cuya estructura no cambia significativamente a lo largo del tiempo. Esta perspectiva en la definición de contexto se conoce como "vista de representación", y fue descrita por primera vez por Dourish (Dourish, 2004) en el año 2004; en contraposición a la "vista de representación", Dourish describió además la denominada "vista de interacción", en la cual se asume que el ámbito de los atributos contextuales se define dinámicamente, y que son ocasionados, y no estables.

Con el objetivo de representar dichas estructuras jerárquicas, en la literatura de sistemas *CARS* se han utilizado desde taxonomías y tesauros (Bao, Zheng, & Mokbel, 2012); (Hawalah & Fasli, 2014); (Nakatsuji & Fujiwara, 2014), hasta modelos conceptuales, como el modelo de clases de *Unified Modeling Language* (*UML*), el modelo relacional de datos y el modelo multidimensional de datos utilizado por el enfoque de análisis multidimensional de grandes volúmenes de datos, *Online Analytical Processing* (*OLAP)* (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005)*;* (Mokbel & Levandoski, 2009); (Mettouris & Papadopoulos, 2013), pasando por ontologías

(Web semántica) (Vallet, Castells, Fernandez, Mylonas, & Avrithis, 2007); (Hong et al., 2009); (Yılmaz & Erdur, 2012); (Ruotsalo et al., 2013); (Karpus, Vagliano, Goczyła, & Morisio, 2016).

En lo que respecta al enfoque de modelado contextual basado en ontologías, vale la pena destacar el trabajo de Vallet y colaboradores (Vallet et al., 2007). Bajo la suposición de que no todas las preferencias de los usuarios de los sistemas de recuperación de información son relevantes en todas las situaciones, por lo que es necesario contextualizarlas teniendo en cuenta las actividades que los usuarios desarrollan en el momento de la interacción con el sistema, los autores propusieron un método dirigido por ontologías para la representación dinámica del contexto de actividades de recuperación de información. En detalle, en ese trabajo se definió el concepto de "contexto semántico de tiempo de ejecución" como el conjunto de temas subyacentes al desarrollo de las actividades de los usuarios en una unidad de tiempo. Dicho conjunto de temas es representado por un conjunto de conceptos ponderados provenientes de una ontología de dominio en la cual las preferencias de los usuarios se representan igualmente como conceptos. De modo que la contextualización de las preferencias radica en el cálculo de la similitud semántica entre cada concepto en las preferencias y cada concepto en el contexto correspondiente.

Asimismo, en (Yılmaz & Erdur, 2012) se propuso un modelo de contexto basado en ontologías para un sistema multi-agente sensible al contexto para provisión de contenido y servicios sobre puntos de interés (*Points Of Interest, POIs*) denominado "iConAwa". En concreto, dicho modelo consiste en una ontología con la que se modela, por un lado, información contextual de bajo nivel: información geográfica, información temporal e información del dispositivo móvil y red utilizada para interactuar con el sistema y, por otro lado, las preferencias de los usuarios; de modo que las preferencias se consideran parte del contexto, y no viceversa. Además, el modelo comprende una serie de reglas basadas en *SWRL* que permiten la inferencia de información contextual de alto nivel a partir de la información de bajo nivel modelada en la ontología. Cabe mencionar que, a diferencia del enfoque de modelado contextual propuesto en (Vallet et al., 2007), en este trabajo, el dominio (*POIs*) se modela en una segunda ontología.

En el campo de la computación ubicua ha habido ya distintos esfuerzos hacia el desarrollo de un modelo contextual de alto nivel reutilizable a lo largo de distintos sub-dominios en el dominio de los sistemas multi-agente sensibles al contexto (Ranganathan & Campbell, 2003); (H. Chen, Finin, & Joshi, 2003); (Wang, Zhang, Gu, & Pung, 2004). No obstante, estos trabajos han inspirado el desarrollo de distintos tipos de sistemas de software en un esfuerzo por integrar capacidades de sistemas sensibles al contexto, tal es el caso del sistema "iConAwa", el cual puede considerarse también un sistema de recomendación.

Adomavicius y colaboradores (Adomavicius et al., 2011) fueron un paso más allá de las perspectivas en la definición del concepto de contexto propuestas en (Dourish, 2004) y definieron seis dimensiones distintas de información contextual en el campo particular de los sistemas de recomendación a partir de las posibles combinaciones entre los posibles valores de dos características identificables de los factores contextuales en dicho contexto. En la Figura 2.4, originalmente presentada en (Adomavicius et al., 2011), se muestran dichas dimensiones de información contextual y características de factores contextuales. En concreto, los autores definieron las siguientes características y valores: (1) el conocimiento que tiene el sistema de recomendación de los factores contextuales, esto es, si dichos factores son: (a) factores completamente observables, (b) factores parcialmente observables o (c) factores inobservables, y (2) si los factores contextuales cambian o no a lo largo del tiempo, es decir, si se trata de: (a) factores estáticos o (b) factores dinámicos.

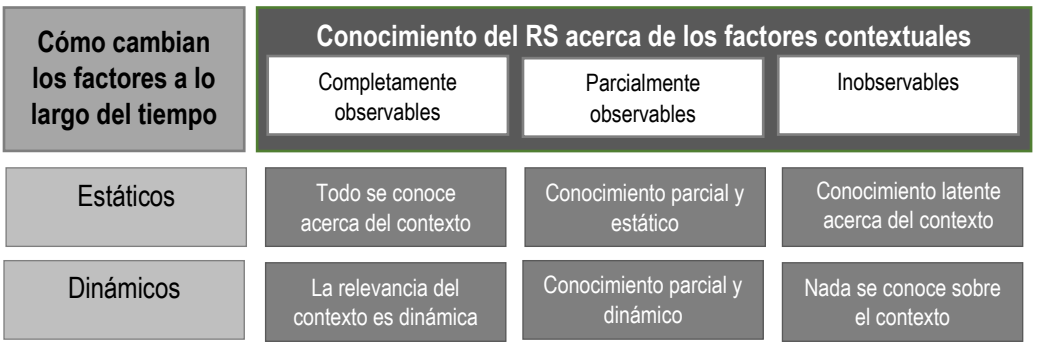


Figura 2.4. Dimensiones de información contextual en sistemas de recomendación.

En el caso particular del tipo de información contextual que comprende atributos observables parcialmente o atributos inobservables, los modelos de variables latentes como *PLSA* y *LDA* han sido ampliamente explotados con fines de representación y modelado bajo la suposición de que las interacciones de los usuarios involucran un conjunto relativamente pequeño de "estados" contextuales que pueden explicar el comportamiento de los usuarios en distintos puntos a lo largo de las interacciones (Mobasher, 2014). En ese sentido, estos modelos se pueden considerar tipos de modelos probabilísticos gráficos en los que se utiliza notación de grafos para expresar estructuras de dependencia condicional entre variables aleatorias (Koller & Friedman, 2009).

Sirva de ejemplo el trabajo de Hariri, Mobasher y Burke del año 2012 (Hariri, Mobasher, & Burke, 2012). En dicho trabajo se presentó un sistema de recomendación de canciones sensible al contexto. Por un lado, los autores propusieron extraer las etiquetas más frecuentemente asociadas a las canciones en sitios Web de etiquetado social como last.fm y, una vez obtenidas las etiquetas, emplear el modelo *LDA* para mapear las secuencias de interacción de los usuarios a secuencias de tópicos latentes que capturan tendencias más generales en los intereses de los usuarios. Por otro lado, se empleó minería de patrones secuenciales para obtener, a partir de las secuencias de tópicos latentes, patrones que representan secuencias frecuentes de transiciones entre tópicos que a su vez representan contextos; para ello se empleó una base de datos de *playlists* compiladas manualmente. Cabe mencionar que, en este caso, la naturaleza no estable (dinámica) de los factores contextuales que componen la información contextual hace necesaria la utilización de la técnica de minería de datos mencionada.

Asimismo, Xu y colaboradores (Xu et al., 2015) propusieron un método (y *framework*) para la recomendación sensible al contexto de *POIs* turísticos en el que los *POIs* se obtienen a partir de fotografías geo-etiquetadas en sitios Web de intercambio de imágenes como Flickr utilizando la técnica de agrupamiento de datos basada en densidad conocida como "DBSCAN". Por un lado, los autores proponen el perfilamiento de cada *POI* mediante la construcción de su historial de visitas; por otro lado, ellos proponen perfilar cada usuario a partir de la construcción de su historial de viajes empleando una técnica basada en grafos, así como utilizar el modelo *PLSA* para extraer la distribución de los tópicos latentes en dicho historial, la cual se interpreta como las preferencias del usuario. En ambos casos se toman en cuenta atributos contextuales, a saber, la temporada y el clima, los cuales se extraen de los *timestamps* de las visitas y los viajes.

De manera similar, en (Ren, Song, E, & Song, 2017) se propuso un método probabilístico y sensible al contexto de descomposición de matrices para recomendación de *POIs* en redes sociales basadas en localización. Por un lado, en este trabajo se propone emplear el modelo *LDA* para aprender la distribución de tópicos latentes de cada usuario y cada *POI* a partir de la agregación de los comentarios textuales asociados a ellos en las redes sociales y, subsecuentemente, extraer un índice de preferencia para cada par usuario-punto de interés a partir del cálculo de la similitud basada en correlación respecto a las distribuciones de tópicos correspondientes. Por otro lado, con el fin de integrar dicho índice de preferencia junto con otros índices de correlación derivados de

factores contextuales de tipo geográfico y social y, finalmente, permitir la predicción del *rating* que un usuario activo otorgaría a un *POI* candidato, los autores proponen una extensión a la técnica *Probabilistic Matrix Factorization* (*PMF*) en la cual las preferencias de los usuarios por los *POIs* se representan como los productos de los índices de preferencia y los índices de factores contextuales.

Cabe mencionar que los factores contextuales considerados en ambos trabajos se pueden considerar los bloques de construcción de factores inobservables pero estables (estáticos), esto es, los tópicos latentes. Como se verá en el próximo capítulo de este documento, tal es el caso del enfoque de modelado y representación de información contextual propuesto como parte de esta tesis doctoral, con la salvedad de que la prioridad en esta investigación no es el perfilado de los usuarios sino la mera minería de factores contextuales inobservables y estáticos a partir de los historiales de *check-ins* ocurridos en los *POIs*, en este caso establecimientos de alimentos y bebidas.

2.5.3. Paradigmas en Integración de Información Contextual en Sistemas de Recomendación

De acuerdo con Adomavicius y Tuzhilin (Adomavicius & Tuzhilin, 2008), la tendencia más reciente en la incorporación de información contextual en sistemas de recomendación corresponde al modelado y aprendizaje de preferencias de usuario sensibles al contexto, a la cual ellos denominan formalmente "obtención y estimación de preferencias contextuales", y que comprende la aplicación de técnicas inteligentes de minería de datos y aprendizaje computacional. No obstante, los procesos de recomendación sensible al contexto basados en dicho enfoque pueden además tomar una de tres formas distintas dependiendo de la fase del proceso de recomendación (representado por la Fórmula 2.11) en la que se explota la información contextual, en otras palabras, dependiendo del componente del proceso en el que se explota dicha información (ver Figura 2.5).

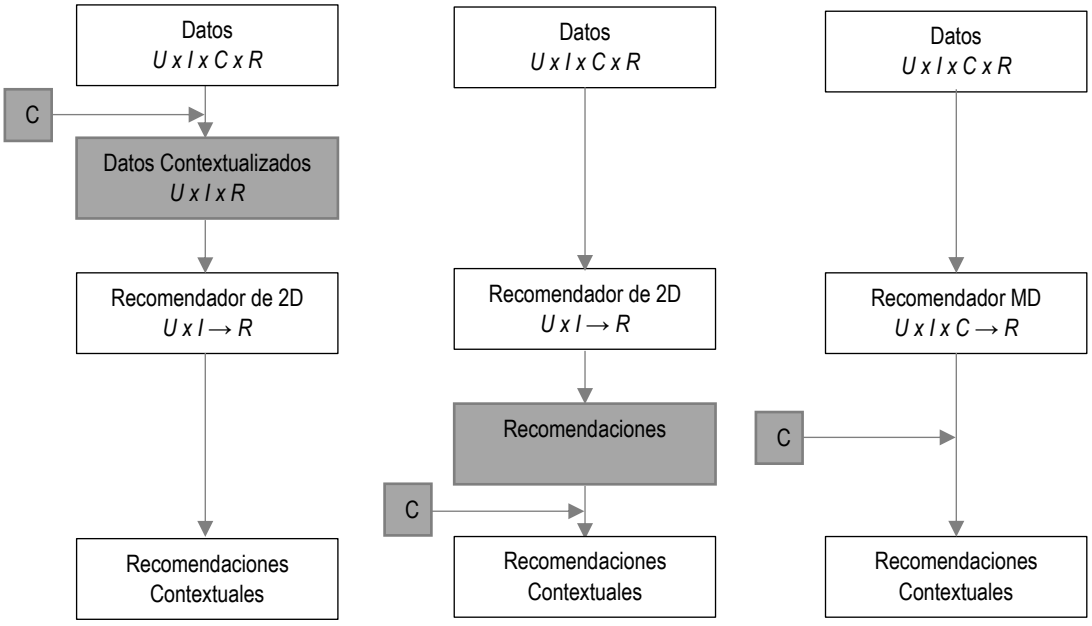


Figura 2.5. Paradigmas en la incorporación de información contextual en sistemas de recomendación sensible al contexto: (a) Pre-filtrado contextual, (b) Post-filtrado contextual y (c) Modelado contextual.

Como se puede apreciar en la Figura 2.5, la cual aparece originalmente en (Adomavicius & Tuzhilin, 2008), el paradigma denominado "modelado contextual" da lugar a procesos de recomendación verdaderamente

multidimensionales (Fórmula 2.11), a diferencia de los paradigmas llamados “pre-filtrado contextual” y “post-filtrado contextual”, los cuales se pueden basar en procesos de recomendación de 2D.

En detalle, el paradigma “pre-filtrado contextual” implica utilizar la información contextual en la etapa temprana del proceso de recomendación de 2D para seleccionar o construir (pre-filtrar) el espacio de datos de usuarios e *ítems* (*U x I*) más relevante para la generación de recomendaciones. En otras palabras, este paradigma se puede interpretar como un paradigma de reducción de procesos de recomendación sensible al contexto multidimensionales a procesos de recomendación de 2D, lo que permite el aprovechamiento de cualquier técnica de recomendación de 2D propuesta en la literatura (al igual que en el caso del paradigma de “post-filtrado contextual”).

Por su parte, el paradigma “post-filtrado contextual” implica ignorar la información contextual en primera instancia y explotarla en la etapa tardía del proceso de recomendación de 2D para alterar el resultado final de dicho proceso: típicamente, la lista ordenada de las top-n recomendaciones. En este sentido, el ajuste de la lista de las top-n recomendaciones se puede llevar a cabo, tanto filtrando las recomendaciones que son irrelevantes dada la información contextual, como ajustando el orden de los *ítems* en dicha lista (reordenando) empleando algún criterio relacionado con la información contextual dada.

Los enfoques de recomendación que siguen este paradigma de procesamiento de información contextual se pueden clasificar a su vez en enfoques basados en heurísticas y enfoques basados en modelos. Los enfoques basados en heurísticas consisten en la identificación de los atributos de los ítems que son comunes a los ítems que aparecen en la lista de las top-n recomendaciones y a las preferencias del usuario (por dichos atributos) en el contexto dado. Para los efectos de esta tesis doctoral, es especialmente relevante el caso de los enfoques de post-filtrado del contexto basado en modelos, cuyo fundamento es, evidentemente, la construcción de modelos predictivos que permitan calcular la probabilidad de que el usuario seleccione un ítem en un contexto determinado y utilizar dicha probabilidad para ajustar la lista de las top-n recomendaciones de una de las dos formas anteriormente mencionadas.

Por último, el paradigma de “modelado contextual” consiste en utilizar la información contextual directamente en la función de recomendación como un predictor explícito de los *ratings* que los usuarios darían a los ítems, dando lugar a un proceso de recomendación de múltiples dimensiones (ver Fórmula 2.11).

Al igual que en el caso del paradigma de “post-filtrado contextual”, los enfoques de recomendación que siguen el paradigma de “modelado contextual” se pueden clasificar a su vez en enfoques basados en heurísticas y enfoques basados en modelos. Evidentemente, la mayoría de estos enfoques consiste en la adaptación a espacios de recomendación multidimensionales de técnicas de recomendación (2D) existentes. En la primera categoría, se encuentran aquellos enfoques en los que se propone la extensión de las métricas de similitud basada en correlación y de similitud basada en distancia utilizadas típicamente en los sistemas de recomendación de 2D. Lo mismo sucede en el caso de los enfoques de recomendación basados en modelos: se trata, principalmente, de enfoques en los que se presentan extensiones a los algoritmos de recomendación (2D) que emplean técnicas estadísticas, técnicas de aprendizaje computacional y técnicas de minería de datos. No obstante, en esta segunda categoría existen también enfoques puramente multidimensionales, los cuales van más allá del enfoque basado en el modelo multidimensional de datos del método de análisis de grandes volúmenes de datos, *OLAP*, mencionado en la subsección anterior.

2.6. Discusión General

A partir de los resultados del análisis del estado del arte resumidos en las tablas 1 y 2 se puede deducir que los dominios de aplicación de los sistemas de recomendación representados por terceros sectores económicos relacionados con actividades parcialmente soportadas como consecuencia de la afluencia turística, por ejemplo, la industria de la restauración, actualmente no cuentan con el apoyo de técnicas y herramientas de

recomendación sensible al contexto en la misma medida en la que el dominio representado por la industria del ocio cuenta con dicho apoyo, aun cuando ésta industria no está necesariamente relacionada con actividades soportadas por turistas.

Asimismo, las pocas contribuciones relacionadas con sistemas de recomendación sensible al contexto existentes para el dominio de la restauración no aprovechan los posibles puntos de convergencia entre las tecnologías de la Web semántica y los modelos estadísticos de clases latentes para la definición de técnicas más potentes de representación y modelado de información contextual y de técnicas de recomendación más efectivas, y emplean, bien tecnologías de la Web semántica, comúnmente el lenguaje de definición y descripción de vocabularios, *OWL*, y el lenguaje de reglas, *SWRL*, bien modelos estadísticos de clases latentes, comúnmente el modelo generativo de tópicos, *LDA*.

En este sentido, dichos trabajos no exploran la importancia de la información contextual compleja relacionada con las situaciones sociales de los usuarios, la cual se presume que es tan importante como la información contextual compleja del tipo temporal, y otros tipos comunes de información contextual simple, como la información de ubicación, en el dominio particular de la restauración.

A partir de los resultados del estudio del estado del arte en sistemas de recomendación basada en conocimiento y Web Semántica (Tabla 2), es posible observar una tendencia en las propuestas actuales hacia el uso de datos publicados bajo el enfoque de Datos Enlazados (*Linked Data*). No obstante, se requiere de más investigación en el empleo de este enfoque en conjunto con enfoques de recomendación sensible al contexto, más allá de enfoques de recomendación tradicional.

En lo que respecta particularmente a las métricas de similitud semántica basada en ontologías propuestas en la literatura de sistemas de recomendación basada en conocimiento (resultados resumidos en la Tabla 3), vale la pena mencionar que la mayoría de estas emplean, bien un enfoque basado en caminos en grafos, bien un enfoque basado en características ontológicas, pero muy pocas emplean un enfoque híbrido basado en caminos y en conjuntos de características. Esto con el objetivo de explotar, por un lado, relaciones taxonómicas y, por otro lado, relaciones no taxonómicas, incluyendo relaciones explícitas y relaciones implícitas.

Con esta tesis doctoral se pretenden aligerar los problemas antes mencionados mediante una serie de contribuciones principales en las siguientes líneas de investigación:

Representación de conocimiento: un modelo ontológico basado en *OWL* del dominio de la restauración que permita enlazar y, finalmente integrar, los vocabularios de las Interfaces de Programación de Aplicaciones (*Application Programming* Interfaces, *APIs*) de redes sociales basados en localización y sitios Web de opiniones de usuarios que proveen contenido heterogéneo en este dominio, utilizando un enfoque de *Linked Data*. Una técnica de modelado de información contextual basada en modelos probabilísticos generativos de tópicos, específicamente modelos *LDA*, que aproveche el modelo ontológico del dominio, así como una base de reglas de inferencia representadas bajo la notación *SPARQL Inferencing Notation* (*SPIN*), para la inferencia de información contextual temporal y social de alto nivel a partir de información relacionada de bajo nivel.

Razonamiento: una métrica de similitud semántica basada en ontologías que emplee un enfoque híbrido de caminos en grafos y de características ontológicas, y permita capturar, además de conocimiento taxonómico, conocimiento no taxonómico tanto explícito como inferido acerca del dominio de la restauración. Una técnica híbrida de filtrado colaborativo basado en modelos y de filtrado colaborativo basado en ítems bajo el enfoque de top-n recomendaciones, en donde el componente basado en modelos lo represente el modelo *LDA* de información contextual de alto nivel (modelo probabilístico de clases latentes), y el componente basado en ítems lo representa le métrica de similitud semántica basada en ontologías.

2.7. Web Semántica

El acuñamiento del término "Web Semántica" se atribuye a Tim Berners-Lee, James Hendler y Ora Lassila (Berners-Lee et al., 2001) en un intento por hacer posible una visión de la World Wide Web en la que el significado del contenido de las páginas Web se dota de estructura, dando lugar a un entorno en el que existen agentes de software que son capaces, no solo de mostrar dicho contenido, sino de procesarlo y "entenderlo" automáticamente para llevar a cabo tareas útiles para los usuarios rápidamente.

Visualizada, no como una Web separada de la Web basada en documentos y enlaces de hipertexto -Web tradicional, sino más bien como una evolución o extensión de ésta, en la que se otorga a la información significado bien definido, a fin de habilitar el trabajo colaborativo entre usuarios (seres humanos) y agentes de software, Berners-Lee y colaboradores definieron los "bloques de construcción" principales necesarios para su materialización: lenguajes de representación de conocimiento, ontologías y agentes.

A partir de dichos bloques de construcción primigenios, y sobre la base representada por el lenguaje de marcado *eXtensible Markup Language* (*XML*), el lenguaje de esquema *XML Schema*, la tecnología *Uniform Resource Identifier* (*URI*) y el estándar de codificación de caracteres UNICODE, todos ellos componentes esenciales de la Web tradicional, se formalizó una primera aproximación a la arquitectura de la Web Semántica, la cual se encuentra actualmente en desarrollo bajo el liderazgo del *W3C*. Dicha arquitectura se muestra en la Figura 2.6.[1]

En este contexto, vale la pena abrir un paréntesis para mencionar que el *W3C* define Web Semántica como: "Un *framework* común para el intercambio y reutilización de datos entre aplicaciones, empresas y comunidades. Un esfuerzo colaborativo liderado por el *W3C* que cuenta con la participación de un gran número de socios procedentes de la academia y de la industria".

La capa de la arquitectura representada por *URI* y UNICODE en conjunto permite el manejo consistente de la información independientemente del sistema de escritura en el que se encuentre representada, así como la identificación inequívoca de la misma: su nombrado y localización, dentro de la Web.

La capa representada por *XML* y *XML Schema* representa, por un lado, el mecanismo para la transmisión y "lectura" de la información estructurada entre los agentes en la Web Semántica y, por otro lado, el fundamento del mecanismo que permite recuperar y consultar el significado de dicha información, y no solo la recuperación de los documentos propiamente dichos. De hecho, *XML* permite dotar a los documentos de cierta estructura, pero no permite indicar nada acerca del significado de la misma.

La capa en la que descansan las tecnologías *RDF* (Cyganiak et al., 2014) y *RDFS* (Brickley, Guha, & McBride, 2014) representa el núcleo de la arquitectura, ya que permite expresar el significado de la información o, propiamente, de los recursos, donde un recurso es cualquier entidad abstracta o del mundo real, como un número o una entidad física. Para ello define una estructura de tripletas, en la que cada tripleta está compuesta por un sujeto, un predicado y un objeto, como ocurre con cualquier sentencia fundamental; con esta representación se pretende ofrecer una manera natural de describir la vasta mayoría de los datos procesados por las máquinas. Así, un conjunto de tripletas representa una "red" de información sobre entidades relacionadas, y debido a que *RDF* emplea *IRIs* (del inglés *Internationalized Resource Identifier*) para codificar dicha información en documentos, se asegura que los conceptos no son solo palabras en el documento sino definiciones únicas que pueden ser accedidas por cualquier persona o agente en la Web. Además, a diferencia de *XML* (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2008) y *XML Schema* (Fallside & Walmsley, 2004), *RDF* y *RDFS* están orientados a la integración eficaz y eficiente de los datos.

[1] https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

La capa denominada “vocabulario ontológico” permite la definición de colecciones de información en forma de taxonomías y conjuntos de reglas llamadas “ontologías”; con ellas se busca mejorar la funcionalidad y expresividad de la Web respecto a las capacidades provistas por la capa anterior. El término “ontología” proviene de la Filosofía, específicamente de la rama de la Metafísica, y hace referencia a teorías sobre la naturaleza de la existencia y de los tipos de cosas que existen; así como a la disciplina que se encarga del estudio de dichas teorías; fue introducido a las Ciencias de la Computación y las Ciencias de la Información por investigadores del campo de Inteligencia Artificial, y se utiliza actualmente para hacer referencia a definiciones formales de relaciones entre conceptos. Si bien no existe una definición universalmente aceptada del término “ontología” dentro de los campos antes mencionados, una de las definiciones más extendidas en la actualidad es la propuesta por Studer, Benjamins y Fensel (Studer, Benjamins, & Fensel, 1998): “una especificación formal y explícita de una conceptualización compartida”.

La capa de lógica representa el esfuerzo por añadir a la Web la capacidad de usar reglas para hacer inferencias automáticas acerca de los datos, tomar cursos de acción y responder a cuestionamientos, esto es, generar nuevo conocimiento a partir del conocimiento explícito disponible.

Las capas de primer nivel (“capa de comprobación” y “capa de confianza”) hacen referencia a una faceta fundamental del funcionamiento de los agentes en la Web Semántica: el intercambio de “pruebas de comprobación”; en concreto, este funcionamiento está relacionado con el funcionamiento de la capa anterior en el sentido de que las reglas se ejecutan bajo un mecanismo de seguridad (confianza) que permite evaluar sus resultados y determinar si es adecuado o no confiar en las pruebas proporcionadas. El desarrollo de estas capas, y de la “capa de firma digital”, que tiene como objetivo fundamental la identificación de alteraciones en los documentos dentro de la Web, es parte de la investigación en curso de los grupos de trabajo del *W3C*.

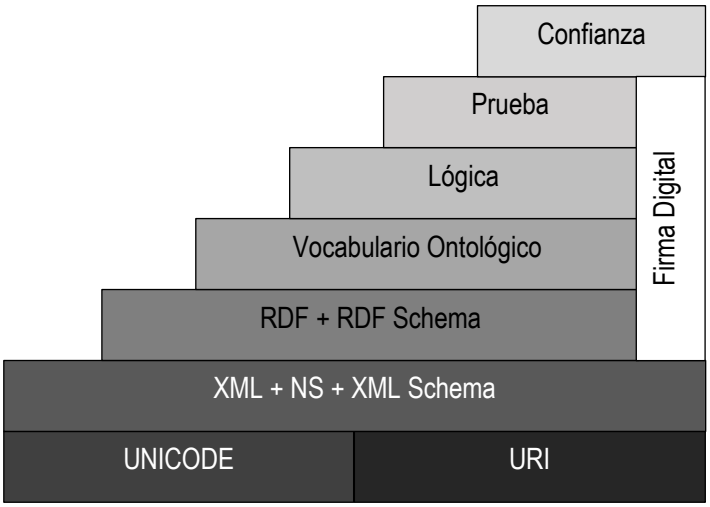


Figura 2.6. Primera aproximación a la arquitectura de la Web Semántica.

En las siguientes subsecciones de esta sección se describen detalladamente los bloques de construcción y las tecnologías de la llamada “pila de tecnologías” de la Web Semántica, la cual representa una visión más actualizada de la arquitectura descrita anteriormente, resultado del trabajo actual del *W3C*.[2]

[2] https://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24)

2.7.1. Vocabularios

En el contexto de la Web Semántica, los vocabularios son la manera de definir las relaciones entre los conceptos, así como los conceptos propiamente dichos, que describen y representan un área de interés. Más concretamente, los vocabularios permiten clasificar los términos (o conceptos) que describen y representan un área de interés, caracterizar las posibles relaciones entre ellos, y definir las posibles restricciones en su uso.

Si bien los términos "vocabulario" y "ontología" se usan en ocasiones como sinónimos ya que no existe una clara distinción entre ellos, se tiende a utilizar el término "ontología" para hacer referencia a los vocabularios más complejos y un tanto formales, mientras que para referirse a los vocabularios simples en los que no se requiere un formalismo estricto se emplea el término "vocabulario" directamente.

Uno de los principales usos de los vocabularios entre las aplicaciones basadas en las tecnologías de la Web Semántica radica en la integración de datos en aquellos casos en los que pueden existir ambigüedades entre términos definidos en distintos *datasets*, o cuando, a partir de la definición de un reducido conjunto de conceptos y relaciones se pueden descubrir relaciones implícitas potencialmente útiles en la fundamentación de procesos de razonamiento relativamente complejos (ver la subsección 2.7.3 de esta sección).

La pila de tecnologías de la Web Semántica proporciona una amplia paleta de tecnologías específicamente destinada a la definición estandarizada de distintas formas de vocabularios, a saber, *RDF*, *RDFS*, *Simple Knowledge Organization System (SKOS), OWL* y *Rule Interchange Format (RIF)*. La elección de la tecnología más apropiada en el contexto del desarrollo de una aplicación particular dependerá de la complejidad y el nivel de formalización requerido por el vocabulario a definir.

A continuación, se describen detalladamente las tecnologías *RDF*, *RDFS* y *OWL*, debido a su protagonismo en esta investigación. Además, la tecnología *RIF* se describe brevemente en la subsección 2.7.3. de esta sección. Nótese como, a menos de que se haga una diferenciación clara entre versiones, en las siguientes secciones se hace referencia a las versiones más recientes de las especificaciones producidas por el *W3C* para dichos estándares.

2.7.1.2. Resource Description Framework

Resource Description Framework (*RDF*) es un *framework* para la representación de información en la Web, cuyo núcleo es un modelo de datos basado en grafos, el cual representa una sintaxis abstracta que permite asociar una sintaxis concreta a su correspondiente vocabulario formal.

La estructura básica de dicha sintaxis es conocida como tripleta. Una tripleta permite nombrar relaciones entre pares de recursos, así como los recursos en sí. Formalmente, una tripleta se compone de tres elementos: un sujeto, un predicado o propiedad, el cual denota la relación, y un objeto.

Cada tripleta *RDF* debe interpretarse como una afirmación, denominada *declaración RDF*, de que la relación denotada por el predicado se mantiene entre los recursos denotados por el sujeto y el objeto. En este contexto, un recurso puede ser cualquier cosa en el mundo real, desde elementos físicos o documentos hasta números, o cualquier entidad abstracta.

Cada conjunto de tripletas *RDF* se conoce como grafo *RDF*, y se representa como un diagrama de nodos y arcos dirigidos en el que cada enlace nodo-arco-nodo representa una tripleta. De este modo, los nodos en cada enlace nodo-arco-nodo representan al sujeto y el objeto en la tripleta, mientras que el arco representa al predicado. La Figura 2.7 muestra la representación gráfica de un grafo *RDF* (una única tripleta).

A su vez, a un conjunto de grafos *RDF* se le conoce como *conjunto de datos RDF*, y comprende un *grafo predeterminado* y cero o más *grafos nombrados*. Un *grafo nombrado* es una pareja que consta de un *IRI* o nodo

en blanco, el cual representa el nombre del grafo, y el grafo *RDF* en sí. Un *grafo predeterminado* es simplemente un grafo *RDF* que opcionalmente puede estar vacío.

A un grafo o un conjunto de datos *RDF* serializado en una sintaxis *RDF* concreta se le conoce como documento *RDF*. Existen distintas sintaxis *RDF* concretas, por ejemplo, RDF/XML (Gandon & Schreiber, 2014), TURTLE (Beckett, Berners-Lee, Prud'hommeaux, & Carothers, 2014) y JSON-LD (Sporny, Longly, Kellogg, Lanthaler, & Lindstrom, 2014). Dichas sintaxis están fuera del alcance de esta revisión del estado del arte.

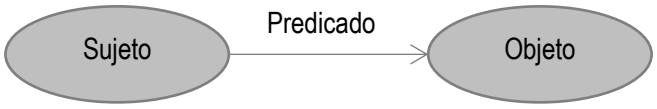


Figura 2.7. Representación gráfica de un grafo *RDF* (una única tripleta).

Los nodos en un grafo *RDF* pueden ser *IRIs*, literales o nodos en blanco. Específicamente, un sujeto puede estar representado por un *IRI* o un nodo en blanco, mientras que un objeto puede estar representado igualmente por un *IRI* o un nodo en blanco, o por una literal. Por otro lado, los predicados, los cuales realmente no están representados por nodos sino por arcos en un grafo *RDF*, son siempre *IRIs*. A continuación, se describe cada uno de los posibles tipos de nodo.

Un *IRI* en un grafo *RDF* es una cadena de caracteres UNICODE que se ajusta a la sintaxis definida por el estándar RFC 3987 del *Internet Engineering Task Force (IETF)* (Duerst & Suignard, 2005). En detalle, un *IRI* es una generalización de un *URI* que permite un rango más amplio de caracteres UNICODE. Al recurso denotado por un *IRI* se le conoce como el *referente* del *IRI*. Un *referente* puede ser fijo para algunos tipos de *IRIs* con significado particular, por ejemplo, aquellos *IRIs* que representan tipos de datos.

Un nodo en blanco en un grafo *RDF* es un nodo que no denota un recurso específico y proviene de un conjunto infinito. De hecho, este conjunto infinito y los conjuntos de todos los *IRIs* y todos los literales forman parejas de conjuntos disjuntos. Una tripleta *RDF* con nodos en blanco debe interpretarse como una declaración que indica que existe algo que forma parte de la relación denotada por el predicado, pero sin nombrarlo explícitamente.

2.7.1.2.1. Tipos de datos y Literales

En RDF, los tipos de datos son abstracciones que permiten representar valores. Dichas abstracciones son compatibles con las abstracciones utilizadas por el lenguaje *XML Schema* para la definición de tipos de datos para el lenguaje *XML*.

Formalmente, un tipo de dato se compone de un *espacio léxico*, un *espacio de valores* y un mapeo entre el espacio léxico y el espacio de valores. Un *espacio léxico* es un conjunto de cadenas de caracteres UNICODE. Un mapeo espacio léxico-espacio de valores es una función del espacio léxico al espacio de valores; de modo que cada miembro del espacio léxico se asocia a un único miembro del espacio de valores, esto es, a un valor, y es una *representación léxica* de dicho valor.

RDF no define abstracciones para la representación de valores comunes, tales como números enteros, cadenas de caracteres y fechas, sino que se adhiere a tipos de datos que se definen separadamente y se identifican con *IRIs*. De hecho, *RDF* reutiliza muchos de los tipos de datos incorporados de *XML Schema*.

Las literales en *RDF* permiten identificar valores mediante representaciones léxicas. Formalmente, una literal se compone de (1) una forma léxica, esto es una cadena de caracteres UNICODE en la Forma de Normalización C (*Normalization Form C*, *NFC*); dicha forma de normalización está definida en el estándar UAX #15 del *Unicode Consortium* (Davis & Whistler, 2016) y (2) un *IRI* que identifica a un tipo de dato, el cual, a partir de la forma

léxica (un miembro de un espacio léxico) y el correspondiente mapeo espacio léxico-espacio de valores, determina el valor asociado a la literal (un miembro de un espacio de valores). Una literal puede contener además un código de idioma que debe ajustarse a la sintaxis definida por el estándar RFC 5646 del *IETF* (Phillips & Davis, 2009). A las literales que contienen este tercer elemento se les denomina *cadenas con etiqueta de idioma.*

Finalmente, la Figura 2.8 muestra un ejemplo de grafo *RDF*. En detalle, este grafo se compone de tres tripletas que expresan los siguientes hechos simples: (1) una persona en una libreta de direcciones está asociada a una dirección, (2) la dirección asociada a una persona en una libreta de direcciones comprende una calle y (3) dicha dirección también comprende una ciudad.

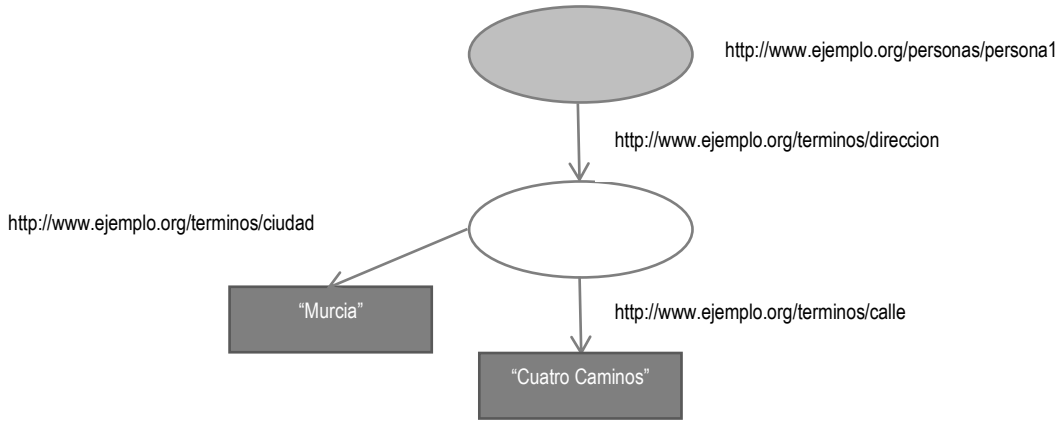


Figura 2.8. Ejemplo de grafo *RDF*.

Si bien el grafo mostrado en la Figura 2.8 se basa en la representación gráfica mostrada en la Figura 6, en este grafo se utiliza una notación particular para diferenciar claramente entre los tipos de nodos que representan a los sujetos y objetos. En particular, los *IRIs* se muestran como óvalos rellenos y los nodos en blanco como nodos vacíos, mientras que los literales se muestran como rectángulos. Es importante mencionar que en esta figura los literales se muestran con una vista simplificada que no incluye los tipos de datos, solo las formas léxicas.

2.7.1.3. Resource Description Framework Schema

Resource Description Framework Schema (*RDFS*) es un vocabulario de modelado de datos para datos *RDF*. Es, de hecho, una extensión al vocabulario básico definido por *RDF* y consiste en una colección de recursos *RDF* que se puede usar para describir otros recursos *RDF* en vocabularios de aplicación específica.

En términos generales, *RDFS* provee mecanismos para describir grupos de recursos relacionados y las relaciones entre dichos recursos: *clases* y *propiedades*. El sistema de clases y propiedades de *RDFS* es similar a los sistemas de lenguajes de programación orientada a objetos como Java. No obstante, en *RDFS* las propiedades se describen en función de los tipos de recursos a los que aplican, esto es, mediante los mecanismos *dominio* y *rango*, mientras que en dichos sistemas las clases se definen en términos de las características que pueden tener sus instancias. Uno de los beneficios de este enfoque centrado en propiedades es que permite a cualquier persona extender descripciones de recursos existentes, uno de los principios arquitectónicos de la Web semántica.

2.7.1.3.1. Clases, instancias y propiedades

Como se mencionó anteriormente, en *RDFS* una clase es un grupo de recursos *RDF* relacionados y, a su vez, una clase es un recurso *RDF*. Por tanto, cada clase está asociada a un conjunto de individuos denominado *extensión* de la clase, y cada individuo en dicho conjunto es conocido como una *instancia* de la clase. El grupo de todos los recursos que son clases *RDFS* forma a su vez una clase denominada rdfs:Class, por lo que toda clase *RDFS*, incluida ella misma, es una instancia de la clase rdfs:Class. Formalmente, la construcción rdfs:Class permite declarar recursos que son clases *RDFS*.

Por otro lado, todo recurso *RDF* es una instancia de la clase rdfs:Resource, la cual es, a su vez, una instancia de la clase rdfs:Class. De hecho, la clase rdfs:Resource representa la clase de todo, por lo que todas las clases *RDFS* son subclases de ella.

Otras clases definidas por el vocabulario *RDFS* son las siguientes:

- rdfs:Literal: es la clase de las representaciones léxicas de los valores, es decir los literales.
- rdfs:Datatype: es la clase de los valores propiamente dichos. Es subclase de la clase rdfs:class (además de su instancia), así como de la clase rdfs:Literal.
- rdf:Property: es la clase de las propiedades *RDFS*. El concepto de propiedad es definido por el modelo de datos de *RDF* como una relación binaria entre recursos denotados por sujetos y objetos.

En *RDFS* se define además el concepto de sub-propiedad para representar relaciones en las que todos los pares de recursos involucrados están asociados mediante otra relación. A diferencia de lo que ocurre con las clases, *RDFS* no define una súper-propiedad raíz. Formalmente, las subpropiedades pueden declararse utilizando la construcción rdfs:subPropertyOf.

Otras propiedades definidas por el vocabulario *RDFS* son las siguientes:

- rdfs:range: es una instancia de la clase rdfs:Property que permite declarar que los valores de una propiedad son instancias de una o más clases.
- rdfs:domain: es una instancia de la clase rdfs:Property que permite declarar que cualquier recurso al que se aplica una propiedad dada es una instancia de una o más clases. Esta propiedad, al igual que la propiedad rdfs:range, aplica solo a propiedades; de modo que el dominio de ambas, la propiedad rdfs:range y la propiedad rdfs:domain, es la clase rdf:Property.
- rdfs:type: es también una instancia de la clase rdfs:Propety. Esta propiedad permite declarar que un recurso es una instancia de una clase.
- rdfs:subClassOf: se trata igualmente de una instancia de la clase rdfs:Property que permite declarar que todas las instancias de una clase son también instancias de otra, esto es, que una clase es subclase de otra.

2.7.1.4. OWL Web Ontology Language

OWL Web Ontology Language (*OWL*) es el lenguaje para la representación formal del significado y las relaciones explicitas entre los términos en los vocabularios usados por los documentos dentro de la Web Semántica (Bechhofer et al., 2004). Dichas representaciones semánticas formales y explícitas son denominadas ontologías y son necesarias dentro de la visión de la Web en la que la información contenida en los documentos debe ser no solo presentada a los humanos sino también procesada e integrada automáticamente por las aplicaciones, la Web Semántica.

OWL añade vocabulario a la semántica básica definida por *RDFS* para la descripción de propiedades y clases de recursos *RDF*. Dicho vocabulario es el que finalmente permite a las aplicaciones realizar los razonamientos útiles que se espera que realicen dentro de la mencionada visión de la Web semántica.

Formalmente, *OWL*, en su versión 1, comprende tres sub-lenguajes de creciente expresividad: OWL-Lite, OWL-DL y OWL-Full. Entretanto, *OWL* 2 comprende cinco sub-lenguajes: OWL 2-EL, OWL 2-QL, OWL 2-RL, OWL 2-DL y OWL 2-Full, sin ninguna relación de inclusión entre ellos. A continuación, se describe brevemente cada uno de estos sub-lenguajes.

- OWL-Lite: Provee un subconjunto mínimo de características del lenguaje *OWL* que permite, básicamente, la creación de jerarquías de subclases, a saber, clases y restricciones de propiedades. De hecho, OWL-Lite permite crear restricciones simples de cardinalidad de cualquier tipo, así como restricciones simples de valores de algunos tipos.
- OWL-DL: Incluye realmente todas las construcciones de *OWL*, pero impone ciertas restricciones en su uso, así como en el uso del vocabulario de *RDFS*. Dichas restricciones están destinadas a proveer soporte al razonamiento basado en lógicas de descripción (*Description Logics*, *DL*), la familia de lenguajes de representación de conocimiento que constituye el fundamento formal de *OWL*.
- OWL-Full: No es formalmente un sub-lenguaje de *OWL* ya que incluye todas las construcciones de *OWL* y permite el uso libre de dichas construcciones en conjunto con las construcciones de *RDFS*. En otras palabras, OWL-Full permite la máxima expresividad, así como la libertad sintáctica de *RDF*, lo cual no garantiza la completitud y decidibilidad de los cálculos, violando las restricciones de los razonadores para lógicas de descripción.
- OWL 2-EL: Se basa en la familia de lógicas descriptivas EL++, está destinado al uso en ontologías con grandes cantidades de clases y propiedades; corresponde al subconjunto de OWL 2-DL para el que los problemas básicos de razonamiento se pueden llevar a cabo en tiempo polinomial.
- OWL 2-QL: Se basa en la familia de lógicas descriptivas DL-Lite, y está orientada a aplicaciones que utilizan grandes volúmenes de datos y en las que es la solución de consultas es una tarea primordial.
- OWL 2-RL: Se basa en la lógica descriptiva *Description Logic Programs* (*DLP*), destinado a aplicaciones que requieren razonamiento escalable sin el sacrificio de las capacidades de expresión, puede interpretarse con razonadores basados en reglas.
- OWL 2-DL: Se basa en la semántica de modelo teórico de la lógica descriptiva SROIQ, y está orientado a habilitar un alto grado de expresividad en las ontologías; este sub-lenguaje relaja ciertas restricciones impuestas sobre OWL-DL, de modo que da lugar a un espectro más amplio de sub-lenguajes.
- OWL 2-Full: Es (respecto a OWL 2-DL) la otra visión semántica de OWL en OWL 2; se puede interpretar como un superconjunto no restringido (y por tanto indecidible) de OWL 2-DL. Partiendo del hecho de que en OWL 2 existen dos maneras distintas de asignar significado a las ontologías: mediante semántica directa y mediante semántica basada en *RDFS*, cabe mencionar que las ontologías en OWL 2-Full se pueden interpretar solo bajo la semántica basada en *RDFS*.

Cada uno de los sub-lenguajes correspondientes a OWL 1 puede verse también como una extensión de su predecesor; de modo que una ontología puede expresarse legalmente y dar lugar a conclusiones válidas en un sub-lenguaje más extenso de OWL 1. En el caso de OWL 2, los sub-lenguajes propiamente dichos (OWL 2-EL, OWL 2-QL y OWL 2-RL) no son subconjuntos unos de otros. Para un mejor entendimiento de estas relaciones, ver la Figura 2.9, la cual aparece originalmente en (Hoekstra, 2009).

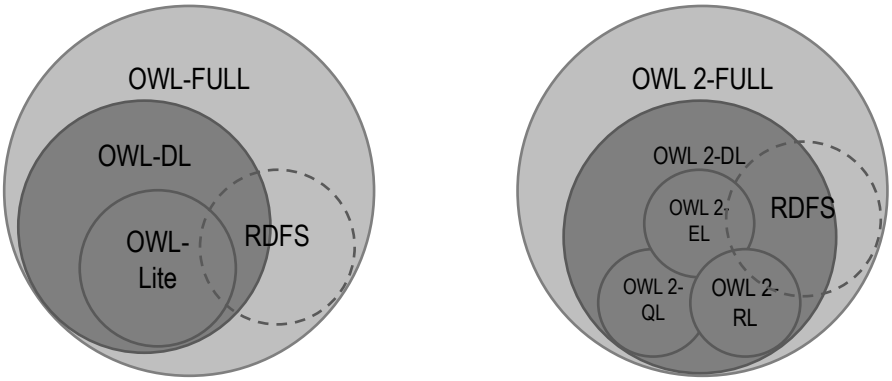


Figura 2.9. Sub-lenguajes de OWL 1 y OWL 2.

En este contexto, es importante mencionar que, considerando que *OWL* es una extensión al vocabulario de *RDF*, cualquier grafo *RDF* es una ontología válida en OWL-Full. Además, el significado añadido por *OWL* a cualquier grafo *RDF* incluye el significado dado por *RDF* al mismo. Por tanto, las ontologías en OWL-Full pueden incluir arbitrariamente contenido en *RDF*.

Uno de los conceptos fundamentales en *OWL* es el de la "suposición de un mundo abierto", por el cual las representaciones de los recursos en la visión de la Web semántica no están limitadas a un ámbito. Esto implica que pueda añadirse nueva información, esto es hechos y conclusiones lógicas, a descripciones existentes; no obstante, dicha información añadida no puede en ningún caso revocar información previa (monotonía).

2.7.1.4.1. Ontologías

En términos generales, una ontología comprende usualmente un Componente Terminológico (del inglés T-Box). No obstante, en OWL una ontología se compone de un Componente Terminológico y un Componente de Afirmaciones (del inglés A-Box); en donde el primer componente representa al conjunto de axiomas de esquema, mientras que el segundo componente representa al conjunto de axiomas de datos (hechos concretos) (Horrocks, Glimm, & Sattler, 2007).

La mayoría de los elementos de una ontología en *OWL* se puede reducir a los siguientes elementos básicos: *clases, instancias y propiedades.*

Una *clase* es una abstracción de un grupo de recursos con características similares. Al igual que en *RDFS*, en *OWL* cada clase está asociada a un conjunto de individuos denominado *extensión* de la clase, y cada individuo en dicho conjunto es conocido como una *instancia* de la clase. Además, una clase *OWL* tiene un significado intencionado que representa un concepto.

Una *propiedad* en *OWL*, al igual que en *RDFS*, es una relación binaria dirigida que permite expresar hechos generales acerca de las extensiones de las clases y hechos particulares sobre las instancias de las mismas. No obstante, *OWL* distingue entre dos tipos de propiedades:

- Propiedades de tipo de dato: son relaciones entre instancias de clases y valores de datos, esto es literales *RDF* y tipos de dato de *XML Schema*.
- Propiedades de objeto: son relaciones entre instancias de clases.

En cuanto a su sintaxis, una ontología en *OWL* es un grafo *RDF*, el cual a su vez es un conjunto de tripletas *RDF*. Por lo tanto, como cualquier grafo *RDF*, la sintaxis de una ontología puede tomar distintas formas de acuerdo a la especificación RDF/XML.

Además, las ontologías se almacenan en la Web en forma de documentos; dichos documentos consisten generalmente en: (1) un encabezado y (2) un número arbitrario de construcciones complejas denominadas *axiomas*, incluyendo *axiomas de clase* y *axiomas de propiedad*, así como un número arbitrario de hechos acerca de instancias de clases. Estos elementos son descritos brevemente a continuación.

2.7.1.4.1.1. Axiomas de clase y axiomas de propiedad

En *OWL*, las clases se definen utilizando *descripciones* de clase, las cuales son una suerte de bloques de construcción básicos que pueden ser combinados para dar lugar a construcciones más complejas denominadas *axiomas* de clases. Así, una *descripción* de clase describe una clase *OWL* ya sea mediante un nombre de clase (mediante un *IRI*) o mediante la especificación de la *extensión* de clase de una *clase anónima*; en el último caso puede ser: (1) una *enumeración* exhaustiva de los individuos que forman el conjunto de instancias de la clase, (2) una *restricción de propiedad* o (3) la *intersección*, *unión* o *complemento* de una o más descripciones de clase.

En particular, las clases nombradas se representan sintácticamente como instancias nombradas de la clase owl:Class, la cual es una subclase de la clase rdfs:Class; específicamente, utilizando el atributo rdf:ID en la construcción owl:Class, o bien usando el atributo rdf:about en dicha construcción. En detalle, este último atributo permite extender la definición de un recurso sin modificar el documento original; lo que es fundamental en la construcción incremental de ontologías distribuidas dentro de la visión de la Web Semántica. En este punto, cabe mencionar que, toda instancia en *OWL* es miembro de la *extensión de clase* de una clase denominada owl:Thing; como consecuencia, toda clase *OWL* es implícitamente subclase de dicha clase. Se puede pensar en la clase owl:Thing como una clase homologa de la clase rdfs:Resource de *RDFS*. De manera similar, toda clase *OWL* tiene como subclase a una clase denominada owl:Nothing, cuya *extensión de clase* está formada por un conjunto vacío, como su nombre lo indica.

En lo que respecta a las descripciones de clase basadas en restricciones de propiedad, vale la pena hacer mención de que *OWL* distingue dos tipos de *restricciones de propiedad*: *restricción de valor* y *restricción de cardinalidad*. Una *restricción de valor* impone una limitación al rango de valores de una propiedad cuando se aplica a una *descripción de clase* particular. Una *restricción de cardinalidad* impone una limitación al número de valores de una propiedad cuando se aplica a una descripción de clase concreta. Ambos tipos de restricción se pueden aplicar tanto a las propiedades de tipo de dato como a las propiedades de objeto, esto mediante la construcción owl:Restriction. En detalle, owl:Restriction es una clase por sí misma, y una subclase de la clase rdfs:Class; por lo tanto, las restricciones en *OWL* se representan sintácticamente como instancias nombradas de la clase owl:Restriction. A continuación, se describen brevemente las distintas propiedades *OWL* que permiten representar restricciones de valor y cardinalidad.

- owl:allValuesFrom y owl:someValuesFrom: permiten describir una clase de todas las instancias para las que todos (o al menos uno de) los valores de la propiedad a la que aplica son, o bien miembros de la *extensión de clase* de la *descripción de clase*, o bien valores de datos dentro del rango de datos especificado.
- owl:hasValue: permite describir una clase de todas las instancias para las que la propiedad a la que aplica tiene al menos un valor semánticamente similar al valor especificado.
- owl:maxCardinality y owl:minCardinality: permiten describir la clase de todos los individuos que tiene como mucho (o como mínimo) *n* valores semánticamente distintos para la propiedad a la que aplica, en donde *n* es el valor de la restricción de cardinalidad
- owl:cardinality: permite describir una clase de todos los individuos que tiene exactamente *n* valores semánticamente distintos para la propiedad a la que aplica.

- owl:maxQualifiedCardinality y owl:minQualifiedCardinality: permiten describir la clase de todos los individuos que tiene como mucho (o como mínimo) *n* valores semánticamente distintos de una clase o rango de datos específico para la propiedad a la que aplica.
- owl:qualifiedCardinality: permite describir una clase de todos los individuos que tiene exactamente *n* valores semánticamente distintos de una clase o rango datos específico para la propiedad a la que aplica.

Una *descripción de clase* del primer tipo representa por si sola el tipo más básico *de axioma de clase*. Los *axiomas de clase* típicamente contienen elementos adicionales que indican características necesarias y/o suficientes de las clases. *OWL* provee tres construcciones para combinar *descripciones de clase* en *axiomas de clase*: rdfs:subClassOf, owl:equivalentClass, owl:disjointWith y owl:disjointUnionOf. La propiedad owl:equivalentClass permite asociar una *descripción de clase* a otra *descripción de clase*; el significado del *axioma de clase* resultante es que las dos *descripciones de clase* involucradas tienen la misma *extensión de clase*. La propiedad owl:disjointWith, al igual que la propiedad anterior, permite asociar una *descripción de clase* a otra *descripción de clase*, pero en este caso, el significado del axioma resultante es que las extensiones de clase involucradas no tienen ninguna instancia en común. Por último, la propiedad owl:disjointUnionOf representa un axioma que establece que una *descripción de clase* es una unión disjunta de otras *descripciones de clase*, las cuales son disjuntas dos a dos.

Entrando en la materia de los axiomas de propiedad, es importante mencionar que estos permiten definir características de una propiedad; no obstante, la forma más simple de un axioma de propiedad es la definición de la mera existencia de una propiedad. *OWL* incluye cuatro tipos de construcciones para definir *axiomas de propiedad*: (1) construcciones de *RDFS*, a saber, las propiedades rdfs:subPropertyOf, rdfs:range y rdfs:domain, (2) relaciones entre propiedades, el cual incluye las propiedades owl:equiavalentProperty, owl:inverseOf y owl:propertyDisjointWith, (3) limitaciones de cardinalidad global, a saber, las propiedades owl:FunctionalProperty y owl:InverseFunctionalProperty y (4) características de propiedades lógicas, el cual incluye las propiedades owl:SymmetricProperty, owl:AsymmetricProperty, owl:ReflexiveProperty, owl:IrreflexiveProperty y owl:TransitiveProperty.

2.7.1.4.1.2. Axiomas de individuo

En *OWL*, los *axiomas de individuo* proporcionan un mecanismo para definir hechos acerca de las instancias que forman parte de las *extensiones* de las clases en una ontología. En este sentido, es posible distinguir entre dos formas distintas de hechos sobre instancias: (1) hechos de pertenencia a clases y valores de propiedad y (2) hechos de identidad. En lo que respecta a los hechos de pertenencia a clases, este corresponde al tipo más básico de *axioma de individuo*, por el que se establece explícitamente la clase (formalmente la *extensión de clase*) a la que pertenece determinado individuo (utilizando las propiedades rdf:ID y rdf:about), así como valores para propiedades cuyo dominio es la clase correspondiente; por último, mencionar que el individuo objetivo puede ser un *individuo nombrado* o un *individuo anónimo*.

En cuanto a los hechos de identidad de individuos, dado que en *OWL* no se asume que una entidad posee un único nombre, sino que es posible que nombres distintos hagan referencia a una misma entidad, ninguno de los dos casos se da por cierto; por lo tanto, a fin de establecer la identidad de un individuo particular, es necesario definir *un axioma de individuo* a partir de alguna de las propiedades (o clases) *OWL* que se describen a continuación:

- owl:sameAs: asocia un par de individuos; permite determinar que dos *IRIs* diferentes de hecho hacen referencia a una misma entidad.
- owl:differentFrom: asocia un par de individuos; permite determinar que dos *IRIs* distintos refieren a entidades diferentes.

- owl:AllDifferent: es una clase que permite indicar que todos los individuos de una lista asociada a ella mediante la propiedad owl:distinctMembers son distintos uno de otro. Representa un atajo para los casos en los que es válida la suposición de un único nombre para las entidades.

2.7.2. Lenguajes de Consulta: SPARQL Protocol and RDF Query Language

En el contexto de la Web Semántica, por consulta se entiende el conjunto de tecnologías, esto es, lenguajes y protocolos, que permiten la recuperación programática de información de la "Web de Datos", donde la Web de Datos es la materialización de la visión de la Web Semántica. En este sentido, *RDF* provee los fundamentos para la publicación y enlazamiento de los datos en dicha Web de Datos. No obstante, al igual que las bases de datos relacionales requieren lenguajes de consulta específicos, la Web de Datos, representada típicamente utilizando *RDF* como formato de datos, necesita un lenguaje de consulta específico para este formato. *SPARQL Protocol and RDF Query Language* (*SPARQL*) cubre esta necesidad proveyendo no solo un lenguaje de consulta sino también una serie de protocolos que, en su conjunto, permiten el envío de consultas y la recuperación de los correspondientes resultados a través de otros protocolos como *HyperText Transfer Protocol* (*HTTP*) o *Simple Object Access Protocol* (*SOAP*).

Las consultas en *RDF* comúnmente se componen a partir de un conjunto básico de patrones de tripletas denominado "patrón básico de grafo "; un "patrón de tripleta" es similar a una tripleta *RDF*, excepto que, en el primero, tanto el sujeto como el predicado y el objeto pueden ser variables, esto es, elementos de la sintaxis abstracta propia de *SPARQL*. En general, un patrón básico de grafo se corresponde con un sub-grafo de los datos *RDF* consultados cuando los términos *RDF* del sub-grafo se pueden intercambiar con las variables del patrón de grafo y el resultado es un grafo *RDF* equivalente a dicho sub-grafo. No obstante, un patrón básico de grafo puede corresponderse con los datos *RDF* en múltiples formas, esto es, con múltiples sub-grafos, el conjunto de los cuales forma lo que se denomina "secuencia de soluciones". Además, es posible combinar más de un patrón básico de grafo como alternativa en una misma consulta de modo que cero o varios de esos patrones se pueden corresponder con el grafo *RDF* de consulta; para ello *SPARQL* provee la palabra clave UNION. Es importante mencionar que, por términos *RDF* se entienden *IRIs*, literales y nodos en blanco de *RDF*.

En detalle, *SPARQL* provee distintos tipos de formas de consulta, en concreto, SELECT, CONSTRUCT, ASK y DESCRIBE. La forma SELECT devuelve directamente las variables y sus asignaciones. Las consultas basadas en la forma CONSTRUCT, por su parte, devuelven un grafo *RDF* especificado por una plantilla de grafo, el cual se forma a partir de las distintas soluciones en la secuencia de soluciones; una plantilla de grafo es similar a un patrón de grafo, excepto porque el primero puede no contener variables, sino simplemente tripletas explícitas. La forma ASK permite probar si un patrón de grafo tiene solución; como respuesta solo provee un valor *booleano*. Por último, las consultas basadas en la forma DESCRIBE devuelven un grafo *RDF* que se compone de información acerca de los recursos consultados; esta información es determinada por el procesador de consultas *SPARQL*, y no prescrita por la consulta.

En este contexto, cabe mencionar que en *SPARQL* es posible asignar el valor de una expresión a un término *RDF* en una solución o secuencia de soluciones mediante el enlace de una nueva variable a dicho valor. Esto se consigue, entre otras maneras, con la utilización de la forma BIND.

La Figura 2.10 muestra un ejemplo de consulta básica en *SPARQL* tomado del documento de especificación de la sintaxis y semántica del lenguaje de consulta de *SPARQL* 1.1 producido por el "Grupo de Trabajo de SPARQL" del *W3C* (Harris, Seaborne, & Prud'hommeaux, 2013).

Datos:

<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .

Consulta:

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

Resultado:

title
"SPARQL Tutorial"

.

Figura 2.10. Ejemplo de consulta *SPARQL* para recuperar el título de un libro desde un grafo dado.

La consulta mostrada en la Figura 2.10 permite recuperar el título de un libro desde un grafo *RDF* dado (consulta basada en la cláusula SELECT); para ello define un patrón básico de grafo que comprende un único patrón de tripleta con una sola variable (?title) en la posición del objeto. En dicha figura se puede observar la sintaxis básica de las consultas en SPARQL, en la cual, la cláusula WHERE permite proveer el patrón de grafo a comparar contra el grafo *RDF* de consulta.

Como en el caso del lenguaje de consulta SQL, *SPARQL* provee un mecanismo que permite aplicar expresiones a grupos de soluciones; para ello requiere que los conjuntos de soluciones, los cuales por defecto constan de un solo grupo que contiene todas las soluciones, sean agrupados empleando la cláusula GROUP BY. En concreto, *SPARQL* 1.1 define siete funciones de agregación representadas por las cláusulas COUNT, SUM, MIN. MAX, AVG, GROUP_CONCAT y SAMPLE. Si bien las seis primeras cláusulas corresponden en funcionalidad a las cláusulas del mismo nombre en SQL, la cláusula SAMPLE devuelve un valor arbitrario de un *multiset* pasado como parámetro.

Asimismo, *SPARQL* 1.1 provee un mecanismo que permite incrustar consultas dentro de otras consultas con el objetivo de conseguir resultados que de otra manera no podrían conseguirse: subconsultas.

En la Figura 2.11 se muestra un ejemplo de una consulta basada en la forma de consulta CONSTRUCT; este ejemplo está contenido en el documento de especificación de la sintaxis y semántica del lenguaje de consulta de *SPARQL* 1.1.

Datos:

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:a    foaf:name   "Alice" .
_:a    foaf:mbox   <mailto:alice@example.org> .
```

Consulta:

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT   { <http://example.org/person#Alice> vcard:FN ?name }
WHERE       { ?x foaf:name ?name }
```

Resultado:

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .

<http://example.org/person#Alice> vcard:FN "Alice" .
```

Figura 2.11. Ejemplo de consulta *SPARQL* basada en la forma CONSTRUCT.

La consulta mostrada en la Figura 2.11 permite construir un nuevo grafo *RDF* para representar una tarjeta de negocios electrónica (formato "vCard") a partir de la información basada en el vocabulario (*Friend Of A Friend*, *FOAF*[3]) provista (un primer grafo *RDF*). En dicha figura se puede observar el uso de la palabra clave PREFIX para asociar "etiquetas de prefijo" a *IRIs* de modo que en la consulta se hace uso de "nombres con prefijo" en lugar de *IRIs completos.* Un nombre con prefijo se representa con la sintaxis: "*etiqueta de prefijo* : *parte local*"; en este caso, las partes locales hacen referencia a los nombres de las propiedades "FN" y "name" de los vocabularios *RDF* "vCard"[4] y "FOAF", respectivamente. Obsérvese que este ejemplo utiliza una versión obsoleta del vocabulario vCard.

2.7.2.1. SPARQL Update

Como un lenguaje acompañante para el lenguaje de consulta para *RDF* de *SPARQL*, el "Grupo de Trabajo de SPARQL" del *W3C* ha propuesto el lenguaje de actualización para grafos *RDF* en almacenes de grafos, *SPARQL Update* (Gearon, Passant, & Polleres, 2013). *SPARQL Update* utiliza una sintaxis derivada de la sintaxis del lenguaje de consulta para *RDF* de *SPARQL* (formalmente, *SPARQL Query Language*), y soporta dos tipos distintos de operaciones de actualización sobre almacenes de grafos *RDF*, a saber, operaciones de actualización propiamente dicha y operaciones de gestión.

En este contexto, el término operación de actualización hace referencia a una acción (expresada como un solo comando) que resulta en la modificación de los datos en un almacén de grafos; a una secuencia de cero o más operaciones de actualización que se envía a un almacén de grafos se le conoce como solicitud. El término almacén de grafos, por su parte, hace referencia a un contenedor mutable de grafos *RDF* gestionado por un solo servicio de actualización; dicho servicio es informalmente denominado *endpoint SPARQL*. De manera similar a un *dataset RDF* (en el contexto del lenguaje de consulta para *RDF* de *SPARQL*), un almacén de grafos,

[3] http://xmlns.com/foaf/spec/
[4] https://www.w3.org/TR/vcard-rdf/

contiene un espacio sin nombre que aloja a un grafo por defecto y cero o más espacios nombrados que albergan a un número arbitrario de grafos nombrados.

Entrando en materia, en el grupo de las operaciones de actualización propiamente dicha se encuentran aquellas operaciones que permiten la modificación de los grafos *RDF* en un almacén de grafos, pero que, como parte de la modificación, no crean ni eliminan dichos grafos. En detalle, la especificación de *SPARQL Update* provee cinco operaciones de actualización principales: INSERT DATA, DELETE DATA, DELETE/INSERTE, LOAD y CLEAR.

La operación INSERT DATA permite añadir una o más tripletas dadas a un grafo *RDF* específico en el almacén de grafos, de modo que si se indica un grafo de destino que no existe en el almacén, la implementación de dicha operación debe crear automáticamente el grafo indicado. Por su parte, la operación DELETE DATA, permite remover, de un grafo específico en un almacén de grafos, una o más tripletas dadas, siempre que el grafo especificado contenga dichas tripletas.

En lo que respecta a la operación DELETE/INSERT, esta se puede usar para remover y añadir tripletas de/a un grafo en el almacén de grafos, a partir de las asignaciones de las variables de un patrón básico de grafo especificado mediante la cláusula WHERE. Como se puede deducir, tanto la cláusula DELETE como la cláusula INSERT deben indicar una plantilla de grafo como en el caso de la cláusula CONSTRUCT (ver subsección 2.7.2 de esta sección). En detalle, una vez que la evaluación del patrón básico de grafo resulta en una secuencia de soluciones, se sustituyen las asignaciones de cada una de las soluciones en la plantilla indicada en la cláusula DELETE, y posteriormente en la plantilla indicada en la cláusula INSERT, a fin de remover e insertar las correspondientes tripletas, respectivamente. Cabe mencionar que, como en el caso de la operación INSERT DATA, cuando se indica un grafo de destino que no existe en el almacén, la creación automática del grafo indicado corre a cargo de la implementación correspondiente.

Opcionalmente, en una misma operación DELETE/INSERT, bien la cláusula INSERT, bien la cláusula DELETE, puede estar vacía. En el primer caso, la operación consistiría únicamente en una operación de eliminación; mientras que en el segundo caso sería exclusivamente una operación de inserción. Asimismo, en el primer caso, la especificación de *SPARQL Update* permite el atajo DELETE WHERE (formalmente una operación más). Dicho atajo permite definir las tripletas a remover a partir de las asignaciones de las variables de un patrón básico de grafo especificado mediante la cláusula WHERE, de modo que el patrón de consulta sirve además como plantilla para la operación de eliminación.

La operación LOAD permite, simplemente, leer un documento *RDF* desde un *IRI*, e insertar en un grafo dado las tripletas contenidas en él, de modo que si se indica un grafo de destino que no existe en el almacén de grafos, la implementación de dicha operación debe crear automáticamente el grafo indicado. Por último, la operación CLEAR se puede utilizar para remover tripletas de un almacén, específicamente de un grafo dado, del grafo por defecto, de todos los grafos nombrados, o de todos los grafos sin excepción.

La Figura 2.12 muestra un ejemplo de operación de actualización basada en el comando INSERT DATA; este ejemplo está contenido en el documento de especificación de la sintaxis y semántica del lenguaje *SPARQL* 1.1 *Update*.

Operación de actualización:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
INSERT DATA
{ GRAPH <http://example/bookStore> { <http://example/book1>  ns:price  42 } }
```

Grafo de destino:

```
# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://example/book1> dc:title "Fundamentals of Compiler Design" .
```

Grafo actualizado:

```
# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .
<http://example/book1> dc:title "Fundamentals of Compiler Design" .
<http://example/book1> ns:price 42 .
```

Figura 2.12. Ejemplo de operación de actualización basada en el comando INSERT DATA de *SPARQL Update*.

En la consulta mostrada en la figura anterior es posible observar la sintaxis básica de las operaciones de actualización en *SPARQL Update*. Cabe resaltar el uso de la palabra clave GRAPH para indicar el grafo de destino de la operación; en el caso específico de las operaciones basadas en el comando DELETE/INSERT, alternativamente se puede emplear la cláusula WITH, la cual, a diferencia de la palabra clave GRAPH, debe preceder cualquier comando DELETE e INSERT cuando estos no indican explícitamente un grafo de destino.

El grupo de comandos de *SPARQL Update* que permite llevar a cabo operaciones de gestión sobre almacenes de grafos no se aborda en este documento de tesis por encontrarse fuera del alcance de la investigación.

2.7.3. Lenguajes de Reglas

Partiendo del hecho de que en la Web Semántica los datos se modelan como un conjunto de relaciones nombradas o explícitas entre recursos, la posibilidad de descubrir, mediante procedimientos automáticos, nuevas relaciones (relaciones implícitas) a partir de los datos y de información adicional en forma de vocabularios representa una de las líneas de investigación actuales en Web Semántica. En efecto, se puede considerar a los lenguajes de reglas, junto con los vocabularios y los lenguajes de consulta, los bloques de construcción fundamentales de la Web de Datos.

Si bien la fuente de esa información extra puede ser, o bien una ontología, o bien un conjunto de reglas, ambos enfoques hacen uso de técnicas de representación de conocimiento. En el caso de los vocabularios el enfoque está más bien relacionado con métodos de clasificación, poniendo énfasis en la definición de clases y subclases, en como los recursos individuales se relacionan con dichas clases, así como en la caracterización de las relaciones entre las clases y sus individuos. Por su parte, el enfoque basado en conjuntos de reglas deriva del paradigma de programación lógica (el lenguaje Prolog, por ejemplo), y está destinado directamente a la definición de un mecanismo general para la generación de nuevas relaciones a partir de relaciones nombradas existentes. En ocasiones, sin embargo, es necesario hacer uso de ambos enfoques con el mismo *dataset* y, de

hecho, debe ser posible; esto debido a que, en teoría, bajo ciertas condiciones estos enfoques pueden llegar a ser complementarios.

En este sentido, es posible decir que *OWL* está optimizado para problemas de razonamiento taxonómico en especificaciones ontológicas (fuera de los datos), mientras que los lenguajes de programación lógica están optimizados precisamente para problemas de razonamiento sobre los datos (fuera de las especificaciones ontológicas). De ahí el esfuerzo del *W3C*, específicamente del "Grupo de Trabajo en el Formato de Intercambio de Reglas (RIF)", en la definición precisa de un núcleo básico de lenguaje de reglas que permita generalizar las distintos paradigmas existentes entre los sistemas basados en reglas (lógica de primer grado, programación lógica, reglas de producción y reglas reactivas o reglas evento-condición-acción) y facilitar el intercambio de las especificaciones resultantes entre los mismos y, por último, pero no menos importante, que defina la relación precisa con el lenguajes *OWL* y su integración con tripletas *RDF*.

En el contexto de la Web Semántica, el concepto de regla de hecho hace referencia a elementos de sistemas basados en reglas ligados a datos semánticos. En este sentido, si bien no es posible definir un tipo de regla que sea capaz de capturar todas las variantes representadas por los sistemas basados en reglas existentes, sí es posible definir, a partir del concepto de cláusula "Horno", un tipo restringido de regla que puedan entender todos estos sistemas; formalmente, una regla "Horno" es una implicación de un antecedente (un conjunto de fórmulas atómicas) a un consecuente (una sola fórmula atómica), donde todas las variables del consecuente deben ocurrir en al menos un átomo del antecedente, y todas se consideran universalmente cuantificadas.

Partiendo de esta premisa, el Grupo de Trabajo en el Formato de Intercambio de Reglas (RIF) del *W3C* ha publicado a la fecha distintas recomendaciones relacionadas con en el desarrollo del formato de intercambio de reglas, *RIF* (del inglés *Rule Interchange Format*), específicamente con el desarrollo de una familia de lenguajes llamados dilectos con rigurosa especificación de sintaxis y semántica: la familia de dialectos de *RIF*. Esta familia de dialectos pretende ser lo suficientemente uniforme como para compartir hasta donde sea posible la sintaxis y semántica del aparato actual de los lenguajes de reglas. Por otro lado, *RIF* ha sido diseñado de modo que sea posible definir la sintaxis de nuevos dialectos a partir de la extensión de los dialectos existentes, añadiendo nuevas características correspondientes a la funcionalidad añadida requerida.

Entretanto, distintas propuestas de extensión de *OWL* con características de lenguajes de reglas han emergido de entre los cuerpos de investigadores en Web Semántica, en un esfuerzo por dotar a dicho lenguaje de una mayor expresividad y permitir la integración directa entre las ontologías y las bases de reglas. En este contexto, dado que los objetivos de esta tesis doctoral no se alinean con los objetivos de la iniciativa del *W3C* referente al desarrollo del lenguaje *RIF*, es el enfoque antes mencionado el que resulta relevante para esta investigación, y del que a continuación se destacan algunos trabajos significativos.

Basado en la combinación de los sub-lenguajes OWL-DL y OWL-Lite de *OWL* con el sub-lenguaje Data log del sistema unificador de familias de lenguajes de reglas Web, *Rule Markup Language (Rulen)*, el lenguaje *Semántica Web Rule Language* (*SWRL*) (Horrocks & Patol-Schneider, 2004) representa una de las propuestas más populares, si no es que la más popular, a este respecto. En detalle, esta propuesta extiende el conjunto de axiomas de la sintaxis abstracta de alto nivel de los sub-lenguajes de *OWL* antes mencionados para soportar reglas Horno; asimismo, plantea una extensión a la semántica del modelo teórico de *OWL* para proveer un significado formal para las ontologías que incluyen reglas basadas en la sintaxis abstracta extendida. La propuesta de hecho se sometió a consideración del *W3C* por el consejo nacional de investigación de Canadá, la *Stuart-up* Network Inferece y la Universidad de Stanford en el año 2004 (Horrocks et al., 2004), y fue publicada como "W3C Member Submission" poco tiempo después ese mismo año; no obstante, esta propuesta no derivó en un estándar *W3C*. En este punto, cabe hacer mención de que muchas de las características de *SWRL* están

actualmente cubiertas por el dialecto *RIF Basic Logic Dialect (RIF-BLD)* de *RIF*, siendo este, evidentemente, una recomendación del *W3C* (un estándar).

Cabe mencionar que *Rulen* se concibió originalmente como un lenguaje de reglas para la Web Semántica (Boley, Tabet, & Wagner, 2001); con la visión de desarrollar *Rulen* como un sistema canónico de lenguajes para reglas Web, la organización sin ánimo de lucro detrás de *Rulen*, *Rulen*, Inc., ha convertido a este en una especificación general de reglas Web (Boley, Paschke, & Shafiq, 2010), considerándose actualmente el estándar de facto de reglas Web. En este contexto, vale la pena mencionar que *RIF* se ha fundamentado en cierta medida en *Rulen* y, al mismo tiempo, ciertas características de *RIF* han sido adoptadas por *Rulen*; lo que demuestra una clara convergencia entre estos esfuerzos sin lugar a dudas relacionados.

Con el objetivo de aprovechar la madurez y el soporte del que a la fecha gozaba *SPARQL* entre los motores de consulta y bases de datos *RDF*, para añadir, directamente sobre *RDF*, la capacidad de definir reglas y restricciones lógicas mediante consultas *SPARQL*, la notación *SPARQL Inferencing Notation* (*SPIN*) surgió como una especificación propia de la compañía TopQuadrant, compañía de integración de datos semánticos, en el año 2011. Ese mismo año, apoyada por TopQuadrant, en conjunto con la compañía de software OpenLink Software y el Instituto Politécnico Rensselaer, dicha especificación se sometió a consideración del *W3C*, y fue publicada por este como "W3C Member Submission" casi inmediatamente el mismo año (Knublauch, Hendler, & Idehen, 2011).

Así, por un lado, *SPIN* define una colección de propiedades y clases *RDF* que es posible usar para "ligar" clases *RDFS* y *OWL* a consultas *SPARQL* que capturan restricciones y reglas, en donde, dichas restricciones formalizan el comportamiento esperado de las clases ligadas. En este contexto, según palabras de los autores, la pila actual de lenguajes de modelado para la Web de Datos (*SKOS*, *RDFS* y *OWL*) provee excelentes mecanismos para capturar la estructura estática de los datos, pero tiene limitada capacidad para definir el comportamiento computacional general de los objetos que forman parte de las clases, de manera similar a como se hace en el paradigma de programación orientada a objetos. De ahí que, conceptualmente, *SPIN* combine características de lenguajes de programación orientada a objetos, además de elementos de lenguajes de consulta y sistemas basados en reglas.

Por otro lado, *SPIN* permite la representación de las consultas *SPARQL* empleando una notación procesable por ordenador alternativa a la notación textual original, a saber, una notación *RDF*. Con ello se hace posible almacenar consistentemente dichas consultas junto con los modelos de dominio; así también, se permite la compartición en la Web de las reglas junto con las definiciones de clases a las que están asociadas (la visión de la Web Semántica). Cabe resaltar que, de hecho, se trata de una representación basada en tripletas *RDF*, con lo que además se facilita el mantenimiento de aquellos modelos híbridos que combinan definiciones *RDF* y *OWL* con expresiones *SPARQL*.

Además de las ventajas que se han dejado entrever, el uso de *SPIN*, respecto al uso de *SWRL* o de otras propuestas basadas en *Rulen*, no requiere de motores de reglas intermediarios, con el añadido de la sobrecarga de comunicación que esto puede suponer, ya que las reglas se pueden ejecutar directamente sobre las bases de datos *RDF*; además al basarse enteramente en las tecnologías de la pila de tecnologías de la Web Semántica, no se precisa del desarrollo de habilidades y la adquisición de conocimientos en lenguajes de reglas propietarios. En este contexto, se puede considerar a *SPIN* como el estándar de facto de la industria para la representación de reglas y restricciones basadas en *SPARQL* en la Web de Datos. Teniendo en cuenta todo lo antes expuesto, en esta investigación se ha optado por la adopción de *SPIN* para la definición de reglas de inferencia de conocimiento no explícito a partir de conocimiento explícito en forma de tripletas *RDF*. Esto se explicará en detalle en el siguiente capítulo de este documento.

2.7.3.1. SPARQL Inferencing Notation

Además de las capacidades relacionadas con la representación de reglas y restricciones, *SPIN* provee capacidades de meta-modelado que permiten la definición de lenguajes de modelado propios, así como de extensiones al lenguaje de consulta *SPARQL*. En concreto, *SPIN* provee mecanismos para el encapsulamiento de consultas *SPARQL* reutilizables: plantillas *SPIN*; dichas plantillas son una suerte de consultas parametrizadas con variables pre-enlazadas. En este sentido, *SPIN* provee también un mecanismo para la definición de funciones *SPARQL* personalizadas, así como de funciones de propiedad denominadas "propiedades mágicas"; estas funciones se formalizan utilizando un vocabulario *RDF* que permite la descripción del cuerpo de las funciones y sus argumentos en *SPARQL*.

La Figura 2.13 muestra los componentes de la arquitectura de *SPIN*, en ella se pueden observar componentes relacionados con las capacidades antes mencionadas. Esta figura está basada en la imagen presentada en la página Web de *SPIN* en el sitio Web de la compañía TopQuadrant[5]. Dada la importancia de las reglas de inferencia en esta investigación, a continuación se describe el componente relacionado con las capacidades de representación de reglas y restricciones, esto es, el vocabulario *RDF* de descripción de clases, así como los elementos relevantes del vocabulario *RDF* para representación de consultas *SPARQL* como tripletas *RDF*, que se denomina formalmente "Sintaxis *RDF* para *SPARQL*". Para una descripción completa de dicha sintaxis, referirse al documento de especificación correspondiente (Knublauch, 2011b).

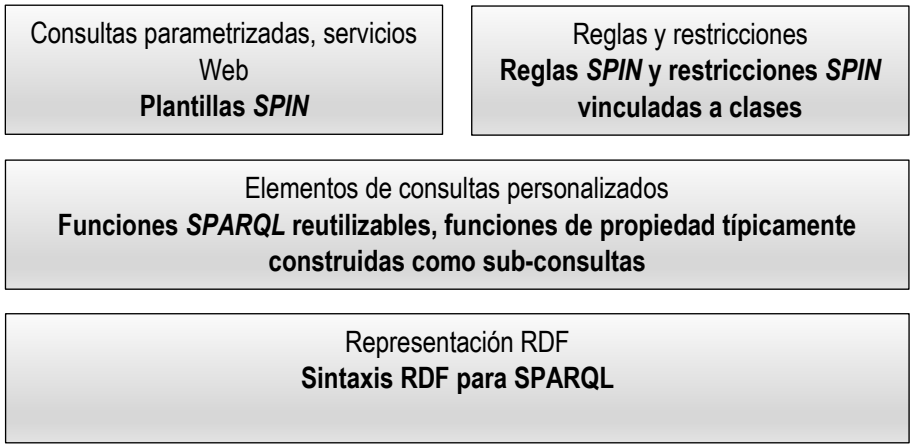


Figura 2.13. Arquitectura de la notación *SPIN*.

2.7.3.1.1. Vocabulario RDF de Descripción de Clases: Reglas y Restricciones SPIN

Como parte del vocabulario de descripción de clases, *SPIN* provee tres propiedades *RDF* distintas: "spin:rule", "spin:constraint" y "spin:constructor".

La propiedad spin:rule permite vincular consultas *SPARQL* basadas en la forma de consulta CONSTRUCT, así como operaciones de actualización basadas en las cláusulas DELETE e INSERT, a clases rdfs:Class (y por herencia clases owl:Class), en donde cada consulta u operación de actualización representa una regla de inferencia que se aplica a todos los individuos de la clase correspondiente y de sus subclases. Aquí, cabe mencionar que, por defecto, un motor de reglas (o de restricciones) *SPIN* enlaza cada uno de los individuos de la clase asociada a la variable predefinida "?this".

[5] http://www.topquadrant.com/technology/sparql-rules-spin/

Concretamente, las reglas representadas por consultas basadas en la forma CONSTRUCT, permiten inferir tripletas adicionales a partir de lo que se establece en la cláusula WHERE; estas tripletas nuevas forman parte de un grafo especial de inferencias que está incluido en el grafo por defecto sobre el que se evalúa la cláusula WHERE; esto permite que los resultados de una primera inferencia estén disponibles para una inferencia posterior. Las reglas representadas por operaciones de actualización basadas en las cláusulas DELETE e INSERT, por su parte, conducen a afirmaciones no inferencias, por lo que no requieren de grafos de inferencias, pudiendo resultar en una ejecución más rápida.

La propiedad spin:constraint permite vincular consultas *SPARQL* basadas en las formas de consulta ASK y CONSTRUCT a clases rdfs:Class (y por herencia clases owl:Class), en donde cada consulta representa una restricción que se asume que ningún individuo de la clase asociada y sus subclases viola. En detalle, una restricción representada por una consulta basada en la forma ASK se evalúa para cada individuo de una clase, de modo que si para alguno de ellos la consulta devuelve un valor *true* significa que ese individuo viola la restricción. Alternativamente, las restricciones representadas por consultas basadas en la forma CONSTRUCT permiten crear instancias de la clase spin:ConstraintViolation a fin de proveer información adicional acerca de las violaciones ocurridas, a cambio su diseño es más complejo comparado con el de las restricciones descritas arriba.

La propiedad spin:constructor, como la propiedad spin:rule, permite vincular consultas *SPARQL* basadas en la forma de consulta CONSTRUCT, así como operaciones de actualización basadas en las cláusulas DELETE e INSERT, a clases rdfs:Class (y por herencia clases owl:Class); no obstante su uso más común es con la forma CONSTRUCT, en donde define una regla de inferencia que se aplica a todos los individuos de la clase correspondiente, aunque en este caso, a diferencia de las reglas definidas mediante la propiedad spin:rule, la regla se ejecuta usualmente solo en el momento de la creación de cada instancia, con el objetivo de inicializar ciertas propiedades de la misma.

En la Figura 2.14 se muestra un ejemplo de una regla de inferencia a partir de una consulta *SPARQL* basada en la forma de consulta *CONSTRUCT*. Este ejemplo aparece originalmente en la sección 2.1.1. del documento de especificación del vocabulario de descripción de clases de *SPIN* (Knublauch, 2011a).

```
ex:Person
    a       rdfs:Class ;
    rdfs:label "Person"^^xsd:string ;
    rdfs:subClassOf owl:Thing ;
    spin:rule
        [ a       sp:Construct;
          sp:templates ([ sp:object sp:_grandParent ;
                          sp:predicate ex:grandParent ;
                          sp:subject spin:_this
                        ]) ;
          sp:where ([ sp:object spin:_this ;
                      sp:predicate ex:child ;
                      sp:subject sp:_parent
                    ] [ sp:object sp:_parent ;
                      sp:predicate ex:child ;
                      sp:subject sp:_grandParent
                    ])
        ] .
```

Figura 2.14. Ejemplo de regla *SPIN* representada por una consulta *SPARQL* basada en la forma de consulta *CONSTRUCT*.

En la figura anterior se puede observar la sintaxis básica de las reglas *SPIN*. Primeramente, es importante mencionar que el prefijo “spin” hace referencia al espacio de nombres del vocabulario *RDF* de descripción de clases; mientras que el prefijo “sp” hace referencia al espacio de nombres del vocabulario *RDF* para representación de consultas *SPARQL* como tripletas *RDF*.

Por otro lado, nótese cómo la cláusula WHERE se codifica como un valor de la propiedad *RDF* “sp:where”, cuyo dominio es la clase “sp:Query” (sintaxis *RDF* para *SPARQL*). Dicho valor es una lista de elementos que se almacena como una lista “rdf:List”, en donde cada elemento es un individuo de una subclase de la clase “sp:Element”, clase que representa a las entidades que forman parte de las cláusulas WHERE en las consultas *SPARQL*, por ejemplo, patrones de tripletas, asignación a variables o sub-consultas. En el caso del tipo de consulta que comprende un patrón básico de grafo (como la consulta mostrada en la Figura 6), cada uno de los patrones de tripleta que componen dicho patrón se representa usualmente como un nodo en blanco no tipificado con exactamente un valor para las propiedades “sp:object”, “sp:predicate” y “sp:subject”; opcionalmente, dicho nodo en blanco puede indicar la propiedad “rdf:type” con el valor “sp:TriplePattern” (la subclase de la clase “sp:Element” que representa patrones de tripletas).

En lo que respecta a la cláusula CONSTRUCT, esta se codifica como un valor de la propiedad “sp:templates”, cuyo dominio es la clase “sp:Construct”. A diferencia de la cláusula WHERE, la cláusula CONSTRUCT se codifica específicamente como una lista de patrones de tripletas (almacenada como una lista “rdf:List”) a fin de representar una plantilla de grafo; cada uno de los patrones de tripleta en dicha plantilla se representa usualmente como un nodo en blanco no tipificado con exactamente un valor para las propiedades “sp:object”, “sp:predicate” y “sp:subject; opcionalmente, dicho nodo en blanco puede estar tipificado, esto es, indicar la propiedad “rdf:type” con el valor “sp:TriplePattern” (la subclase de la clase “sp:Element” que representa patrones de tripletas).

En general, las consultas se representan como instancias de las subclases de la clase “sp:Query”, a saber, las clases “sp:Ask”, “sp:Select”, “sp:Construct” y “sp:Describe”. En detalle, estas se representan como nodos en

blanco tipificados, y se asocian a las clases en los modelos de dominio mediante las propiedades "spin:rule" y "spin:constraint" (en el ejemplo mostrado en la Figura 6 se asocia una regla representada por una consulta basada en la forma de consulta CONSTRUCT a una clase llamada "Person"). Adicionalmente, las consultas en la notación *RDF* para *SPARQL* de *SPIN* se pueden representar con la forma textual de las consultas *SPARQL* utilizando la propiedad "sp:text", cuyo dominio es la clase "sp:Query". Esto puede contribuir a la legibilidad de las reglas y restricciones y, además, puede ser útil para aquellas herramientas que no soportan completamente la sintaxis *RDF* para *SPARQL* de *SPIN*.

2.7.4. Linked Data

Linked Data es la pieza restante del *puzzle* que representa a la Web de Datos. Esta pieza es la que permite a las aplicaciones en el entorno habilitado por las tecnologías de la Web Semántica acceder no solo a los datos en su estado puro sino a las relaciones entre los mismos, a fin de materializar dicha Web de Datos como una colección de *datasets* interconectados a la que se le denomina popularmente *Linked Data*.

A diferencia de las subsecciones anteriores de esta sección, la presente no está orientada a una descripción basada en los distintos tipos de especificaciones técnicas provistas por el *W3C*, sino solo a la descripción de los principios detrás de las buenas prácticas para la publicación de datos enlazados de modo que sea fácil para otros desarrolladores consultar datos procedentes de distintas fuentes de una vez sin la necesidad de que estos compartan un mismo esquema común. Como se puede deducir, dichos principios son perpendiculares a todos los bloques de construcción de la Web Semántica, y permiten, por tanto, la integración de datos a gran escala y el razonamiento sobre los grandes volúmenes de datos integrados.

De acuerdo con la nota sobre el diseño de *Linked Data* de Tim Bernes-Lee (año 2006)[6], son cuatro los principios fundamentales detrás de *Linked Data* como paradigma: (1) el uso de *URIs* para identificar "cosas", (2) el uso de *URIs HTTP* de tal suerte que las personas y los ordenadores (agentes de software) puedan encontrar dichos identificadores fácilmente, (3) la provisión de información relevante sobre las "cosas" como respuesta a la búsqueda de identificadores y (4) la inclusión de enlaces a otros *URIs* a fin de permitir el descubrimiento de nuevas "cosas".

Basándose en estos cuatro principios, el *W3C* ha publicado una serie de diez buenas prácticas para la publicación de datos enlazados mediante el uso de la familia de estándares para representación de datos de *RDF* y el estándar de lenguaje de consulta *SPARQL*. De esta serie de buenas prácticas, que abarca desde tareas de gestión y administración de información, hasta tareas de desarrollo, destacan las que se explican a continuación.

El uso de "buenas" *URIs*: está relacionada con los dos primeros principios definidos previamente por Berners-Lee; tiene como objetivo conseguir un efecto de red a gran escala al aprovechar el alcance global de la tecnología *URI*. El uso de *HTPP URIs* en concreto, hace posible a las personas la recuperación de una representación de los recursos representados por dichos *URIs*, lo que se conoce típicamente como "desreferenciación".

El uso de vocabularios estandarizados: en el entendimiento de que es preferible reutilizar vocabularios existentes antes que crear nuevos vocabularios, el *W3C* alienta a los desarrolladores a utilizar, en la medida de lo posible, vocabularios estandarizados para promover la inclusión y expansión de la Web de Datos.

Proveer (a los ordenadores) acceso automático a los datos enlazados: beneficiarse de la paleta de métodos disponibles para permitir el acceso y lectura automática de los datos en la Web, esta incluye *APIs RESTful* (del inglés *REpresentational State Transfer*), *endpoints SPARQL*, resolución directa de *URIs* y descargas de

[6] https://www.w3.org/DesignIssues/LinkedData.html

ficheros. Esto se basa, además, en la suposición de que se provee al menos una representación procesable por ordenador por cada recurso identificado por un *URI*.

Estas y otras buenas prácticas han sido consideradas en esta tesis doctoral; como se verá en el próximo capítulo de este documento, con el objetivo de integrar los vocabularios de distintas *APIs* de redes sociales y otros servicios basados en localización, como sitios Web de opiniones de usuarios, que proveen contenido heterogéneo en el dominio de la restauración.

2.8. Objetivos

2.8.1. Motivación

Con la creciente popularidad de las redes sociales basadas en localización y los sitios Web de opiniones de usuarios a lo largo de distintos dominios, un entendimiento más amplio y exacto acerca de las preferencias de los usuarios está cada vez más al alcance de las empresas que anuncian sus servicios a través de dichas redes sociales y sitios Web de opiniones. Al mismo tiempo, los usuarios de dichos servicios están cada vez más en posición de tomar decisiones más y mejor informadas acerca de que servicios cumplen sus requerimientos en circunstancias específicas al considerar las sugerencias de otros usuarios respecto a sus experiencias personales con los servicios.

No obstante, resulta imprescindible, para las empresas y para los usuarios, el uso de herramientas de software que brinden algún tipo de soporte frente al problema de la sobrecarga de información acerca de las preferencias de los usuarios y de los servicios en sí, respectivamente. Una solución natural a este problema la representan los sistemas de recuperación de información y, específicamente, los sistemas de recomendación. Como se puede deducir a partir de los resultados del análisis del estado del arte presentados anteriormente, el uso de este tipo de herramientas de software está bastante extendido en los dominios del turismo y del ocio. En este contexto, como sectores de la economía, la industria del ocio no está necesariamente relacionada con actividades soportadas por turistas; sin embargo, como revela dicho análisis, terceras industrias relacionadas con actividades parcialmente soportadas como consecuencia de la afluencia turística, por ejemplo, la industria de la restauración, no cuentan actualmente con soporte en la misma medida en la que cuenta con soporte la industria del ocio.

De acuerdo con la Asociación Nacional de Restaurantes de Estados Unidos (*National Restaurant Association*, *NRA*), las ventas de la industria de la restauración en Estados Unidos representan actualmente 4% del Producto Interno Bruto (PIB). En Europa, el sector de la hostelería, el cual incluye los sectores de la hotelería y de la restauración (restaurantes, bares, pubs y cafés), es un contribuyente importante de la economía europea, con un gran impacto en la generación de empleos y en el crecimiento económico. De acuerdo al reporte de la firma de servicios profesionales Ernst & Young (actualmente EY), comisionado por la asociación europea sin ánimos de lucro The Brewers of Europe, y soportado por la asociación europea de hoteles, restaurantes y cafés (HOTREC), del año 2010, el sector de la hostelería soportó alrededor de 10,2 millones de empleos directos en Europa, esto es, cerca del 5% de la fuerza de trabajo europea; esto representó para la economía europea un crecimiento de más de 460 billones de euros, es decir, el 3,7% del PIB de Europa de ese año.

Por otro lado, si bien existen propuestas en este sentido, especialmente en el campo de los sistemas de recomendación basada en filtrado colaborativo, se requiere de una mayor investigación en técnicas de recomendación más sofisticadas, concretamente, técnicas basadas en modelos estadísticos de clases latentes, y más concretamente, técnicas basadas en el modelo *LDA*, los cuales, aunque han sido explotados para los propósitos de la minería, representación y modelado de información contextual, no han sido ampliamente aplicados en las tareas de recomendación propiamente dichas.

2.8.2. Objetivo general y objetivos específicos

El objetivo que se pretende alcanzar con el desarrollo de esta tesis es el siguiente.

Diseñar e implementar un método híbrido, basado en conocimiento y en filtrado colaborativo bajo un enfoque de modelos estadísticos de clases latentes, de recomendación sensible al contexto para el dominio de la restauración en el contexto de las *APIs* de redes sociales basadas en localización y de sitios Web de opiniones de usuarios.

Este objetivo se puede descomponer a su vez en una serie de objetivos específicos que son mencionados a continuación.

- Diseñar y construir un modelo ontológico basada en el lenguaje *OWL* (y base de reglas basada en la notación *SPIN*) del dominio de la restauración que permita integrar y enlazar datos de *APIs* heterogéneas de redes sociales basadas en localización y de sitios Web de opiniones de usuarios.
- Diseñar e implementar una técnica basada en el *framework RDF* de recuperación de datos semi-estructurados e instanciación automática del modelo ontológico del dominio a partir de *APIs* heterogéneas de redes sociales basadas en localización y de sitios Web de opiniones de usuarios.
- Diseñar e implementar una técnica de modelado de información contextual basada en el modelo generativo de tópicos *LDA* que aproveche el modelo ontológico del dominio para inferir información contextual de alto nivel.
- Diseñar e implementar una técnica de perfilamiento de usuarios basada en el modelo ontológico del dominio y en el modelo *LDA* de información contextual de alto nivel.
- Diseñar e implementar una métrica de similitud semántica basada en el modelo ontológico del dominio que permita capturar conocimiento taxonómico y no taxonómico explícito e inferido acerca de los establecimientos de alimentos y bebidas.
- Diseñar e implementar una técnica de recomendación de filtrado colaborativo basada en la métrica de similitud semántica y en la técnica de perfilamiento de usuarios.
- Diseñar una arquitectura de software que permita integrar las distintas técnicas diseñadas e implementadas en un conjunto de componentes de software interoperables.
- Validar el método y la arquitectura propuesta mediante la implementación de un prototipo de sistema de recomendación como prueba de concepto.

2.8.3. Hipótesis

La hipótesis con la que se pretende demostrar la veracidad de la tesis de esta investigación se puede expresar en el siguiente enunciado.

La combinación de tecnologías de la Web Semántica y modelos estadísticos de clases latentes permitirá el diseño y la implementación de técnicas más potentes, tanto de representación y modelado de información contextual, como de recomendación sensible al contexto.

A su vez, esta hipótesis se puede descomponer en las siguientes sub-hipótesis.

Sub-hipótesis 1. Es posible emplear en conjunto tecnologías de la Web Semántica, específicamente lenguajes de definición y descripción de vocabularios y lenguajes de reglas, y modelos estadísticos de clases latentes, específicamente modelos probabilísticos generativos de tópicos, para el diseño y la implementación de técnicas de representación y modelado de información contextual y de recomendación sensible al contexto.

- ¿Cuáles son las ventajas y desventajas de las aproximaciones existentes para la combinación de tecnologías de la Web Semántica y modelos estadísticos de clases latentes para los propósitos antes mencionados?

- ¿Cuáles son los lenguajes de definición y descripción de vocabularios y los lenguajes de reglas de la Web Semántica más apropiadas para los propósitos antes mencionados?
- ¿Cuáles son los modelos estadísticos de clases latentes más apropiados para los propósitos antes mencionados?

Sub-hipótesis 2. Las técnicas de representación y modelado de información contextual resultantes permitirán la representación y el modelado de información contextual compleja potencialmente relevante para los procesos de recomendación sensible al contexto.

- ¿Qué tipos de información contextual compleja son comúnmente consideradas por las técnicas de representación y modelado de información contextual resultantes?
- ¿Cuáles son los tipos de información contextual compleja más relevantes en el dominio de la restauración?

Sub-hipótesis 3. Las técnicas de recomendación sensible al contexto resultantes propiamente dichas harán posible la generación de recomendaciones más exactas en el sentido tradicional de la palabra y más realistas desde el punto de vista del usuario).

- ¿En que medida las técnicas de recomendación sensible al contexto resultantes mejoran la exactitud de las recomendaciones?
- ¿En que medida las recomendaciones resultantes son más realistas desde el punto de vista del usuario?

2.8.4. Metodología

La metodología a seguir en esta investigación a fin de alcanzar los objetivos propuestos y permitir la demostración de la tesis de la investigación comprende principalmente cuatro tareas.

1. Análisis del estado del arte. Esta primera tarea representa el estudio del estado del arte en dos vertientes principales: (1) sistemas de recomendación y (2) tecnologías de la Web Semántica, con un claro énfasis en los puntos de convergencia entre ambas áreas. A continuación, se ahonda en las dos subtareas correspondientes al análisis del estado del arte en las áreas antes mencionadas.
    a. Sistemas de recomendación. Estudio del estado de la técnica en sistemas de recomendación basada en filtrado colaborativo, sistemas de recomendación basada en conocimiento y sistemas de recomendación híbridos, enfatizando en los antecedentes históricos de los mismos y en los tipos de técnicas de recomendación y métodos de hibridación comúnmente empleados por estos.
    b. Web Semántica. Estudio técnico del estado actual de las tecnologías y bloques de construcción de la pila de la Web Semántica, incluyendo los lenguajes de descripción y definición de vocabularios, *OWL* y *RDFS*, el *framework* de propósito general para la representación de datos en la Web, *RDF*, el lenguaje de consulta para *RDF*, *SPARQL*, y la notación para la definición basada en *SPARQL* de reglas de inferencia y restricciones, *SPIN*, vocabularios y ontologías.
2. Formalización de la propuesta. Esta tarea comprende una serie de subtareas descritas en detalle en el siguiente capítulo de este documento, las cuales resultan en la definición de un método híbrido, basado en conocimiento y en filtrado colaborativo bajo un enfoque de modelos estadísticos de clases latentes, de recomendación sensible al contexto para el dominio de la restauración en el contexto de las APIs de redes sociales basadas en localización y de sitios Web de opiniones de usuarios. A continuación, se ahonda en las subtareas correspondientes a la formalización de la propuesta a un alto nivel de abstracción.

   a. Diseño y construcción de un modelo ontológico del dominio de la restauración y una técnica de integración y enlazamiento de conocimiento semántico a partir de datos de *APIs* heterogéneas de servicios basados en localización y sitios Web de opiniones de usuarios.
   b. Diseño e implementación de una técnica de representación y modelado de información contextual basada en modelos estadísticos de clases latentes.
   c. Diseño e implementación de una métrica de similitud semántica basada en ontologías.
   d. Diseño e implementación de una técnica de recomendación híbrida basada en conocimiento de filtrado colaborativo basado en modelos estadísticos de clases latentes.
   e. Diseño de una arquitectura de software que permita integrar las distintas técnicas diseñadas e implementadas en un conjunto de componentes de software interoperables.
3. Implementación de la propuesta. Esta tarea representa la implementación de un prototipo de sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas a partir de una arquitectura de software diseñada como resultado de las tareas de formalización de la propuesta.
4. Validación de la propuesta. Esta última tarea comprende el diseño y ejecución mediante un estudio de usuario y un experimento *offline* de un método doble de evaluación en forma de análisis comparativo basado en un enfoque de Ciencias de la Información, específicamente, un enfoque de recuperación de información orientado a la medición de la exactitud bajo dos interpretaciones distintas: predicción de ratings y predicción de uso. El estudio de usuario y el experimento *offline* permiten evaluar el método de recomendación propuesto bajo dos escenarios distintos, respectivamente: escasez de *ratings* y suficiencia de *ratings*.

2.9. Conclusión

Las amenazas y debilidades inherentes de cada técnica de recomendación existente en la literatura de sistemas de recomendación han obligado a los investigadores de dicha área a buscar maneras diversas de combinar dichas técnicas a fin de aligerar o compensar dichas amenazas y debilidades, lo que ha dado lugar a la conceptualización de la idea de sistema de recomendación híbrido. En dicha búsqueda, las técnicas de recomendación basada en conocimiento han jugado tradicionalmente un papel fundamental; originadas en el campo de la investigación en sistemas de razonamiento basado en casos, este tipo de técnicas posibilitan una nueva visión de cara al problema de la sobrecarga de información. En este contexto, las técnicas basadas en tecnologías de la Web Semántica permiten capturar un tipo de conocimiento distinto: conocimiento semántico, y además representan, junto con otras propuestas en representación de conocimiento basada en semántica, un tipo de técnicas de recomendación de naturaleza distinta frente a técnicas de recomendación primigenias, especialmente frente a las técnicas de filtrado basado en contenido consideradas técnicas sintácticas: técnicas semánticas.

Por otro lado, en la búsqueda de soluciones para la generación de recomendaciones no solo más exactas sino más relevantes desde el punto de vista del usuario, la investigación en el área de sistemas de recomendación ha recurrido desde hace casi dos décadas al uso de características de sistemas sensibles al contexto, dando lugar a una línea de investigación con un sin número de esfuerzos en desarrollo: sistemas de recomendación sensibles al contexto. En este contexto, resulta imprescindible el desarrollo de técnicas más eficaces y eficientes para la representación y el modelado de la información contextual.

Según los resultados del análisis del estado del arte presentado en este capítulo, los modelos estadísticos de clases latentes representan un esfuerzo actual importante en este sentido, así como en el desarrollo de técnicas de recomendación más eficaces y eficientes. En particular, las técnicas de recomendación basadas en modelos probabilísticos de clases latentes corresponden a la categoría de técnicas de filtrado colaborativo basado en modelos, la cual representa una línea de investigación actual muy activa en el área de investigación en sistemas

de recomendación dadas sus fortalezas y oportunidades respecto a la categoría de técnicas de filtrado colaborativo originarias.

No obstante, según dichos resultados existe una evidente brecha entre la investigación en el uso de las tecnologías de la Web Semántica en el campo de la investigación en sistemas de recomendación y la investigación en el uso de los modelos probabilísticos de clases latentes en el mismo campo de investigación con dos objetivos principales: la creación de técnicas híbridas de recomendación y la creación de técnicas híbridas de modelado y representación de información contextual. En este sentido, la propuesta de esta tesis doctoral pretende aprovechar los posibles puntos de convergencia entre dichos tipos de tecnologías y técnicas computacionales y contribuir a cerrar dicha brecha en el largo plazo.

Capítulo 3 . Método Propuesto

3.1. Introducción

En esta investigación se han aprovechado las tecnologías de la Web Semántica, principalmente los lenguajes de descripción y definición de vocabularios, *OWL* y *RDFS*, el *framework* de propósito general para la representación de datos en la Web, *RDF*, el lenguaje de consulta para *RDF*, *SPARQL*, y la notación para la definición basada en *SPARQL* de reglas de inferencia y restricciones, *SPIN*, así como el modelo probabilístico generativo de tópico, *LDA*, con el objetivo de diseñar e implementar un método híbrido, basado en conocimiento y en filtrado colaborativo bajo un enfoque estadístico de clases latentes y basado en memoria, de recomendación sensible al contexto de establecimientos de alimentos y bebidas.

Específicamente, dichas técnicas y tecnologías computacionales se han aprovechado en esta tesis doctoral a fin de dar lugar a una serie de contribuciones primordiales en dos áreas principales: representación de conocimiento y razonamiento.

En el primer caso se ha propuesto un modelo ontológico basado en *OWL* del dominio de la restauración, el cual pretende, en primer lugar, enlazar y, finalmente integrar, los vocabularios de las *APIs* de redes sociales basadas en localización y otros servicios basados en localización, como sitios Web de opiniones de usuarios, que proveen contenido heterogéneo en dicho dominio, utilizando un enfoque de *Linked Data*, y, en segundo lugar, modelar las preferencias y el contexto de los usuarios de los establecimientos en dicho dominio. Asimismo, se ha propuesto una técnica híbrida de modelado y representación de información contextual basada en modelos probabilísticos generativos de tópicos, específicamente modelos *LDA*, la cual aprovecha el modelo ontológico del dominio, así como una base de reglas de inferencia representadas bajo la notación *SPIN*, para la inferencia de información contextual temporal y social de alto nivel a partir de información relacionada de bajo nivel.

De acuerdo con lo anterior, la técnica de modelado y representación de información contextual propuesta en esta investigación representa también una contribución en el área del razonamiento de conocimiento; en el mismo ámbito se ha propuesto una métrica de similitud semántica basada en ontologías, la cual emplea un enfoque híbrido taxonómico/no taxonómico de conjuntos de características ontológicas, y permite capturar, además de conocimiento taxonómico, conocimiento no taxonómico explícito e inferido. En este contexto, y en el ámbito específico de la recomendación, a manera de *framework* se ha propuesto una técnica híbrida de filtrado colaborativo basado en modelos y de filtrado colaborativo basado en memoria bajo el enfoque de top-n recomendaciones, en donde el componente basado en modelos lo representa el modelo *LDA* de información contextual de alto nivel (modelo probabilístico generativo de tópicos), y el componente basado en memoria lo representa la métrica de similitud semántica basada en ontologías.

En este capítulo se describen detalladamente las subtareas correspondientes a la tarea de formalización de la metodología definida a fin de dar alcance a los objetivos propuestos y permitir la demostración de la tesis de la investigación mediante la aplicación rigurosa del método científico. La ejecución de dichas subtareas hace posible la obtención de las contribuciones antes descritas. A un alto nivel de abstracción las subtareas correspondientes a la tarea de formalización de la propuesta comprenden: (1) el diseño del modelo ontológico del dominio y el diseño y la implementación de una técnica de integración y enlazamiento de conocimiento semántico a partir de datos de *APIs* heterogéneas de servicios basados en localización en dicho dominio, (2) el diseño y la implementación de la técnica de representación y modelado de información contextual basada en el modelo probabilístico generativo de tópicos, *LDA*, y en reglas de inferencia (3) el diseño y la implementación de la métrica de similitud semántica basada en ontologías, (4) el diseño y la implementación de la técnica de recomendación híbrida basada en conocimiento de filtrado colaborativo basada en modelos probabilísticos generativos de tópicos (5) el diseño de una arquitectura de software que permita integrar las distintas técnicas diseñadas e implementadas en un conjunto de componentes de software interoperables.

3.2. Arquitectura de Software Propuesta

La Figura 3.1 muestra la arquitectura de software a partir de la cual es posible implementar el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas propuesto en esta tesis doctoral. Como se ha dejado entrever en el capítulo anterior, a fin de validar la propuesta de esta tesis, se ha construido un prototipo del sistema de recomendación resultante; no obstante, en este capítulo en ocasiones se hace referencia a dicho sistema como concepto.

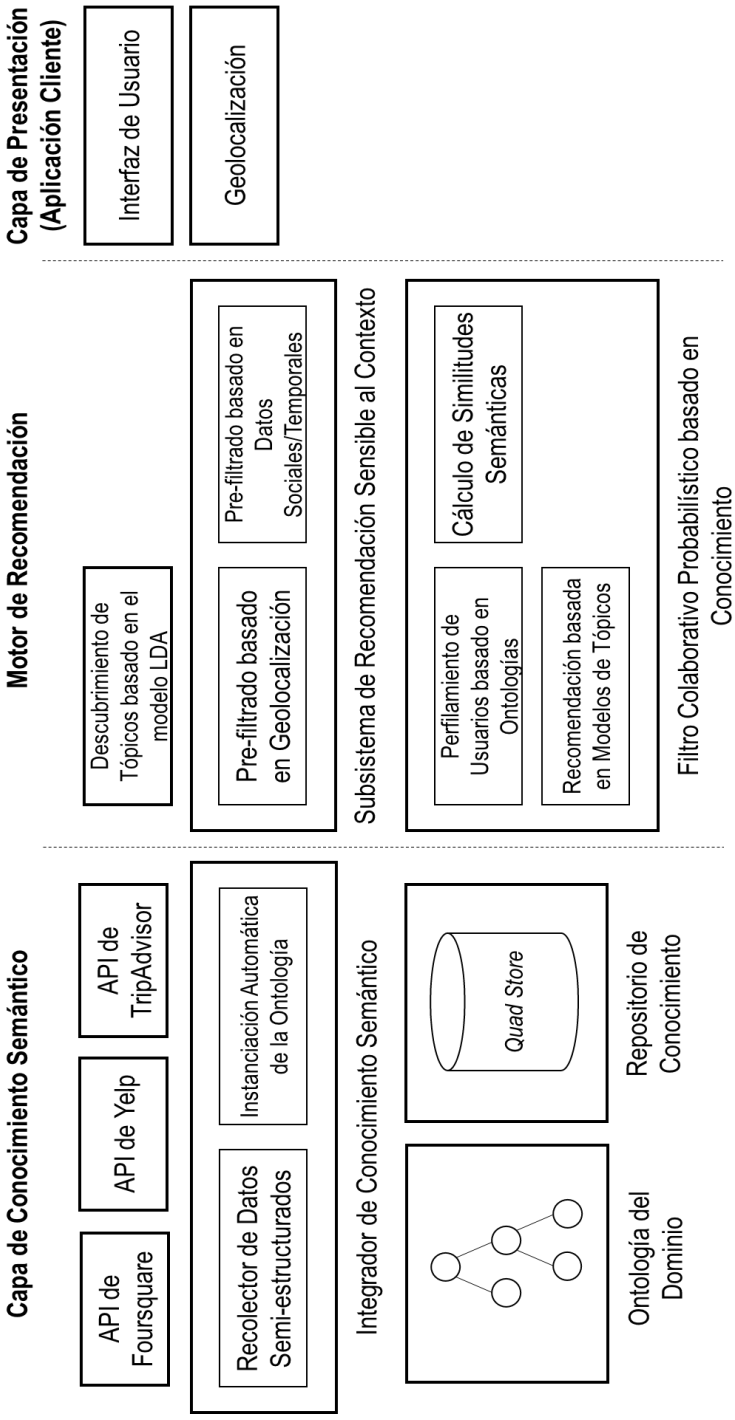


Figura 3.1. Arquitectura de software propuesta.

A un alto nivel de abstracción, la arquitectura propuesta se compone de ocho componentes principales, a saber, el integrador de conocimiento semántico, la ontología del dominio, el repositorio de conocimiento, el componente encargado del descubrimiento de tópicos basado en el modelo *LDA*, el subsistema de recomendación sensible al contexto, el componente encargado del perfilamiento de usuarios basado en ontologías, el filtro colaborativo probabilístico basado en conocimiento y la interfaz de usuario.

Como se puede observar en la Figura 1, estos componentes están organizados en una arquitectura lógica cliente/servidor de tres capas, en la que cada capa tiene un rol especifico que depende de los roles individuales de sus componentes. Se ha propuesto un único punto de acceso a la arquitectura: la interfaz de usuario. Junto con la funcionalidad de geolocalización (la capa de presentación), la interfaz de usuario se ha implementado como una aplicación móvil nativa basada en tecnologías Web, a la cual de aquí en adelante se hace referencia como aplicación cliente. De acuerdo con esta descripción, la capa del motor de recomendación representa realmente una serie de servicios Web accesibles para la aplicación cliente. La comunicación entre dichas capas descansa en el formato de texto ligero para el intercambio de datos, *JavaScript Object Notation* (*JSON*); el lenguaje utilizado para la implementación de la capa del motor de recomendación y la capa de conocimiento semántico es Java.

En las siguientes secciones de este capítulo se describen en detalle las etapas del método de recomendación sensible al contexto de establecimiento de alimentos y bebidas en el contexto de los componentes de la arquitectura propuesta para su implementación.

3.3. Definición de la Ontología del Dominio

La ontología del dominio en esta tesis doctoral se definió manualmente utilizando el lenguaje OWL 2 Web Ontology Language 2. El objetivo de definir una nueva ontología para el dominio de la restauración, a pesar de las propuestas que pudiesen existir, era más bien enlazar y, finalmente integrar, los vocabularios (formalmente las partes relevantes para el dominio de interés) de las *APIs* de redes sociales y otros servicios basados en localización, como sitios Web de opiniones de usuarios, que proveen contenido heterogéneo en este dominio, utilizando un enfoque de *Linked Data*. A saber, en esta tesis doctoral se ha intentado integrar los vocabularios de las *APIs* de Foursquare, Yelp y TripAdvisor.

No obstante, se ha utilizado como base la versión *OWL* del vocabulario Squema.org desarrollada por TopQuadrant; específicamente, se ha utilizado la clase “Restaurant”, la cual es subclase de clases más generales como “FoodEstablishment”, “LocalBusiness”, “Organization” y “Place”, así como las clases “GeographicCoordinates”, “PostalAddress”. En términos generales, Squema.org es una iniciativa de colaboración lanzada en el año 2011 con la misión de crear, mantener y fomentar esquemas para datos estructurados en Internet; actualmente es patrocinada por empresas como Google, Yahoo!, Yandex y Microsoft.

El núcleo de la ontología en esta investigación es una jerarquía de clases que representa categorías de establecimientos de alimentos y bebidas, cuya clase raíz es la clase “FoodEstablishment”, y una jerarquía de clases que representa categorías de estilos de cocina, cuya clase raíz es la clase “Cuisine”.

En este contexto, es importante mencionar que, si bien las *APIs* de redes sociales y otros servicios basados en localización proveen taxonomías generales de lugares, estas generalmente contienen subcategorías de establecimientos de alimentos y bebidas que hacen referencia a estilos de cocina. No obstante, en algunas ocasiones, dichas *APIs* proveen listas independientes de estilos de cocina (formalmente no son taxonomías) para el caso particular de los lugares en este dominio, las cuales pueden extenderse a fin de clasificar cada lugar bajo una o más categorías de lugar de alimentos y bebidas y, al mismo tiempo, una o más categorías de estilos de cocina.

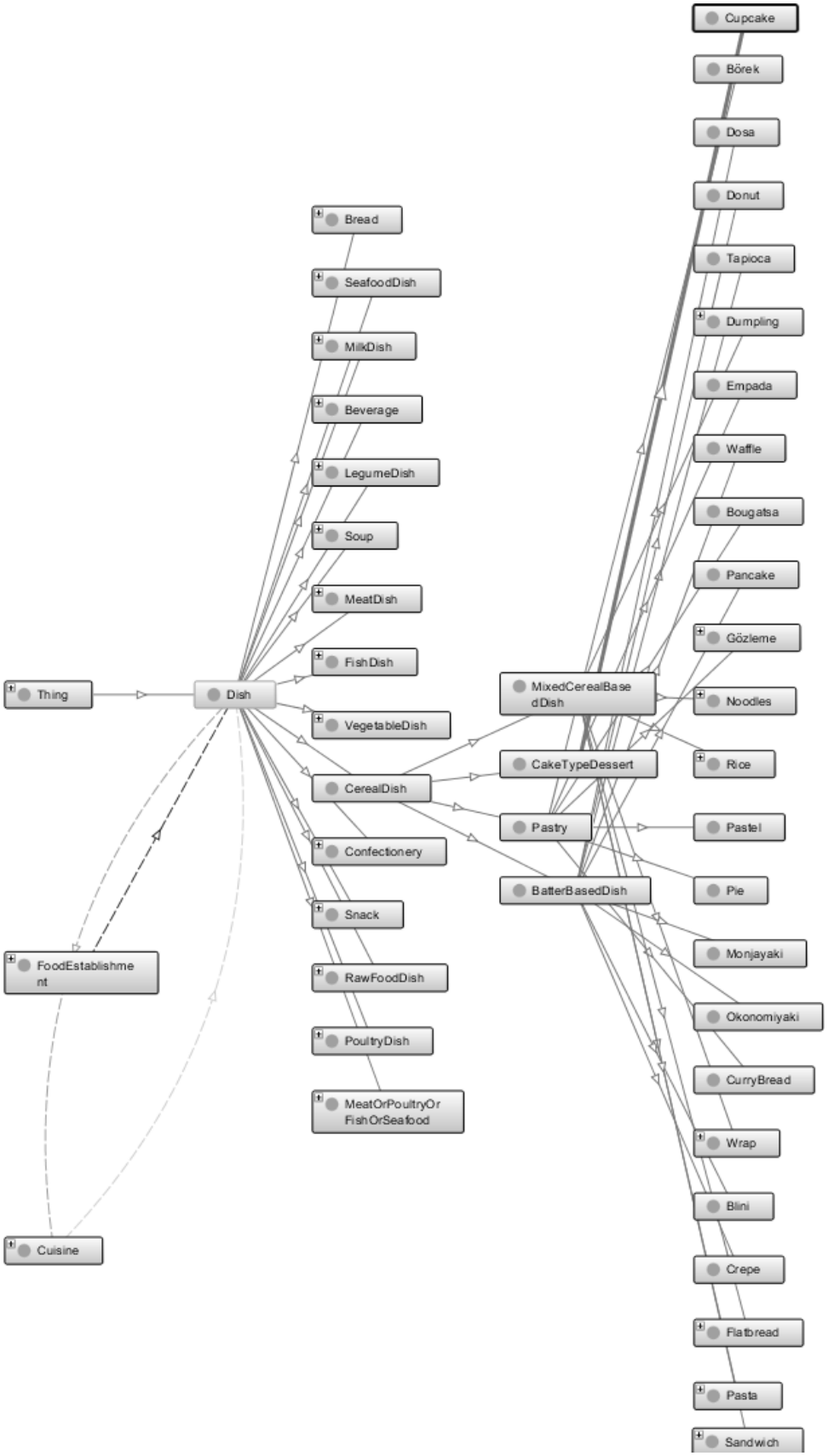
Así, las jerarquías de clases antes mencionadas se han creado aprovechando dicha oportunidad. En particular, los estilos de cocina a los que se hace referencia en las categorías de lugares se han definido como subclases en la jerarquía de clases que representa categorías de estilos de cocina, y al mismo tiempo las categorías de lugares en sí se han definido como clases equivalentes en la otra jerarquía de clases (la que representa categorías de establecimientos de alimentos y bebidas); específicamente, como clases equivalentes a clases anónimas cuya descripción de clase consiste en la intersección de la descripción de clase de la clase general "FoodEstablishment" y una restricción de valor sobre la propiedad de objeto que permite asociar las instancias de dicha clase a instancias de subclases particulares de la clase "Cuisine", a saber, la propiedad de objeto "servesCuisine".

Es importante mencionar que, adicionalmente, se ha creado una jerarquía de clases que representa más bien platillos a los que se hace referencia en las categorías de establecimiento de alimentos y bebidas (la jerarquía de clases cuya clase raíz es la clase "Dish"). Al igual que en el caso de los estilos de cocina, dichas categorías de establecimiento se han definido al mismo tiempo como clases equivalentes en la jerarquía que representa categorías de establecimiento de alimentos y bebidas. En este caso, utilizando una restricción de valor sobre la propiedad "servesDish", cuyo dominio y rango son las clases "FoodEstablishment" y "Dish". En la Figura 3.2 se muestra, respectivamente, un extracto (cuatro niveles) de la jerarquía de clases encabezada por la clase "Cuisine" con énfasis en la subclase que representa a la cocina americana (la clase "AmericanCuisine"), así como un extracto (tres niveles) de la jerarquía de clases encabezada por la clase "Dish" con énfasis en la subclase que representa a los platillos basados en cereales (la clase "CerealDish").

Por otro lado, los estilos de cocina explícitos que denotan realmente categorías de establecimientos de alimentos y bebidas se han definido como subclases en la jerarquía de clases que representa categorías de establecimiento, según conocimiento del dominio existente en recursos externos, específicamente enciclopedias colaborativas como Wikipedia y bases de datos léxicas como Wordnet.

En este proceso de construcción manual de la ontología propuesta se han conservado todas aquellas categorías de establecimiento de alimentos y bebidas y todos aquellos estilos de cocina explícitos que no cumplen con las condiciones antes mencionadas. Por cada estilo de cocina se ha creado además una clase equivalente en la jerarquía de clases que representa categorías de establecimiento. Específicamente, se trata de clases equivalentes a clases anónimas cuya descripción de clase consiste en la intersección de la descripción de clase de la clase general "FoodEstablishment" y una restricción de valor sobre la propiedad de objeto "servesCuisineStyle". Estas clases son complementarias a las clases equivalentes resultantes de las categorías de establecimiento que hacen referencia a estilos de cocina y platos.

a)


Cupcake
Börek
Dosa
Donut
Tapioca
Dumpling
Empada
Waffle
Bougatsa
Pancake
Gözleme
Noodles
Rice
Pastel
Pie
Monjayaki
Okonomiyaki
CurryBread
Wrap
Blini
Crepe
Flatbread
Pasta
Sandwich
Bread
SeafoodDish
MilkDish
Beverage
LegumeDish
Soup
MeatDish
FishDish
VegetableDish
CerealDish
Confectionery
Snack
RawFoodDish
PoultryDish
MeatOrPoultryOrFishOrSeafood
Thing
Dish
FoodEstablishment
Cuisine
MixedCerealBasedDish
CakeTypeDessert
Pastry
BatterBasedDish

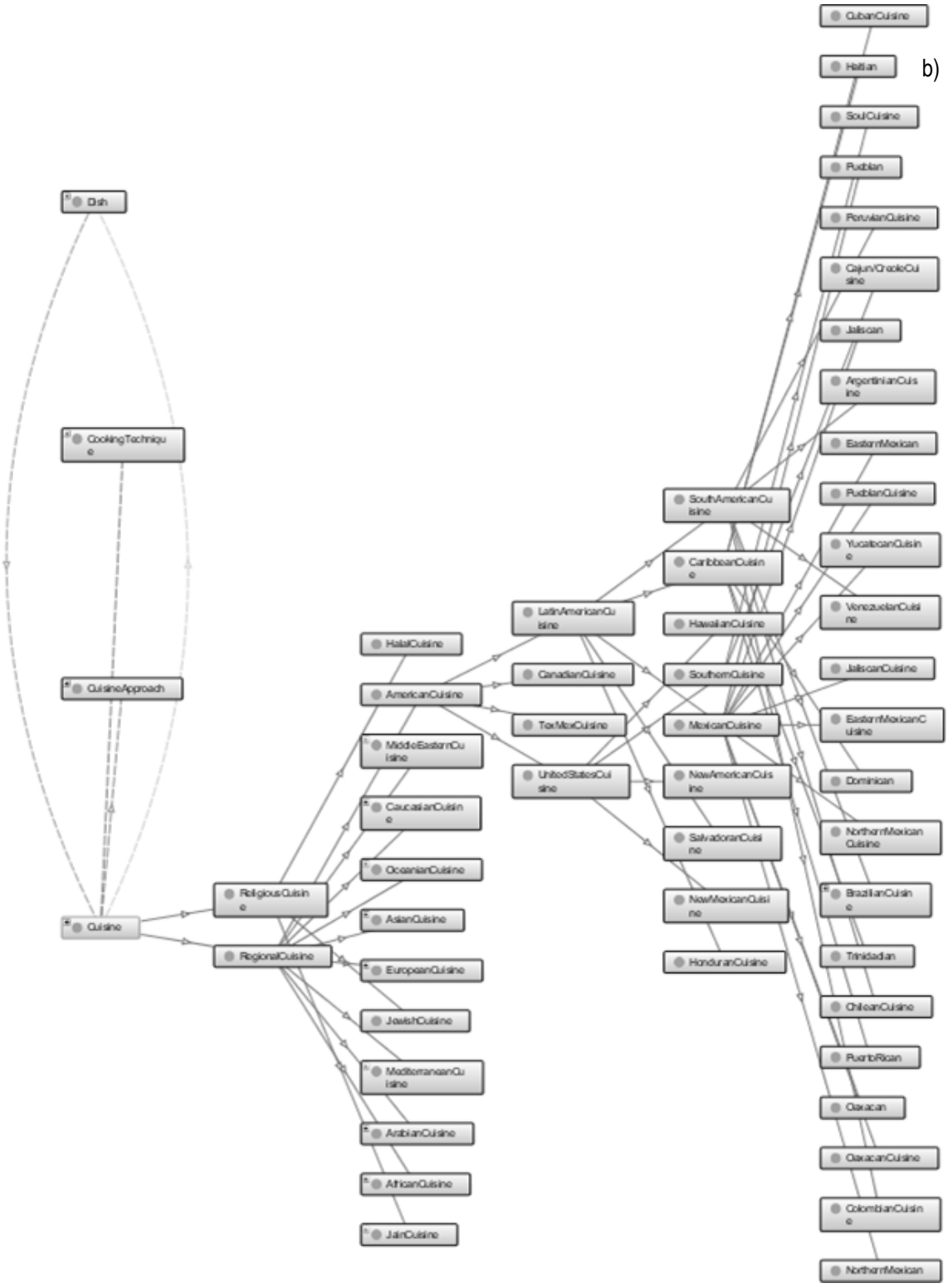


Figura 3.2. Extracto de las jerarquías de clases de la ontología del dominio: a) jerarquía de clases encabezadas por la clase “Dish”, b) jerarquía de clases encbezada por la clase “Cuisine”.

Finalmente, cabe mencionar que la ontología del dominio comprende clases y propiedades de dato y objeto relacionadas con el perfilado de usuarios y contextos, además de clases y propiedades relacionadas con el modelado del dominio propiamente dicho. A continuación, se describen las clases y propiedades más relevantes para estos dos aspectos del método de recomendación sensible al contexto de establecimientos de alimentos y bebidas propuesto en esta tesis doctoral.

- **Person**: esta clase permite modelar información de identificación acerca de cualquier persona en el dominio de la restauración (establecimientos de alimentos y bebidas). En el caso particular de esta tesis doctoral, esta clase es útil para modelar la información de aquellas personas que, sin tener el rol de usuario del sistema, acompañan a los usuarios en las visitas a los establecimientos de alimentos y bebidas.
- **User**: esta clase es subclase de la clase "Person", y permite modelar información adicional acerca de las personas que ocupan el rol de usuario del sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas basado en la arquitectura de software propuesta en esta tesis doctoral (ver Figura 3.1 en la sección anterior) (de aquí en adelante llamado usuario), esto es, las personas que, de hecho, hacen *check-in* en los establecimientos. En detalle, esta clase permite modelar, por ejemplo, ciudades y países de residencia (las propiedades de dato "cityOfResidence" y "countryOfResidence"). Es importante mencionar que las relaciones sociales de los usuarios del sistema, las cuales son fundamentales en el proceso de recomendación, se representan a través de las propiedades de objeto "hasRelative", "hasSignificantOther", "hasSpouse", "hasFriend" y "hasCoWorker", cuyo dominio y rango son las clases "User" y "Person", respectivamente. Los parentescos particulares entre los usuarios y las personas no son relevantes para esta ontología del dominio.
- **GeographicPoint**: esta clase permite modelar un punto geográfico mediante las coordenadas geográficas latitud y longitud (las propiedades de dato "lattitude" and "longitude").
- **PostalAddress**: esta clase es subclase de la clase "GeographicPoint", y permite representar direcciones postales a través de las propiedades de dato "street", "country" y "postalCode", por nombrar solo algunas. En otras palabras, esta clase permite añadir una dirección postal a un punto geográfico representado por las coordenadas latitud y longitud.
- **RecommendationRequest**: esta clase permite representar las solicitudes de recomendación de los usuarios activos en el sistema, y es igualmente utilizada para representar información sobre el contexto en el que dichas solicitudes se realizan, es decir, el contexto de los usuarios activos del sistema. En concreto, una solicitud de recomendación está representada por una instancia de la clase "User" y una o más instancias de la clase "Person" asociadas a una instancia de la clase "RecommendationRequest" a través de las propiedades de objeto "hasUser" y "hasUserCompanion", respectivamente. Los *timestamps* de las solicitudes se modelan como valores de tipo "dateTimeStamp" utilizando la propiedad de dato "timestamp"; de manera similar, las coordenadas de los puntos geográficos desde los cuales los usuarios realizan las solicitudes se modelan como pares de valores de tipo cadena a través de la propiedad de objeto "hasGeographicPoint", cuyo rango es la clase "GeographicPoint". Esta clase también permite modelar información contextual de alto nivel inferida a partir de la información contextual explicita de bajo nivel (utilizando las propiedades de dato "hasSocialSituation", "hasIntendedDayOfWeek" y "hasIntendedPeriodOfDay")
- **Place**: esta clase permite representar lugares, es decir, entidades que tienen extensiones físicas delimitadas, en el contexto de los servicios basados en localización. Particularmente, esta clase permite modelar las direcciones postales de los lugares a través de la propiedad de objeto "hasPostalAddress", cuyo rango es la clase "PostalAddress". Asimismo, esta clase permite modelar el *rating* promedio, así como el número total de *ratings* y el número total de *check-ins* hechos por los

usuarios en los lugares (las propiedades de dato “aggregatedRating”, “ratingCount” y “checkInCount”, respectivamente).

- **Cuisine**: esta clase es la clase raíz de una extensa jerarquía de clases que permite representar estilos de preparación de alimentos, esto es cocinas. Dado que una cocina está comúnmente relacionada con una región, nación o cultura, esta jerarquía de clases incluye clases que representan distintas categorías de cocina, a saber, cocinas regionales o étnicas y cocinas religiosas. De acuerdo con lo anterior, los países en los que se las cocinas tienen relevancia se modelan mediante la propiedad de dato “availableCountry”. Algunos ejemplos de clases de nivel inferior en esta jerarquía de clases son las clases “LatinAmerican”, “Italian” y “AsianFusionCuisine”.
- **CookingTechnique**: esta clase es la clase raíz de una jerarquía de clases que permite representar técnicas o métodos de preparación de alimentos, como la técnica conocida como “Barbecue” o la técnica denominada “Robatayaki”. Debido a que estas técnicas están comúnmente relacionadas con una cocina o con la cocina de una región o etnia en particular, se ha creado la propiedad de objeto “hasCookingTechnique”, cuyo dominio y rango es la clase “Cuisine” y la clase “CookingTechnique”, respectivamente, y que permite modelar dicha asociación.
- **CuisineApproach**: esta clase es la clase raíz de una jerarquía de clases que permite representar categorías de enfoques de cocina más que de estilos. Algunos ejemplos de clases en esta jerarquía son las clases “VegetarianCuisine” y “SignatureCuisine”.
- **Dish**: esta clase es la clase raíz de una jerarquía de clases que permite representar indistintamente platos, comidas o bebidas que, si bien pueden formar parte de un estilo de cocina particular, no se pueden considerar por sí solos estilos. Algunos ejemplos de clases en esta jerarquía de clases son las clases “Pizza”, “FriedChicken” y “Crepe”. El objetivo de esta clase no es permitir representar los distintos platos que representan a los estilos de cocina sino los platos servidos tradicionalmente por los establecimientos de alimentos y bebidas, los cuales, al igual que los estilos de cocina, les pueden conferir categorías de establecimiento de alimentos y bebidas. Esta clase tampoco está destinada a representar los distintos platos que pueden servir los establecimientos en un momento dado (menús).
- **FoodEstablishment**: esta clase es subclase de la clase “Place”, y permite representar aquellos lugares que son específicamente establecimientos de alimentos y bebidas. En particular, permite modelar información de identificación como los nombres legales, los teléfonos y correos electrónicos de contacto, así como los niveles relativos de los precios en los establecimientos en una escala numérica de tres puntos, donde “1” significa “bajo”, y “3” significa “alto” (mediante la propiedad de dato “priceTier”); asimismo, permite modelar los estilos de cocina (y/o los enfoques de cocina) y los platillos servidos por dichos establecimientos a través de las propiedades de objeto “servesCuisine” (y/o “hasCuisineApproach”) y “servesDish”, cuyos rangos son las clases “Cuisine” (y/o “CuisineApproach”) y “Dish”, respectivamente. De hecho, esta clase es la clase raíz de una jerarquía de clases que representa categorías de establecimientos de alimentos y bebidas. Algunos ejemplos de clases en esta jerarquía son las clases “Gastropub”, “Buffet” y “Cafeteria”, así como las clases “PizzaRestaurant” y “Creperie” (clases equivalentes). Cabe mencionar que la clase “FoodEstablishment” es la clase de referencia del método de recomendación propuesto.
- **Rating**: esta clase permite representar los *ratings* dados por los usuarios a los lugares. En detalle, cada instancia de esta clase se relaciona con una instancia de la clase “User” y una instancia de la clase “Place” a través de las propiedades de objeto “hasAuthor” y “hasRatedPlace”. Los *ratings* propiamente dichos se modelan utilizando la propiedad de dato “ratingValue”, y los correspondientes *timestamps* se modelan como valores de tipo “dateTime” utilizando la propiedad de dato “timestamp”.
- **CheckIn**: esta clase permite representar los *check-ins* hechos por los usuarios en los lugares, y es igualmente utilizada para representar información sobre el contexto en el que los *check-ins* se realizaron, es decir, el contexto de los usuarios no activos del sistema. En detalle, cada instancia de

esta clase se relaciona con una instancia de la clase “User”, una instancia de la clase “Place” y una o más instancias de la clase “Person” a través de las propiedades de objeto “hasVisitor”, “hasVisitedPlace” y “hasVisitorCompanion”, respectivamente; como se mencionó anteriormente, las instancias de la clase “Person” representan a las personas que acompañaron a los usuarios durante las visitas a los lugares. Los *timestamps* de los *check-ins* se modelan como valores de tipo “dateTime” utilizando la propiedad de dato “timestamp”. Al igual que la clase “RecommendationRequest”, esta clase permite modelar información contextual de alto nivel inferida a partir de la información contextual explicita de bajo nivel (utilizando las propiedades de dato “visitedInSocialSituation”, “visitedAtDayOfWeek” y “visitedAtPeriodOfDay”).

3.4. Integración de conocimiento semántico

La siguiente etapa en el método de recomendación propuesto es la integración de conocimiento semántico sobre los establecimientos a partir de corpus de documentos semi-estructurados procedentes de *APIs* de redes sociales y otros servicios basadas en localización disponibles en Internet.

En detalle, se trata de la creación automática de instancias y valores de propiedades para las clases y propiedades de dato y objeto en una ontología del dominio (instanciación automática de ontologías) a partir de datos semi-estructurados basados en *XML* y *JSON* procedentes tanto de servicios Web basados en el estilo arquitectónico *Service Oriented Architecture* (*SOA*) como de recursos Web basados en el estilo arquitectónico *RESTful*.

Conceptualmente esta etapa se puede descomponer en dos etapas: (1) la recuperación de los datos semi-estructurados de las fuentes de datos subyacentes y (2) la instanciación automática de la ontología propiamente dicha.

3.4.1. Recuperación de datos semi-estructurados

Para este propósito se ha desarrollado un esquema de clases Java de envoltura para las fuentes de datos subyacentes (*APIs*) de manera que resulte relativamente fácil extender la implementación de la arquitectura propuesta a las diferentes redes sociales y servicios basados en localización, cuyos vocabularios son enlazados en la ontología del dominio propuesta, a saber, Foursquare, Yelp y TripAdvisor.

Este mecanismo tiene sus bases en el trabajo previo del grupo de investigación en integración de servicios de la nube heterogéneos (Colombo-Mendoza, Alor-Hernández, Rodríguez-gonzález, & Valencia-garcía, 2014), y consiste en la creación de una clase adaptadora o envoltorio para adaptar las funcionalidades (servicios o recursos Web) de una *API* particular a la interfaz de una clase en una jerarquía de clases, según su estilo arquitectónico. Así, existe una clase base que abstrae las funcionalidades mínimas requeridas de las *APIs* de redes sociales basadas en localización, así como un par de clases hijas que redefinen dichas funcionalidades de acuerdo a un estilo arquitectónico particular, a saber, *SOA* y *RESTful*.

Es importante mencionar que la recuperación de datos semi-estructurados, y la subsecuente instanciación de la ontología del dominio, se realiza bajo demanda, específicamente, como resultado de la interacción del usuario (a fin de solicitar recomendaciones de establecimientos de alimentos y bebidas para un contexto geográfico determinado) con la interfaz de usuario de la arquitectura propuesta.

En detalle, el tipo de información recuperada durante esta etapa es información de identificación de los establecimientos de alimentos y bebidas, así como información sobre su categoría o tipo, sus estilos de cocina, y las localizaciones geográficas y direcciones de los mismos.

En ocasiones, las *APIs* de redes sociales basadas en localización exponen un servicio o recurso Web para la búsqueda de establecimientos existentes dentro de un área geográfica relativa a una posición geográfica dada.

Como respuesta, dichos servicios o recursos Web proveen un documento con tantos resultados (etiquetas *XML* u objetos *JSON*) como establecimientos encontrados. Cada resultado provee cierta información sobre un establecimiento, como el nombre e identificador del mismo.

En este caso, las *APIs* proveen también servicios o recursos Web que permiten obtener detalles acerca de un establecimiento en particular, entre ellos su categoría (el tipo de establecimiento de alimentos y bebidas), sus estilos de cocina, su localización geográfica (coordenadas latitud-longitud) y los distintos componentes de su dirección, por ejemplo, calle, número y ciudad, al consultar por su identificador. Como respuesta, dichos servicios o recursos Web proveen un documento con un único resultado (etiqueta *XML* u objeto *JSON*) que representa los detalles del lugar consultado.

Por lo tanto, es necesario realizar una llamada al servicio o recurso Web de consulta de detalles sobre un establecimiento, por cada resultado provisto por el servicio o recurso Web de búsqueda de establecimientos. En detalle se ha propuesto un mecanismo de invocación de clientes asíncrona, que utiliza las *APIs* de Java *Java API for XML Web Services* (*JAX-WS*) y *Java API for RESTful Web Services* (*JAX-RS*) para la invocación asíncrona de los servicios Web basados en *SOA* y los recursos Web basados en *RESTful*, respectivamente.

A efectos prácticos, en la siguiente subsección se explicará la etapa de instanciación automática de la ontología del dominio en función del caso antes descrito, lo que de ninguna manera significa que el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas propuesto en esta tesis doctoral sea dependiente de la fuente de datos, dado el esquema de clases de envoltura de fuentes de datos en el que se sustenta la etapa de integración de conocimiento semántico.

3.4.2. Instanciación automática de la ontología del dominio

Esta etapa consiste en el mapeo directo de los datos semi-estructurados recolectados como resultado de etapa anterior como instancias de clases, y valores de propiedades de tipo de dato y objeto, de la ontología del dominio. Para este propósito se ha aprovechado el *framework* Apache Jena. Apache Jena es un *framework* Java para la construcción de aplicaciones de Web semántica y de datos enlazados, el cual integra diversas APIs para interactuar con grafos *RDF* y ontologías *OWL* y razonar sobre datos de instancias y descripciones de clases.

Trabajar con ontologías en Apache Jena (*API Ontology*) supone crear modelos representados por la clase “OntModel” para extender al modelo básico representado por la clase “Model”, la cual permite acceder a colecciones de tripletas *RDF* representadas a su vez por la clase “Statement”, añadiendo así soporte para las construcciones específicas de un lenguaje, como *RDFS* o *OWL*, necesarias para ello: clases, propiedades e individuos. En esta investigación se crea un modelo Jena de ontología a partir de un grafo de la ontología del dominio (serializado en formato RDF/XML) almacenado en un almacén de grafos *RDF* nombrados (para más detalles acerca de esto ver la subsección 3.5 de este capítulo), y se añade soporte para el lenguaje *OWL*, específicamente para el subconjunto OWL-DL. Asimismo, no se añade soporte para algún razonador, por las razones que se explicarán más adelante en este capítulo.

En detalle, por un lado, es necesario crear una instancia de una subclase concreta de la clase “FoodEstablishment” y una instancia de la clase “PostalAddress” por cada resultado provisto por el servicio o recurso Web de consulta de detalles sobre un establecimiento de alimentos y bebidas particular. Asimismo, es necesario asociar dichas instancias a través de la propiedad de objeto “hasPostalAddress”, cuyo dominio y rango es la clase “FoodEstablishment” y la clase “PostalAddress”, respectivamente. El tipo concreto de la instancia que representa al establecimiento se define a partir de las categorías y subcategorías de establecimiento de alimentos y bebidas y los estilos de cocina asociados a dicho establecimiento, los cuales son igualmente provistos por el servicio o recurso Web antes mencionado.

Ciertos datos relacionados con el establecimiento propiamente dicho, a saber, sus datos de identificación y sus niveles relativos de precios, se toman como valores de propiedades de dato cuyo dominio es la clase "FoodEstablishment"; las ubicaciones geográficas (coordenadas latitud y longitud) y ciertos componentes de las direcciones postales asociadas a los establecimientos se toman como valores de propiedades de dato cuyo dominio es la clase "PostalAddress".

En este punto es importante mencionar que cada clase y propiedad de dato y objeto en la ontología se ha mapeado a un objeto o atributo correspondiente (formalmente al nombre de este) en una o más de las *APIs* de redes sociales y servicios basados en localización que agrega. Para ello se han utilizado axiomas de anotación, en los que, vistos como tripletas *RDF*, los objetos son literales del tipo de dato *string* que representan a los nombres de los objetos y los atributos. Cabe mencionar que, con el fin de enlazar distintos vocabularios, esta práctica es común en el ámbito biomédico. Por ejemplo, la ontología "Diabetes Ontology", la cual fue creada recientemente como resultado del proyecto europeo BioMedBridges, se basa en la terminología definida por "SNOMED Clinical Terms" (SNOMED-CT). SNOMED-CT es una terminología médica procesable por computadora desarrollada por la organización sin ánimo de lucro SNOMED International; actualmente es considerada la más amplia y precisa en el mundo.

En concreto, en esta investigación se han creado las propiedades de anotación "foursquareName", "yelpName" y "tripAdvisorName". Utilizando la súper-clase "OntModel" de la API *Ontology* de Apache Jena es posible recuperar, mediante un *IRI*, una propiedad de anotación de un modelo Jena; una vez obtenida la propiedad, es posible obtener sus valores en las afirmaciones de anotación, propiedades de objeto o instancias, de manera similar a como se hace con los de las propiedades de tipo de dato y objeto. Así, es posible determinar el tipo concreto de la instancia a crear para representar a un establecimiento de alimentos y bebidas en particular contrastando tanto una lista con las categorías y subcategorías de establecimiento asociadas a él, como una lista con sus estilos de cocina, con una lista con los valores de una de las propiedades de anotación antes mencionadas (según sea el caso) en las afirmaciones de anotación en las que, vistas como tripletas *RDF*, el sujeto es alguna subclase de la clase "FoodEstablishment".

De manera similar, por cada propiedad de dato cuyo dominio es la clase "FoodEstablishment" se recupera el valor de la propiedad de anotación "foursquareName", "yelpName" o "tripAdvisorName" (según sea el caso) en las afirmaciones de anotación en las que, vistas como tripletas *RDF*, el sujeto es dicha propiedad de dato. De esta manera, el nombre del objeto o atributo correspondiente (formalmente al nombre de este) en la *API* de la red social o servicio basados en localización subyacente se determina. El mismo procedimiento se aplica a las propiedades de dato cuyo dominio es la clase "PostalAddress".

Por otro lado, es necesario crear una instancia de una o más subclases concretas de la clase "Cuisine" para representar a las cocinas de cada establecimiento de alimentos y bebidas. Al igual que se crea una instancia de una subclase concreta de la clase "FoodEstablishment" para representar a un establecimiento propiamente dicho, los tipos concretos de las instancias a crear para representar a sus cocinas se determinan a partir de las categorías y subcategorías de establecimiento y de los estilos de cocina asociados al mismo en la *API* de red social o servicio basado en localización subyacente. Concretamente, se determinan contrastando la lista de categorías y subcategorías de establecimiento y la lista de estilos de cocina antes creadas, con una lista con los valores de la propiedad de anotación "foursquareName", "yelpName" o "tripAdvisorName" (según sea el caso) en las afirmaciones de anotación en las que, vistas como tripletas *RDF*, el sujeto es alguna subclase de la clase "Cuisine". Asimismo, es necesario asociar las instancias creadas a la instancia que representa al establecimiento de alimentos y bebidas correspondiente a través de la propiedad de objeto "servesCuisine", cuyo dominio y rango es la clase "FoodEstablishment" y la clase "Cuisine", respectivamente.

El mismo procedimiento se aplica para la creación de una o más subclases concretas de la clase “Dish” para representar a los platos servidos por cada establecimiento de alimentos y bebidas. Las instancias creadas se deben asociar a la instancia que representa al establecimiento de alimentos y bebidas correspondiente a través de la propiedad de objeto “servesDish”, cuyo dominio y rango es la clase “FoodEstablishment” y la clase “Dish”, respectivamente.

3.5. Persistencia de conocimiento semántico

Las instancias de las clases de la ontología del dominio (*OWL*) creadas bajo demanda como resultado de esta etapa se almacenan en un almacén de grafos *RDF* nombrados (*quadstore*), específicamente como tripletas *RDF* en un segundo grafo nombrado serializado en formato RDF/XML. A diferencia de otros trabajos del grupo de investigación, en esta tesis doctoral se ha optado por esta solución, en contraposición al mapeo a una base de datos relacional o incluso al uso de ficheros en un sistema de archivos, con el objetivo de aprovechar las capacidades de razonamiento provistas por este tipo de base de datos (por clasificarlos de alguna manera). De esta manera es posible, además, poner a disposición de otras personas o aplicaciones la base de conocimiento creada como resultado, específicamente a través de un *endpoint SPARQL*, lo que representa uno de los principios del método de publicación de datos estructurados *Linked Data*, que se implementa en esta investigación.

En detalle, el almacén de grafos *RDF* nombrados utilizado es OpenLink Virtuoso, la versión *open source* del servidor multiplataforma y escalable Virtuoso Universal Server, el cual es mantenido por la compañía de software OpenLink Software. En realidad, OpenLink Virtuoso combina funciones de gestión de datos relacionales, grafos (almacenamiento nativo) y documentos con funcionalidades de servidor Web y de plataforma de servicios Web.

Para la comunicación entre Apache Jena y OpenLink Virtuoso se ha utilizado el proveedor de datos “Virtuoso Jena RDF Data Provider”, definido como “un proveedor completamente funcional de almacenamiento nativo de modelos de grafos”, así como el *driver* Virtuoso *JDBC* (*Java DataBase Connectivity*). Como resultado, es posible acceder directamente al almacén de grafos *RDF* de Virtuoso desde Apache Jena, tanto para extraer datos almacenados como para almacenar nuevos datos mediante consultas estáticas basadas en los protocolos y lenguajes de consulta para *RDF*, *SPARQL* y *SPARQL Update*, respectivamente; alternativamente, es posible utilizar el motor de consultas para Apache Jena, ARQ, el cual soporta los protocolos y lenguajes de consulta antes mencionados. En esta investigación se ha optado por el primer mecanismo, con el objetivo de habilitar el uso de cualquier extensión sobre *SPARQL* y *SPARQL Update* específica de OpenLink Virtuoso, de hecho, en esta investigación se ha usado la operación INSERT DATA de *SPARUL Update*.

Dado que, los modelos Jena de ontologías no alteran la representación subyacente basada en *RDF*, sino que agregan clases y métodos útiles para manipular dicha representación de manera más sencilla, es posible obtener, mediante el uso de la clase “Statement”, las tripletas *RDF* subyacentes a las instancias en un modelo Jena de ontología *OWL* para su persistencia en un almacén de tripletas *RDF* (*triplestore*) o en un almacén de grafos *RDF* nombrados (*quadstore*), como es el caso de esta investigación.

En este punto es importante mencionar que se ha propuesto el aprovechamiento de las capacidades de razonamiento de la implementación *SPARQL* de OpenLink Virtuoso con el objetivo de utilizar los lenguajes *RDFS* (subconjunto de implicaciones *RDFS*) y *OWL* (subconjunto de construcciones OWL-Lite) para inferir dinámicamente o “al vuelo” hechos adicionales, es decir, tripletas *RDF* que representan conocimiento semántico no explícito, a partir de las tripletas *RDF* físicamente almacenadas en un grafo *RDF* nombrado y una serie de axiomas asociados. En detalle, Virtuoso OpenLink soporta las propiedades de *RDFS* “subClassOf” y “subPropertyOf” (relaciones de subclase y sub-propiedad), así como las propiedades de OWL 1

"equivalentClass" (axiomas de clase), "equivalentProperty" (axiomas de propiedad) y "sameAs" (axiomas de identidad de instancias).

Para ello, se debe emplear un mecanismo denominado "contexto de inferencia". Dicho mecanismo consiste en la creación de un grafo nombrado para los axiomas (grafo de esquema), llamado "conjunto de datos de reglas", y la asociación programática de dicho grafo al grafo compuesto por los datos de las instancias. Dicha asociación se consigue, o bien agregando el llamado "pragma de reglas de inferencia de virtuoso" a las consultas destinadas a inferir hechos adicionales acerca de las instancias, o usando un enfoque de consulta basado en los operadores de rutas de propiedades definidos por el lenguaje de consulta *SPARQL*. Como ya se esbozó en la subsección 3.4.2 de este capítulo, en esta investigación se ha creado un grafo para la ontología propiamente dicha (serializada en formato RDF/XML), la cual comprende solo un componente terminológico; esto se ha realizado utilizando la herramienta Virtuoso Conductor, la herramienta Web de administración provista por default con cualquier instancia del servidor OpenLink Virtuoso. Asimismo, se ha optado por el uso del pragma de reglas de inferencia en las consultas; una práctica bastante extendida entre los *endpoints SPARQL* basados en Virtuoso Universal Server.

En principio, se ha propuesto utilizar los axiomas de clase basados en la propiedad de *OWL* "equivalentClass" definidos en la ontología del dominio y, por lo tanto, en el grafo de esquema almacenado en OpenLink Virtuoso, para obtener los tipos inferidos de las instancias de las subclases de la clase "FoodEstablishment" que se crean programáticamente utilizando Apache Jena. Al habilitar dicho razonamiento al nivel del repositorio de conocimiento de la arquitectura propuesta, se asegura que la base de conocimiento, que, como se mencionó previamente, se pone a disposición de otras personas o aplicaciones a través de un *endpoint SPARQL*, consiste no solamente en tripletas *RDF* declaradas sino también en tripletas inferidas.

3.6. Descubrimiento de tópicos basado en el modelo Latent Dirichlet Allocation

La siguiente etapa en el método de recomendación propuesto es el descubrimiento de tópicos en un corpus derivado de los *check-ins* (instancias de la clase "CheckIn" en el repositorio de conocimiento) hechos por los usuarios, utilizando una técnica basada en el modelo probabilístico generativo de tópicos *LDA*.

Como se describió en la sección 3.3 de este capítulo, en el dominio del sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas existen, además de usuarios, personas que, sin tener dicho rol, participan indirectamente en los procesos de recomendación, a saber, personas con el rol de acompañante de usuario en los *check-ins* hechos en los establecimientos. No obstante, en este punto es importante mencionar que los usuarios a su vez se pueden clasificar en usuarios activos y usuarios pasivos, siendo los primeros aquellos para quienes se calculan las recomendaciones, y los últimos aquellos de quienes se toma la información contextual histórica (*check-ins* hechos en los establecimientos) a partir de la cual es posible construir un modelo *LDA* que sirva de base para recomendaciones futuras. En general, de aquí en adelante, se utiliza el término simple "usuario" para referirse al rol de usuario pasivo, mientras que para referirse al rol de usuario activo se utiliza, precisamente, el término "usuario activo". La Figura 3.3 muestra un diagrama de casos de uso (notación *UML*) que representa la jerarquía de usuarios en el dominio del sistema.

Para ello, primero es necesario inferir información contextual de alto nivel (valores de propiedades de tipo de dato cuyo dominio es la clase "CheckIn") a partir de cierta información contextual de bajo nivel asociada a los *check-ins* (instancias de la clase "CheckIn"). En detalle, los tipos de información contextual de alto nivel a inferir son: (1) información social, a saber, las situaciones sociales (la propiedad de dato "visitedInSocialSituation") de los usuarios durante las visitas a los establecimientos, las cuales se infieren a partir de los tipos de relaciones sociales de estos con las personas que los acompañaron durante las visitas y (2) información temporal, a saber, los periodos del día y los días de la semana (las propiedades de tipo de dato "visitedAtPeriodOfDay" y

"visitedAtDayOfWeek") en los que los usuarios, junto con sus acompañantes, visitaron los establecimientos; dicha información se infiere a partir de los *timestamps* de los *check-ins* en sí.

Intuitivamente, dicho proceso de inferencia, a diferencia del proceso de inferencia basada en *RDFS* y *OWL* descrito en la sección anterior, está fuera del alcance de las capacidades de inferencia habilitadas por las ontologías, y más bien corresponde al enfoque de inferencia basada en conjuntos de reglas. En este contexto, se ha propuesto emplear la notación *SPIN*, formalmente una representación *RDF* del lenguaje de consulta *SPARQL*, a fin de definir dichos conjuntos de reglas utilizando el mismo medio utilizado en el proceso de inferencia basada en *RDFS* y *OWL*, esto es el protocolo y lenguaje de consulta *SPARQL*.

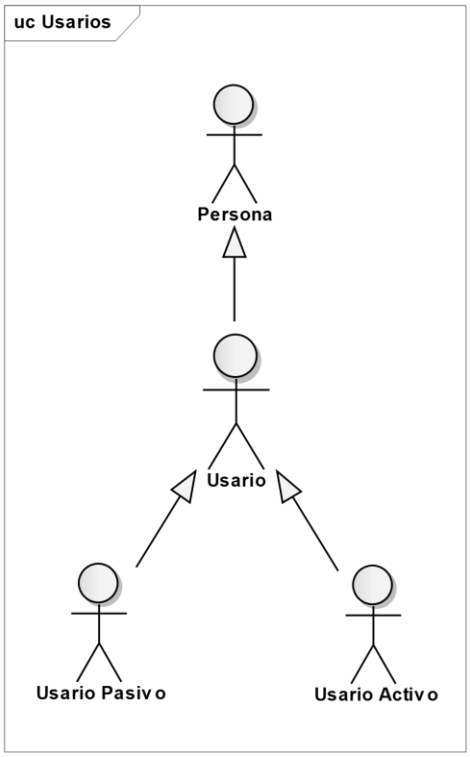


Figura 3.3. Actores en el dominio del sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas.

Hasta donde el autor conocía a fecha del desarrollo de esta tesis doctoral, la versión *open-source* del servidor Virtuoso Universal Server no soportaba la notación *SPIN*; por lo que se optó por realizar el proceso de inferencia antes descrito al nivel del componente integrador de conocimiento de la arquitectura propuesta. En otras palabras, se propuso integrar la notación *SPIN* con el framework Apache Jena. Es importante mencionar que, a efectos prácticos, esto es explicado en este punto de este capítulo.

Concretamente, se utilizó la biblioteca Java *open-source* desarrollada por la compañía de software TopQuadrant, Inc., TopBraid SPIN API, la cual de hecho está basada en el *framework* Apache Jena. La biblioteca TopBraid SPIN API no solo provee un motor de inferencia basado en reglas para *SPIN*, sino también un motor de comprobación de restricciones para dicha notación, siendo dos de las funciones principales de las propiedades del vocabulario basado en *RDF* definido por la misma, dejando de lado la definición de plantillas y funciones.

Para ello, se ha creado una regla *SPIN* basada en la expresión CONSTRUCT del lenguaje de consulta *SPARQL*, esto es, una regla de inferencia, para inferir el valor de la propiedad "visitednSocialSituation" y, al mismo tiempo, el valor de la propiedad "visitedAtPeriodOfDay" para cada instancia de la clase "CheckIn" en el repositorio de conocimiento (ver la Figura 3.4).

```
my:CheckIn
     a       owl:Class ;
     rdfs:subClassOf owl:Thing ;
     spin:rule
           [ a       sp:Construct ;
             sp:text """ CONSTRUCT {
   ?this my:visitedInSocialSituation ?ss .
   ?this my:visitedAtPeriodOfDay ?pd .
}
WHERE {
   {
      ?this my:hasUserCompanion ?companion .
      ?companion my:isFriendOf ?user .
      ?this my:timestamp ?dateTime .
      BIND ("friends" AS ?ss) .
      BIND (hours(?dateTime) AS ?hours) .
      BIND (IF(((?hours > 6) && (?hours < 12)), "morning", IF(((?hours > 12) && (?hours < 2), "lunchtime", […]
)) AS ?pd) .
   }
   UNION
   {
      ?this my:hasUserCompanion ?companion .
      ?companion my:isRelativeOf ?user .
      ?this my:timestamp ?dateTime .
      BIND ("family" AS ?ss) .
      BIND (hours(?dateTime) AS ?hours) .
      BIND (IF(((?hours > 20) && (?hours < 24)), "morning", IF(((?hours > 12) && (?hours < 2), "lunchtime",
[…] )) AS ?pd) .
   }
[…] .
}
```

Figura 3.4. Extracto de la regla *SPIN* para inferencia de información contextual de alto nivel.

En detalle, se ha empleado un patrón de unión (el operador UNION) dentro de la expresión WHERE de *SPARQL*, de modo que se construye un grafo por cada posible tipo de relación social entre un usuario y un acompañante (las propiedades de objeto "hasFriend", "hasRelative", "hasSignificantOther", "hasSpouse" y "hasCoWorker"), combinándose finalmente en único grafo de acuerdo al patrón indicado en la expresión CONSTRUCT. Intuitivamente, cada uno de estos grafos comprende más de una tripleta *RDF*; de hecho, también incluye una tripleta en la que el predicado es la propiedad de dato "timestamp" y el objeto es una variable de la cual se extrae posteriormente la hora utilizando la función "xsd:Integer HOURS(xsd:DateTime args)" de *SPARQL*. Dicha hora se asigna después a una segunda variable utilizando la expresión BIND, también de *SPARQL*, y posteriormente se evalúa utilizando una serie de sentencias IF anidadas, para asignar, a una tercera variable, la cual es finalmente referenciada en la expresión CONSTRUCT, la palabra que representa al periodo

del día correspondiente (a saber, las palabras “morning”, “lunchtime”, “afternoon”, “evening” y “night”). De manera similar, la palabra que representa a la situación social correspondiente al tipo de relación social (a saber, las palabras “friends”, “family”, “couple”, “married couple” y “co-workers”) se asigna a otra variable que también se referencia en la expresión CONSTRUCT.

Básicamente, utilizar la *API* TopBraid SPIN API se resume a utilizar la clase “SPINInferences”, la cual permite ejecutar reglas de inferencia mediante el método “run”, para lo que requiere, principalmente, un modelo Jena con las reglas de inferencia (modelo base), el cual evidentemente debe contener una ontología (axiomas *RDFS* u *OWL*) serializada en algún formato de serialización de RDF y un segundo modelo vacío donde almacenar las tripletas RDF a inferir.

La regla como tal se añadió al fichero de la ontología del dominio utilizando la herramienta gratuita TopBraid Composer. TopBraid Composer es un entorno de desarrollo de aplicaciones de Web Semántica, cuya versión gratuita se basa en la plataforma de software Eclipse. En dicha herramienta, es posible importar ficheros con extensión “rdf” para asociar reglas *SPIN* a axiomas de clase mediante la propiedad “rule” de *SPIN* (spin:rule), esto a través de una interfaz de usuario intuitiva. El fichero resultante realmente remplazó al fichero original de la ontología, por lo que fue necesario remplazar también el correspondiente grafo nombrado en OpenLink Virtuoso. Por otro lado, dado que en esta investigación los datos de las instancias se almacenan separadamente como tripletas *RDF* (formato RDF/XML) en un segundo grafo nombrado en OpenLink Virtuoso, la ejecución de la regla de inferencia antes descrita requiere la previa construcción de un modelo Jena de ontología (el modelo base) a partir de los resultados de la ejecución de una consulta *SPARQL* basada en la expresión CONSTRUCT, la cual permita unir ambos grafos.

En lo que respecta a los valores de la propiedad “visitedAtDayOfWeek”, dado que ni *SPARQL* ni *SPIN* proveen una función semejante a la función “HOURS”, que permita obtener un día de la semana a partir de un literal del tipo de dato “dateTime” de XMLSchema, se ha propuesto usar la clase Calendar de la *API* Util de Java para obtener dichos valores directamente de los valores de la propiedad “timestamp” en las tripletas *RDF* en el modelo inferido, una vez que este modelo se ha obtenido.

La Figura 3.5 presenta un ejemplo de inferencia de información contextual de alto nivel a partir de la información contextual de bajo nivel asociada a la instancia de la clase “CheckIn” identificada por el IRI “CheckIn1”. Se emplea un formato textual con el objetivo de introducir al lector en el proceso de construcción del modelo *LDA*, el cual se describe en detalle a continuación, y proveer una visión general de la estructura de los documentos en dicho modelo.

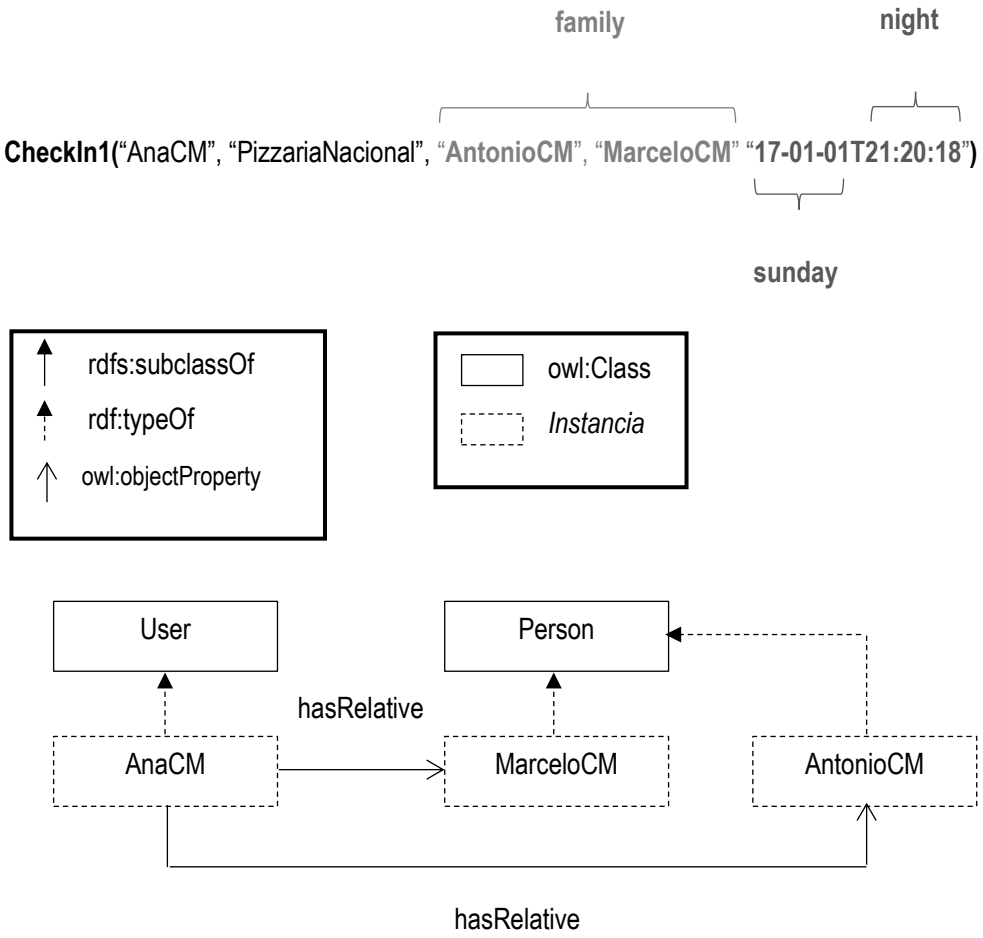


Figura 3.5. Ejemplo de inferencia de información contextual de alto nivel.

Así, en la Figura 2, la palabra "AnaCM" representa el *IRI* que identificada a la instancia de la clase "User" asociada a la instancia "CheckIn1" (a través de la propiedad de objeto "hasVisitor"). De manera similar, la palabra "PizzariaNacional" representa el *IRI* que identifica a la instancia de la clase "FoodEstablishment" asociada a la instancia "CheckIn1" (a través de la propiedad de objeto "hasVisitedPlace"). Las palabras "MarceloCM" y "AntonioCM" representan *IRIs* que identifican a instancias de la clase "Person" asociadas a la instancia "CheckIn1" (a través de la propiedad de objeto "hasUserCompanion"). La palabra "17-01-01T21:20:18" representa el valor del literal del tipo de dato "dateTime" de XML Schema asociado a dicha instancia (a través de la propiedad de tipo de dato "timestamp").

Por otro lado, la palabra "family" en dicha figura representa el valor del literal del tipo de dato "string" que representa la situación social inferida (la propiedad de tipo de dato "visitedInSocialSituation"). Las palabras "sunday" y "night" representas valores de literales del tipo de dato "string" que representan respectivamente el día de la semana y el periodo del día inferidos (las propiedades de tipo de dato "visitedAtDayOfWeek" y "visitedAtPeriodOfDay", respectivamente). A fin de facilitar el entendimiento del proceso de inferencia ejemplificado, se incluye también un extracto del perfil de usuario del usuario de interés, específicamente, sus relaciones sociales.

Una vez inferida la información contextual de alto nivel de interés, es posible llevar a cabo la construcción del modelo *LDA*. En dicho modelo, valores de atributos obtenidos a partir de dicha información se interpretan como

las palabras, mientras que los documentos se corresponden con los historiales de los *check-ins* ocurridos en cada lugar (descritos por dichas palabras). En este punto cabe mencionar que con esta técnica se asume que grupos de variables (valores de atributos contextuales de alto nivel) observadas en los eventos de *check-in* se pueden explicar a partir de grupos de variables (tópicos) no observadas en dichos eventos, y que, finalmente, existe una relación de coocurrencia significativa entre dichos valores interpretados como palabras. Esta relación de coocurrencia es la que permite recomendar los establecimientos de alimentos y bebida que mejor se acercan a las preferencias de los usuarios dentro de contextos socio-temporales específicos, esto es, tópicos.

La Tabla 3.1 presenta los posibles valores de los atributos contextuales de alto nivel usados como palabras en la construcción del modelo *LDA*.

Ontología del Dominio		**Modelo LDA**	
I.C. de Bajo Nivel (Declarada)	**I.C. de Alto Nivel (Inferida)**	**Atributo**	**Valor (Palabra)**
-hasUserCompanion (propiedad de objeto) -isFriendOf, isRelativeOf, isSignificantOtherOf, isSpouseOf, y isCoWorkerOf (propiedades de objeto, dominio clase "Person")	visitedInSocialSituation (propiedad de tipo de dato)	SocialSituation	{"friends", "family", "couple", "married_couple", "co-workers"}
timestamp (propiedad de tipo de dato)	visitedAtPeriodOfDay (propiedad de tipo de dato)	PeriodOfDay	{"morning", "lunchtime", "afternoon", "evening", "night"}
	visitedAtDayOfWeek (propiedad de tipo de dato)	DayOfWeek	{"sunday", "monday", "tuesday", "thursday", "friday", "saturday"}

Tabla 3.1. Atributos contextuales de alto nivel usados en la construcción del modelo *LDA* (I.C.= Información Contextual).

Para la construcción propiamente dicha del modelo *LDA* se ha utilizado el algoritmo de *clustering* de documentos basado en *LDA* implementado en la biblioteca Java LingPipe (la clase estática "LatentDirichletAllocation"). Formalmente LingPipe es un *toolkit* Java para el procesamiento de texto mediante técnicas de lingüística computacional. Dicha implementación permite principalmente: (1) estimar los parámetros de un modelo *LDA* a partir de un corpus no etiquetado de documentos dadas, por un lado, las dos distribuciones de probabilidad de Dirichlet a priori y, por otro lado, un número predefinido de tópicos y (2) construir un modelo *LDA* a partir de una de las muestras de la estimación, las cuales se obtienen mediante el algoritmo de muestreo de Gibbs.

A efectos prácticos, en esta tesis doctoral se ha propuesto utilizar distribuciones de Dirichlet simétricas, específicamente se han utilizado los parámetros de concentración (fijos) $\alpha=0.1$ y $\beta=0.01$ (distribución tópico-palabra y distribución documento-tópico). Cabe señalar que las investigaciones en el campo de *LDA* usan comúnmente este enfoque en la elección de las distribuciones de probabilidad de Dirichlet a priori. No obstante, existen trabajos que evidencian que el uso de distribuciones de Dirichlet asimétricas sobre las distribuciones tópico-palabra provee ventajas reales en términos del rendimiento de los modelos, recomendando el uso de este tipo de distribución de Dirichlet en combinación con el tipo simétrico para el caso de la distribución documento-tópico (Wallach, Mimno, & McCallum, 2009). Estos trabajos demuestran, adicionalmente, que la

optimización de los parámetros de concentración usando algoritmos simples pero eficientes de inferencia hace incluso mayores dichas ventajas. Como se verá al final de este documento, este aspecto ha sido considerado una futura línea de investigación de esta tesis doctoral.

En esta investigación se ha puesto mayor énfasis en la elección del número de tópicos (*k*) a utilizar, de modo que se ha propuesto calcular dinámicamente dicho parámetro empleando el método de análisis de estabilidad propuesto por Greene, O'Callaghan, & Cunningham (2014). Esté método permite calcular la similitud entre un modelo *LDA* generado a partir de un corpus completo de documentos (modelo de referencia) y una serie de modelos de prueba generados a partir de diferentes subconjuntos de documentos del mismo corpus (modelos de prueba). La similitud entre cada par de modelos se calcula como el nivel de correspondencia, dado por el coeficiente de similitud de Jaccuard, entre las *n* palabras con mayor probabilidad en sus tópicos, donde un nivel de correspondencia alto significa un grado mayor de similitud. La repetición de este procedimiento para un rango de valores tentativos de *k* (número de tópicos a usar) eventualmente permite determinar la mejor opción dentro de dicho rango de valores.

Como es posible inferir, este método requiere la construcción previa de los modelos *LDA* a analizar; para ello, se ha usado el algoritmo paralelizado mostrado en la Figura 3.6, el cual, como se puede observar, requiere tres parámetros de entrada distintos: (1) el intervalo de valores tentativos de *k* (intervalo de números enteros), el cual indica el número de modelos de referencia a generar, es decir, un modelo de referencia por cada valor tentativo de *k*, (2) el número de modelos de prueba a generar por cada modelo de referencia y (3) el porcentaje de documentos en el corpus a usar como subconjunto de documentos en la generación de los modelos de prueba.

```
static String[] WORDS = new String[] {
        "friends", "family", "couple", "married_couple", "co-workers", "morning", "lunchtime", "afternoon",
        "evening", "night", "sunday", "monday", "tuesday", "thursday", "friday", "saturday"
};

static SymbolTable SYMBOL_TABLE = new MapSymbolTable();
        static {
                for (String word : WORDS)
                        SYMBOL_TABLE.getOrAddSymbol(word);
}

static double DOC_TOPIC_PRIOR = 0.1;
static double TOPIC_WORD_PRIOR = 0.01;

static int BURNIN_EPOCHS = 15;
static int SAMPLE_LAG = 1;
static int NUM_SAMPLES = 16;

short right;
short left;
short nTModels;
File corpus;
short pDocuments;

LatentDirichletAllocation[] rModels;
LdaRunnable[][] tModels;
```

```
Thread[][] tModelsThreads;

LDA(short mRight, short mLeft, short mNTM, File mCorpus, short mPDocuments) {
	right = mRight;
	left = mLeft;
	nTModels = mNTM;

	corpus = mCorpus;
	pDocuments = mPDocuments;

	rModels = new LatentDirichletAllocation[right-1];
	tModels = new LdaRunnable[right-left+1][nTModels];
	tModelsThreads = new Thread[right-left+1][nTModels];
}

public LatentDirichletAllocation getLda(LatentDirichletAllocation.GibbsSample sample) {
	return sample.lda();
}

public void sample () {
	short i;
	short j;
	CharSequence [] documents;

	for(i=0; i<right-left+1; i++) {
		documents = Corpus.readCorpus((short) 100);
		rModels[i] =
		getLda(LatentDirichletAllocation.gibbsSampler(Corpus.tokenizeDocuments(documents,
		SYMBOL_TABLE), (short) (left+i), DOC_TOPIC_PRIOR, TOPIC_WORD_PRIOR,
		BURNIN_EPOCHS, SAMPLE_LAG, NUM_SAMPLES, new Random(), new
		LdaReportingHandler(SYMBOL_TABLE))
		);

	for(j=0; j<nTModels; j++) {
		documents = Corpus.readCorpus(pDocuments);
		tModels[i][j] = new LdaRunnable(Corpus.tokenizeDocuments(documents,
		SYMBOL_TABLE), (short) (j+1), new Random(), new
		LdaReportingHandler(SYMBOL_TABLE));
	}
}

for(i=0; i<right-left+1; i++)
	for(j=0; j<nTModels; j++)
		tModelsThreads[i][j] = new Thread(tModels[i][j]);

for(i=0; i<right-left+1; i++)
	for(j=0; j<nTModels; j++)
		tModelsThreads[i][j].Stuart();

for(i=0; i<right-left+1; i++)
	for(j=0; j<nTModels; j++)
```

```
            try {
                tModelsThreads[i][j].join();
            } catch (InterruptedException ex) {
                Logger.getLogger(LDA.class.getName()).log(Level.SEVERE, null, ex);
            }
}

static class LdaRunnable implements Runnable {
            LatentDirichletAllocation mLda;
            final int[][] mDocWords;

            final ObjectHandler<LatentDirichletAllocation.GibbsSample> mHandler;
            final Random mRandom;

            LdaRunnable(int[][] docWords, i, Random random,
            ObjectHandler<LatentDirichletAllocation.GibbsSample> handler) {
                mDocWords = docWords;
                mI = i;
                mRandom = random;
                mHandler = handler;
            }

            public void run() {
                mLda = sample(mDocWords, mI, mRandom, mHandler);
            }

            public LatentDirichletAllocation sample(int[][] mDocWords, short mI, Random mRandom,
            ObjectHandler<LatentDirichletAllocation.GibbsSample> mHandler) {
                LatentDirichletAllocation.GibbsSample sample;
                sample = LatentDirichletAllocation.gibbsSampler(mDocWords, mI, DOC_TOPIC_PRIOR,
                TOPIC_WORD_PRIOR, BURNIN_EPOCHS, SAMPLE_LAG, NUM_SAMPLES, new
                Random(), new LdaReportingHandler(SYMBOL_TABLE)
                );
                return sample.lda();
            }
}
```

Figura 3.6. Algoritmo para generación de modelos LDA para análisis de estabilidad.

Cabe mencionar que, como se puede apreciar en la Figura 3.6, el método "gibbsSampler" de la clase "LatentDirichletAllocation" de la biblioteca LingPipe, requiere la representación de los documentos como secuencias de identificadores numéricos almacenadas en un *array* bidimensional. En dicho array, cada posición i_j, representa un documento, y cada posición i_{jk} hace referencia a una instancia de una palabra, es decir, un *token*, representada por un identificador numérico. La conversión requiere, básicamente, el uso de un mapa de símbolos (la clase "MapSymboTable" de la misma biblioteca), el cual permite asociar símbolos, esto es, palabras, a identificadores numéricos.

En esta investigación, no es necesario el uso previo de *tokenizadores* para la conversión de texto bruto procedente de un corpus en secuencias de *tokens*. Esto se debe a que el corpus (archivo de texto) se construye a medida a partir de la información contextual inferida (ver Tabla 3.2), conteniendo directamente *tokens*. En este contexto, se ha desarrollado una clase estática denominada "Corpus", la cual permite leer un subconjunto

aleatorio de documentos *tokenizados* de un corpus y realizar la conversión de *tokens* a identificadores numéricos (a partir de un mapa de símbolos). Los detalles de esta clase, así como los de la clase "LdaReportingHandler", la cual también se ha construido como parte de esta tesis doctoral, no son presentados en este documento. Basta decir que dicha clase implementa la interfaz "ObjectHandler<GibbsSample>" de la biblioteca LingPipe (el método "handle"), y que el método "gibbsSampler" requiere como parámetro un objeto de una clase que implemente dicha interfaz, a fin de que el método "handle" actúe como función de devolución de llamada. Asimismo, el método "gibbsSampler" requiere tres parámetros de configuración del proceso de muestreo: el número de iteraciones del periodo de quemado, la frecuencia de muestreo y el número total de muestras (BURNIN_EPOCHS, SAMPLE_LAG, NUM_SAMPLES); para más detalles acerca de estos parámetros, referirse a la documentación en línea de la biblioteca LingPipe.

Una vez construidos los modelos *LDA*, es posible realizar el análisis en sí, para ello se ha implementado un algoritmo, cuyo núcleo se representa a continuación (Fórmula 3.1) mediante la notación sigma y la notación de teoría de conjuntos (coeficiente de similitud de Jaccard).

$$Similarity(rModel_{\#k}, tModel_{\#k}, n) = \frac{1}{\#tModels * \#k^2} \sum_{i=1}^{\#tModels} \sum_{j=1}^{\#k} \sum_{k=1}^{\#k} Jacqard(sub(rModel[j], n), sub(tModel[i][k], n))$$

$$Jacqard(rModel[j], tModel[i][k]) = \frac{rModel[j] \cap tModel[i][k]}{rModel[j] \cup tModel[i][k]} \quad (3.1)$$

Donde:

- $rModel_{\#k}$ representa un modelo *LDA* de referencia generado para un valor en el rango de valores tentativos de *k* (número de tópicos) y $tModel_{\#k}$ representa el conjunto de modelos *LDA* de prueba generados para el mismo valor de *k*.
- *#tModels* representa el número de modelos de prueba.
- *#k* representa el número de tópicos.
- *n* representa la profundidad de los *arrays* de palabras
- *sub(rModel[j],n)* representa el *array* de las *n* palabras con mayor probabilidad de ser asignadas al tópico *j* en el modelo de referencia *rModel*.
- *sub(tModel[i][k],n)* representa el *array* de las *n* palabras con mayor probabilidad de ser asignadas al tópico *k* en el modelo de prueba *tModel[i]*.

En detalle, la similitud entre un modelo de referencia *m* y un modelo de prueba *n* (con un número de tópicos *k*) para una profundidad *d* se calcula como el promedio de los coeficientes de similitud de Jaccard calculados para todos los pares de tópicos *i*, *j* en los que el tópico *i* proviene del modelo de referencia y el tópico *j* proviene de uno de los modelos de prueba correspondientes. Específicamente, se calcula como el promedio de los coeficientes de similitud de Jaccard entre los pares de conjuntos de palabras *i'*, *j'* en los que el conjunto de palabras *i'* representa las *d* palabras con mayor probabilidad de ser asignadas al tópico *i* y el conjunto de palabras *j'* representa las *d* palabras con mayor probabilidad de ser asignadas al tópico *j*. Así, el valor de *k* (número de tópicos) dentro de un rango de valores tentativos correspondiente al par de modelos con la similitud más alta se toma como el valor óptimo de *k*.

Una vez determinado el valor óptimo de *k*, el correspondiente modelo de referencia se toma como el modelo *LDA* definitivo. Para los efectos del perfilamiento de usuarios y el procesamiento de la información del contexto social/temporal del usuario, cuyos detalles se discuten en las siguientes dos secciones de este capítulo, es necesario calcular la distribución de probabilidad de los tópicos en los documentos (historiales de *check-ins*

ocurridos en los establecimientos de alimentos y bebidas, referidos de aquí en adelante solo como los establecimientos de alimentos y bebidas, para efectos prácticos) en los que se basa dicho modelo *LDA*. Para ello se debe calcular la proporción de: (1) la suma de la distribución documento-tópico a priori y el número de veces que el tópico se asignó al documento en el modelo y (2) la suma del número de documentos en los que se basa el modelo y el producto de la distribución documento-tópico a priori y el número de tópicos en el modelo (ver Fórmula 3.2). A efectos prácticos, por cada documento en el modelo *LDA*, el tópico con mayor probabilidad se asocia a la instancia de la clase de referencia que representa al establecimiento correspondiente en el repositorio de conocimiento (mediante la propiedad de tipo de dato "hasBestRankedTopic").

$$topicProbability(doc, t) = \frac{docTopicPrior + docTopicCount(doc, t)}{numDoc + docTopicPrior * k} \quad (3.2)$$

Donde:

- *t* es un tópico en el modelo *LDA*.
- *doc* es el documento para el cual se calcula la probabilidad del tópico *t*.
- *docTopicCount(doc,t)* es el número de veces que el tópico *t* se asignó al documento *doc*.
- *docTopicPrior* es la distribución documento-tópico a priori.
- *numDoc* es el número de documentos en los que se basa el modelo.
- *k* es el número de tópicos en el modelo.

Cabe mencionar que el número de documentos en los que el modelo se basa, así como el número de veces que el tópico se asignó al documento se pueden recuperar fácilmente a partir de un objeto de la clase estática "LatentDirichletAllocation.GibbsSample" (tipo de retorno del método "gibbsSampler" de la clase "LatentDirichletAllocation") de la biblioteca *LingPipe*. Específicamente, utilizando los métodos "numDocuments" y "documentTopicCount", respectivamente.

3.7. Perfilamiento de Usuarios basado en Ontologías

La siguiente etapa en el método de recomendación propuesto representa el punto de entrada del proceso de recomendación en sí. Esto se debe a que esta etapa consiste en la construcción y actualización constante de un perfil de características y preferencias por cada usuario. En detalle, las preferencias en los perfiles de usuario se actualizan a partir de los *ratings* explícitos e implícitos dados por los usuarios a los establecimientos durante las visitas a los mismos. Como resultado de los *ratings* explícitos e implícitos, la ontología del dominio o, mejor dicho, el repositorio de conocimiento, se puebla con instancias de las clases "Rating" y "CheckIn", respectivamente.

En este contexto, se ha considerado el número total de *check-ins* hechos por un usuario en un establecimiento como un indicador de la preferencia de ese usuario por ese establecimiento, específicamente, como el *rating* implícito de ese usuario para ese establecimiento. Dado que en las redes sociales y, en general, los servicios basados en localización, los usuarios a menudo hacen *check-in* en los establecimientos que visitan al momento de arribar a ellos y además proporcionan valoraciones a los mismos al momento de abandonarlos, en esta investigación se ha propuesto calcular la preferencia de un usuario por un establecimiento como la combinación de un *rating* explícito y un *rating* implícito (cuando ambos están disponibles).

En detalle, dicha combinación se ha calculado como la media aritmética ponderada del *rating* explícito dado por el usuario al establecimiento y un *rating* implícito calculado mediante la técnica estadística *Term Frequency-Inverse Document Frequency* (*TF-IDF*), como se muestra en la Fórmula 3.3. El uso de la técnica *TF-IDF* para el aprendizaje de las preferencias implícitas de los usuarios está inspirado en el trabajo de Bao, Zheng, & Mokbel (2012). En ese trabajo, se propone un sistema de recomendación de lugares basado en ubicación geográfica y sensible a preferencias, en el cual dichas preferencias se descubren a partir de las categorías de

los lugares en los historiales de ubicación de los usuarios y no de los lugares en sí, resultando en árboles de preferencias ponderados (un enfoque distinto al de esta tesis doctoral).

$$iRating(p,u) = \frac{|count(?\,c\ hasVisitor\ u\ \&\&\ ?\,c\ hasVisitedPlace\ p)|}{|count(?\,c, hasVisitor, u)|} * \log\frac{|count(?\,c\ hasVisitor\ ?\,u)|}{1 + |count(?\,c\ hasVisitor\ ?\,u\ \&\&\ ?\,c\ hasVisitedPlace\ p)|} \quad (3.3)$$

Donde:

- *p* es la instancia de la clase de referencia que representa al establecimiento de alimentos y bebidas para el cual se va a calcular el *rating* implícito.
- *u* es la instancia de la clase "User" que representa al usuario cuyo *rating* implícito va a ser calculado.
- *?c* representa cualquier instancia de la clase "CheckIn".
- *?u* representa cualquier instancia de la clase "User".
- *count(?c hasVisitor u && ?c hasVisitedPlace p)* es el número de instancias de la clase "CheckIn" asociadas al mismo tiempo a la instancia *u* y a la instancia *p*, a través de las propiedades de objeto "hasVisitor" y "hasVisitedPlace", respectivamente.
- *count(?c hasVisitor u)* es el número de instancias de la clase "CheckIn" asociadas a la instancia *u* a través de la propiedad de objeto "hasVisitor".

En detalle, la primera parte de la Fórmula 3.3 hace referencia a la medida *TF* del establecimiento *p* en el historial de *check-ins* del usuario *u*, es decir, a la proporción del número de *check-ins* asociados al usuario *u* y al mismo tiempo al establecimiento *p* y el número total de *check-ins* asociados a dicho usuario; la segunda parte de la fórmula hace referencia a la medida *IDF* del establecimiento *p* (en el conjunto de historiales de *check-ins* de usuarios existentes en el sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas), esto es, a la proporción del número total de *check-ins* asociados a un usuario cualquiera y el número de *check-ins* asociados a un usuario cualquiera y al establecimiento *p*. Cabe mencionar que se ha empleado la función de normalización "L2-norm" (normalización Euclidiana) para normalizar cada vector obtenido como resultado de repetir este procedimiento para todos los establecimientos visitados por cada usuario.

Por otro lado, como se puede inferir de la Fórmula 3.3, la notación utilizada para representar los términos count(?c hasVisitor u && ?c hasVisitedPlace p) y count(?c hasVisitor u) hace referencia a la sintaxis y semántica de las consultas *SPARQL* que, evidentemente, subyacen al aprendizaje de los ratings implícitos, específicamente a la función de agregación COUNT. De hecho, en esta tesis doctoral se ha diseñado una serie de consultas *SPARQL* basadas en la forma de consulta SELECT y la función COUNT, las cuales permiten aplicar dicha función al conjunto de la solución de los patrones *"?c my:hasVisitor my:u . ?c my:hasVisitedPlace my:p*", "*?c my:hasVisitor ?u . ?c my:hasVisitedPlace my:p*", "*?c my:hasVisitor my:u*" y "*?c my:hasVisitor ?u*". En dichos patrones, "u" se remplaza con el nombre (*IRI*) de la instancia de la clase "User" que representa al usuario cuyo *rating* va a ser calculado, "p" se remplaza con el nombre de la instancia de la clase "FoodEstablishment" que representa al establecimiento de alimentos y bebidas para el cual se va a calcular el *rating* y "*my*" es el prefijo definido para el espacio de nombres de la ontología del dominio.

El cálculo de la media aritmética ponderada que resulta en el *rating* compuesto que indica la preferencia de un usuario por un establecimiento se representa mediante la Fórmula 3.4.

$$cRating(p,u) = \frac{eRating(p,u) * 0{,}45 + \text{norm}\big(iRating(p,u)\big) * 0{,}55}{2} \quad (3.4)$$

Donde:

- *p* es la instancia de la clase de referencia que representa al establecimiento para el cual se va a calcular el *rating* compuesto.
- *u* es la instancia de la clase "User" que representa al usuario respecto al cual el *rating* compuesto para el establecimiento *p* va a ser calculado.
- *eRating(p,u)* representa el *rating* explicito dado por el usuario *u* al establecimiento *p*.
- *norm(iRating(p,u))* representa el valor normalizado del *rating* implícito calculado para el establecimiento *p* respecto al usuario *u* (usando la Fórmula 3.4).

Como se puede observar en la Fórmula 3.4, el peso para los *ratings* explícitos se ha establecido en el valor decimal 0,45 y el peso para los *ratings* implícitos en el valor decimal 0,55. De hecho, estos valores son relativos, lo que quiere decir que la suma de ellos es igual a la unidad, y se han obtenido por experimentación, en un esfuerzo por maximizar el balance entre la precisión y la exhaustividad (ver el siguiente capítulo de este documento).

Como parte del perfilamiento de usuarios se ha propuesto considerar también las preferencias de los usuarios por los tópicos en el modelo *LDA*. Para ello se ha propuesto emplear las distribuciones de probabilidad de los tópicos en los documentos, las cuales se obtienen gracias al uso de la Fórmula 3.2, y calcular la preferencia de un usuario *u* por un tópico *t* como la medida *TF* del tópico *t* en el historial de *check-ins* del usuario *u*. Específicamente, la preferencia de un usuario *u* por un tópico *t* se calcula como la proporción de: (1) el número de *check-ins* asociados al usuario *u* y al mismo tiempo a un establecimiento de alimentos y bebidas *p* (formalmente, al correspondiente documento en el modelo *LDA*) para el cual el tópico *t* es el tópico con mayor probabilidad y (2) el número total de *check-ins* asociados a dicho usuario (ver Fórmula 3.5).

$$preference_{u,t} = \frac{\big|\big(count(?\,c\ hasVisitor\ u\ \&\&\ ?\,c\ hasVisitedPlace\ ?\,p\ \&\&\ ?\,p\ hasBRTopic\ t)\big)\big|}{\big|count(?\,c\ hasVisitor\ u)\big|} \quad (3.5)$$

Donde:

- *t* es el valor que representa al identificador del tópico en el modelo *LDA* para el cual se va a calcular la preferencia.
- *u* es la instancia de la clase "User" que representa al usuario respecto al cual la preferencia por el tópico *t* va a ser calculada.
- *?c* representa cualquier instancia de la clase "CheckIn".
- ?p representa cualquier instancia de la clase de referencia.
- *count(?c hasVisitor u && ?c hasVisitedPlace p && ?p hasBRTopic ?t)* es el número de instancias de la clase "CheckIn" asociadas al mismo tiempo a la instancia *u* y a una instancia de la clase de referencia (a través de las propiedades de objeto "hasVisitor" y "hasVisitedPlace", respectivamente) asociada al valor que representa al identificador del tópico *t* (a través de la propiedad de dato "hasBestRankedTopic").
- *count(?c hasVisitor u)* es el número de instancias de la clase "CheckIn" asociadas a la instancia *u* a través de la propiedad de objeto "hasVisitor".

Como en el caso de la Fórmula 3.3, la notación utilizada para representar los términos de la Fórmula 3.5 hace referencia a la sintaxis y semántica de las consultas *SPARQL*, en este caso, consultas que subyacen al cálculo de las preferencias de los usuarios por los tópicos.

En este punto cabe aclarar lo siguiente. Como se vio en la sección anterior, la técnica basada en el modelo probabilístico generativo de tópicos *LDA* propuesta en esta tesis doctoral para el descubrimiento de tópicos a partir de información contextual histórica reside en conocimiento no explícito inferido a partir de las instancias de la clase "CheckIn"; de ahí que resulte primordial en esta investigación, no solo para el perfilado de usuarios, específicamente para la obtención de las preferencias del usuario, sino también para el descubrimiento de tópicos, la obtención de *ratings*, específicamente *ratings* implícitos. Sin embargo, no es posible obtener dicha retroalimentación por parte de los usuarios si no se producen recomendaciones y, como se explicará más adelante en este capítulo, no es posible producir recomendaciones si no se construyen previamente el modelo *LDA* y los perfiles de usuario.

Por lo tanto, se ha propuesto el siguiente enfoque alternativo en la obtención de las preferencias del usuario. Inicialmente, esto es, durante la construcción de cada perfil de usuario, las preferencias (por establecimientos) se deberán calcular únicamente a partir de *ratings* explícitos obtenidos mediante la aplicación de una técnica de propagación de preferencias en jerarquías de clases en perfiles de usuario basados en ontologías. De este modo, para los pares usuario-establecimiento para los que exista un *rating* explícito, pero no un *rating* implícito, el *rating* explícito por sí solo se considerará el *rating* final (*rating* compuesto), por lo que no se requerirá el uso de la Fórmula 3.4.

En detalle, dicha técnica de propagación de preferencias consiste, por un lado, en la presentación al usuario de un cuestionario dinámico de preguntas aleatorias destinadas a revelar sus preferencias por al menos cinco estilos de cocina o platos distintos; por otro lado, dichas preferencias se deben asignar a todas las instancias de subclases de la clase "FoodEstablishment" (existentes en el repositorio de conocimiento) que estén asociadas a las instancias de las subclases de la clase "Cuisine" y "Dish" que representan a los estilos de cocina y platos correspondientes. Cabe resaltar que dichos estilos de cocina y platos deberán ser, necesariamente, estilos de cocina y platos disponibles en la ciudad y país de residencia especificados por el usuario como parte de su información personal; dicha información personal deberá solicitarse antes de presentar el cuestionario de preferencias. De este modo se asegura que los establecimientos afectados por el algoritmo de propagación de preferencias son, únicamente, establecimientos existentes dentro del área geográfica demarcada por el país y la ciudad de residencia del usuario. Como en el caso de los *ratings* explícitos para los establecimientos, el repositorio de conocimiento se puebla con instancias de la clase "Rating" como resultado de los *ratings* explícitos para los estilos de cocina y platos.

En dicho cuestionario, las preguntas deberán hacer referencia a instancias de superclases en las jerarquías de clases cuyas clases raíz son las clases "Cuisine" y "Dish". Así, se asegurará la posibilidad de propagar las preferencias establecidas por el usuario desde las clases de dichas instancias hacia sus subclases en dicha jerarquía. Para ello se ha propuesto un algoritmo de propagación que penaliza las preferencias de acuerdo a la similitud taxonómica entre la clase raíz y la clase hija, de modo que entre más se descienda en la jerarquía menor sea dicha similitud y mayor sea la diferencia entre ellas y, por tanto, menor sea el valor del *rating* propagado. Este algoritmo está basado en el concepto de distancia semántica taxonómica propuesto por Sánchez, Batet, Isern, & Valls (2012), por lo que la similitud entre dos clases se ha calculado realmente como el complemento del valor racional (en el rango [0, 1]) que representa la diferencia entre ellas. La Fórmula 3.6 representa el núcleo de dicho algoritmo utilizando notación de teoría de conjuntos.

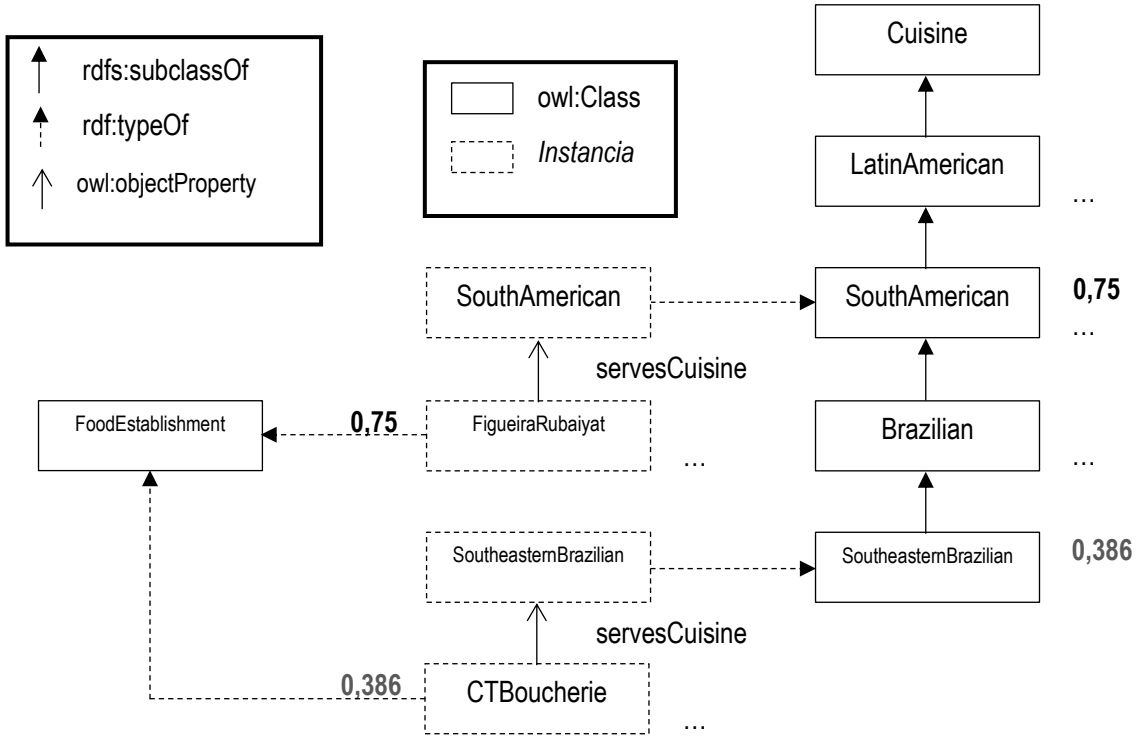
$$eRating(b, a, u) = eRating_{a,u} * tSimilarity(a, b)$$

$$tSimilarity(a, b) = 1 - \left(\log_2\left(1 + \frac{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)|}{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)| + |\emptyset(A) \cap \emptyset(B)|}\right)\right) \quad (3.6)$$

Donde:

- *u* es la instancia de la clase "User" que representa al usuario para el cual se está construyendo el perfil de usuario.
- *a* representa a una instancia de una superclase (representada como *A*) en la jerarquía de clases encabezada por la clase "Cuisine", o en la jerarquía de clases encabezada por la clase "DIsh", puntuada por el usuario *u* a través del cuestionario de preferencias.
- *b* representa a una instancia de una subclase de la clase *A* (representada como, *B*), a la cual se va a asignar el valor penalizado del *rating* dado a la instancia *a* y propagado desde la clase *A* ($rating_{a,u}$).
- *Ø(A)* representa al conjunto de características taxonómicas que describen a la clase *A* en términos de subsunción de conceptos.
- *Ø(B)* representa al conjunto de características taxonómicas que describen a la clase *B* en términos de subsunción de conceptos. Es importante mencionar que, estos términos se pueden obtener fácilmente utilizando el método "listSuperClasses*"* en instancias de la clase "OntClass" de la *API* Ontology de Apache Jena que representen a las clases *A* y *B*.
- *Ø(X)\ Ø(Y)* es la cardinalidad del conjunto de características taxonómicas diferenciales de la clase *X* respecto a la clase *Y*.
- *Ø(X) ∩ Ø(Y)* es la cardinalidad del conjunto de características taxonómicas comunes a las clases *X* y *Y*.

La Figura 3.7 muestra un ejemplo del uso del algoritmo de propagación de preferencias antes descrito. En dicha imagen se ejemplifica la valoración de la superclase "SouthAmerican" en la jerarquía de clases cuya clase raíz es la clase "Cuisine" por parte del usuario representado por la instancia de la clase "User" identificada por el *IRI* "AnaCM", y la propagación del *rating* otorgado (a saber, 0,75) a la subclase "SoutheasternBrazilian" (valor mostrado en color rojo). Asimismo, se muestra la asignación del valor penalizado de dicho *rating* a la instancia de la clase "FoodEstablishment" identificada por el *IRI* "CTBoucherie" (instancia asociada a la instancia de la clase "SoutheasternBrazilian" en el repositorio de conocimiento).



Donde:

- *u* = AnaCM
- *a* = SouthAmerican
- *b* = SoutheasternBrazilian
- *Ø(A)* = {SouthAmerican, LatinAmerican, Cuisine}
- *Ø(B)* = {SoutheasternBrazilian, Brazilian, SouthAmerican, LatinAmerican, Cuisine}
- $rating_{a,u}$ = 0,75

$$Similrity(SoutheasternBrazilian, SouthAmerican) = 1 - \left(\log_2\left(1 + \frac{|0| + |2|}{|0| + |2| + |3|}\right)\right)$$
$$= 1 - \left(\log_2(1 + 0{,}4)\right) = 1 - \left(\log_2(1{,}4)\right) = 1 - 0{,}485 = 0{,}515$$

eRating(CTBoucherie, FigueiraRubaiyat, AnaCM) = 0,75 * 0,515 = 0,386

Figura 3.7. Ejemplo de propagación de *ratings* explícitos obtenidos mediante el cuestionario de preferencias.

De manera similar, durante la construcción de cada perfil de usuario, las preferencias por los tópicos se deberán obtener directamente del usuario a través del cuestionario de preferencias. Esto quiere decir que, en ese caso no será necesario recurrir al uso de la Fórmula 3.5.

Además, como se explicará más a detalle en el siguiente capítulo de este documento, durante la fase de "entrenamiento" de la puesta en marcha del sistema, esto es, la fase de recolección de historiales de *check-ins* de usuarios destinada a la construcción del modelo *LDA* de información contextual de alto nivel, no será posible recurrir al uso de la Fórmula 3.5, y ni siquiera al uso de la Fórmula 3.2. En este punto cabe mencionar que, con

esto quedan inhabilitadas las funcionalidades del componente de la arquitectura propuesta encargado del pre-filtrado de establecimientos basado en datos sociales/temporales; asimismo queda parcialmente inhabilitada la funcionalidad del componente de la arquitectura encargado del pre-filtrado de establecimientos basado en geolocalización. Esto quiere decir que, durante dicha fase de la puesta en marcha del sistema, las recomendaciones se producen al margen del enfoque de recomendación sensible al contexto que se describe en las siguientes subsecciones de este documento.

Regresando al tema del perfilamiento de usuarios, para los propósitos de la usabilidad del sistema, la escala utilizada para la puntuación tanto de establecimientos de alimentos y bebidas como de estilos de cocina y platos, es una escala de cinco puntos basada en la escala psicométrica Likert (Likert, 1932). Dicha escala permite valorar elementos en cualquier tipo de dimensión subjetiva u objetiva en términos de valores cuantitativos, en este caso en la dimensión subjetiva representada por el nivel de agrado/desagrado hacia establecimientos de alimentos y bebidas, estilos de cocina y platos. En detalle, la escala utilizada en esta investigación es la siguiente: *-1=strongly disliked*, *-0.5=disliked*, *0=indiferent*, *0.5=liked* y *1=strongly liked*. Las preferencias por los tópicos deberán ser establecidas por los usuarios directamente en términos de distribuciones de probabilidad (porcentajes, a efectos de usabilidad).

A manera de resumen es posible decir que las técnicas empleadas en esta investigación para la obtención de las preferencias del usuario son una mezcla de técnicas de retroalimentación explícita e implícita, gracias a la cual el sistema es capaz de recomendar establecimientos de alimentos y bebidas aún en el escenario de nuevo usuario.

En dicho escenario, además de solicitar al usuario el establecimiento de sus preferencias a través del cuestionario previamente descrito, también se requiere que establezca otro tipo de información a fin de posibilitar la construcción del correspondiente perfil de usuario. Concretamente, sus relaciones sociales, las cuales se pueden obtener alternativamente de sus redes sociales, específicamente de Facebook, para lo cual es necesario que se proporcionen las credenciales de una cuenta de usuario en dicha red social. En los casos en las que las relaciones sociales no se encuentren explícitamente asociadas a la cuenta de usuario, el usuario deberá establecerlas seleccionando personas de su lista de amigos; invariablemente, este es el caso de las relaciones de trabajo (la propiedad de objeto “hasCoworker” en la ontología del dominio), la cual no se considerada explícitamente en la *API* Graph de Facebook, a diferencia de las relaciones de amistad, románticas y de familia. Evidentemente, durante la fase de entrenamiento del método de recomendación propuesto, el establecimiento de las relaciones sociales por parte de los nuevos usuarios no es necesario, y de hecho no se requiere.

3.8. Pre-filtrado basado en Geolocalización

De acuerdo con Adomavicius & Tuzhilin (2008), el procesamiento de los datos contextuales en un sistema de recomendación sensible al contexto puede ser distribuido a través de las distintas etapas del proceso de recomendación típico de dos dimensiones: usuario y elemento. En esta investigación, una vez que se aprenden las preferencias del usuario, las recomendaciones se calculan de acuerdo al enfoque de recomendación sensible al contexto descrito a continuación, el cual, formalmente, corresponde al paradigma de pre-filtrado. Concretamente, se ha propuesto distribuir el procesamiento de la información contextual, a saber, información de ubicación, información social e información temporal, a través de dos operaciones diferentes durante una etapa previa a la recomendación (operaciones de pre-filtrado): pre-filtrado basado en geolocalización y pre-filtrado basado en información social/temporal.

Durante la operación de pre-filtrado basado en geolocalización, se debe estimar, primeramente, la ubicación geográfica del usuario (el usuario activo) en términos de coordenadas geográficas latitud y longitud. De acuerdo con la arquitectura propuesta, esta actividad corresponde a la funcionalidad de geolocalización en tiempo real

que forma parte de la aplicación cliente, esto es, la aplicación móvil híbrida que materializa la capa de presentación de dicha arquitectura. Por lo tanto, se ha decidido emplear la *API* de geolocalización provista por el *framework* de desarrollo de aplicaciones móviles multiplataforma utilizado para la implementación de dicha aplicación, a saber, el *framework* Apache Cordova.

La *API* de geolocalización de Apache Cordova es una implementación de la especificación de la *API* de geolocalización propuesta por el *W3C* en un esfuerzo por estandarizar una interfaz de alto nivel para acceder mediante lenguajes de *scripts* a la información de ubicación geográfica en dispositivos del lado del cliente. Dicha API puede utilizar distintas fuentes de información de ubicación como el sistema *Global Positioning System* (*GPS*), información inferida a partir de señales de redes de comunicaciones como direcciones IP, direcciones *Media Access Control* (*MAC*) de *Radio Frequency Identification* (*RFID*), Wi-Fi y Bluetooth, así como identificadores de células *Global System for Mobile communications* (*GSM*) y *Code Division Multiple Access* (*CDMA*). De acuerdo con esto, la funcionalidad de geolocalización implementada como parte de la aplicación cliente es capaz de cambiar automáticamente la fuente de la información de ubicación de manera transparente según sea necesario. Este comportamiento representa una característica deseable de las redes sociales y servicios en general basados en localización, y se debe sustentar en la evaluación del estado del dispositivo cliente en todo momento.

Por otro lado, uno de los factores más explotados en los sistemas de recomendación sensible al contexto para *POIs* es, obviamente, la posición geográfica del usuario respecto a la ubicación de los *POIs*, dado que este comúnmente busca aquellos *POIs* más cercanos a su posición en un momento dado. No obstante, asumiendo que las preferencias personales del usuario resultado de las experiencias previas en los *POIs* pueden tener mayor impacto que su posición geográfica por sí sola, en esta investigación se ha propuesto un enfoque de pre-filtrado basado en geolocalización y parcialmente basado en preferencias personales. Específicamente, este enfoque de pre-filtrado basado en geolocalización radica parcialmente en las preferencias del usuario por los tópicos en el modelo *LDA*, las cuales son calculadas por el componente de la arquitectura propuesta encargado del perfilamiento de usuarios (ver sección 3.7 de este capítulo).

En detalle, una vez que se ha aprovechado la funcionalidad de geolocalización antes descrita, a fin de identificar en tiempo real la posición geográfica del usuario o, mejor dicho, la posición geográfica del dispositivo móvil en el que se ejecuta la aplicación cliente de la arquitectura propuesta, es posible identificar todos los establecimientos de alimentos y bebidas dentro del área que rodea a dicha posición geográfica. Para ello es necesario recurrir a la *API* de la red social o servicio basado en localización subyacente, específicamente al servicio o recurso Web para la búsqueda de establecimientos existentes dentro de un área relativa a una posición geográfica especificada. Este tipo de servicios o recursos Web comúnmente permiten especificar el área geográfica de búsqueda en términos de la longitud en metros del radio de la base de un casquete esférico relativo a la posición geográfica especificada. Alternativamente, el área geográfica de búsqueda se puede especificar como un rectángulo delimitado por un par de coordenadas latitud-longitud (búsqueda acotada).

En esta investigación se ha optado por el primer enfoque. De hecho, como parte de las solicitudes de recomendación de establecimientos de alimentos y bebidas, los usuarios (usuarios activos) deben indicar, además de las personas con quienes pretenden visitarlos, la longitud en metros del radio de la base del casquete esférico del área geográfica de búsqueda en la red social o servicio basado en localización subyacente. Como resultado de las solicitudes de recomendación, la ontología del dominio o, mejor dicho, el repositorio de conocimiento se puebla con instancias de las clases de la ontología del dominio "RecommendationRequest" y "GeographicPoint". De hecho, los radios de búsqueda se representan utilizando la propiedad de tipo de dato "searchRadious", cuyo dominio es la clase "RecommendationRequest". Por otro lado, las ubicaciones geográficas de los usuarios, si bien no son indicadas explícitamente por estos en las solicitudes, sino obtenidas automáticamente mediante la funcionalidad de geolocalización, se representan

utilizando la propiedad de objeto “hasGeographicPoint”, cuyo dominio y rango son las clases “RecommendationRequest” y “GeographicPoint”, respectivamente.

Una vez que se han identificado todos los establecimientos de alimentos y bebidas dentro del área que rodea a la posición geográfica del usuario activo al momento de la solicitud de recomendación, es posible consultar, utilizando el lenguaje de consulta *SPARQL*, el repositorio de conocimiento a fin de recuperar las correspondientes instancias de la clase “FoodEstablishment” de la ontología del dominio (la clase que representa los establecimientos de alimentos y bebidas en sí) a partir de los nombres y las coordenadas geográficas de los establecimientos identificados. Por cada resultado provisto en la respuesta de la *API* de la red social o servicio basado en localización subyacente para el que la consulta no produzca un resultado se deberá poblar el repositorio de conocimiento con una instancia de la clase antes mencionada, una instancia de la clase “PostalAddress” (la clase que representa las direcciones postales de los establecimientos) y, eventualmente, una instancia de la clase “Cuisine” (la clase que representa los estilos de cocina de los establecimientos) y una instancia de la clase “Dish” (la clase que representa los platos servidos por los establecimientos) (ver la subsección 3.4.2 de este capítulo).

Con el objetivo de considerar las preferencias de los usuarios por los tópicos en la operación de pre-filtrado basado en geolocalización, se ha propuesto un algoritmo para el ajuste iterativo de la longitud del radio de búsqueda establecida por el usuario activo en la solicitud, de tal forma que se asegure que los resultados de dicha operación de pre-filtrado cubren suficientemente la mayor cantidad de tópicos de interés para él/ella, a cambio del incremento mínimo posible de la longitud del radio de búsqueda. Dicho algoritmo requiere la identificación del tópico con mayor probabilidad de ser asignado a cada documento que representa a un establecimiento en el área geográfica de búsqueda (una instancia de la clase “FoodEstablishment” de la ontología del dominio). Para ello, se deberá recurrir al conjunto de resultados producido previamente por la consulta *SPARQL* antes mencionada. Nótese que este algoritmo corresponde a la parte de la funcionalidad del componente de la arquitectura propuesta encargado del pre-filtrado de establecimientos basado en geolocalización que queda inhabilitada durante la fase de “entrenamiento” de la puesta en marcha del método de recomendación propuesto.

Evidentemente, los establecimientos de alimentos y bebidas para los que no exista una instancia de la clase “FoodEstablishment” en dicho conjunto de resultados, esto es, los establecimientos nuevos en el sistema de recomendación deberán considerarse con la misma probabilidad de ser asignados a cada uno de los tópicos del modelo *LDA* construido a partir de los *check-ins* hechos por los usuarios en los establecimientos. El mismo criterio se deberá aplicar a las instancias de la clase “FoodEstablishment” en dicho conjunto de resultados para las que aún no exista un tópico asociado a través de la propiedad de tipo de dato “hasBestRankedTopic”, cuyo dominio es la clase “FoodEstablishment”. El objetivo de estas medidas es afrontar el reto de la escasez de datos. En otras palabras, estas medidas permiten promover la diversidad en las recomendaciones al evitar que los establecimientos para los que no existan datos históricos (es decir, *check-ins*) en el sistema sean filtrados en este punto del proceso de recomendación.

La Figura 3.8 muestra un diagrama de secuencia (notación *UML*) que modela las interacciones entre los objetos involucrados en la implementación del algoritmo para el ajuste iterativo de la longitud del radio de búsqueda.

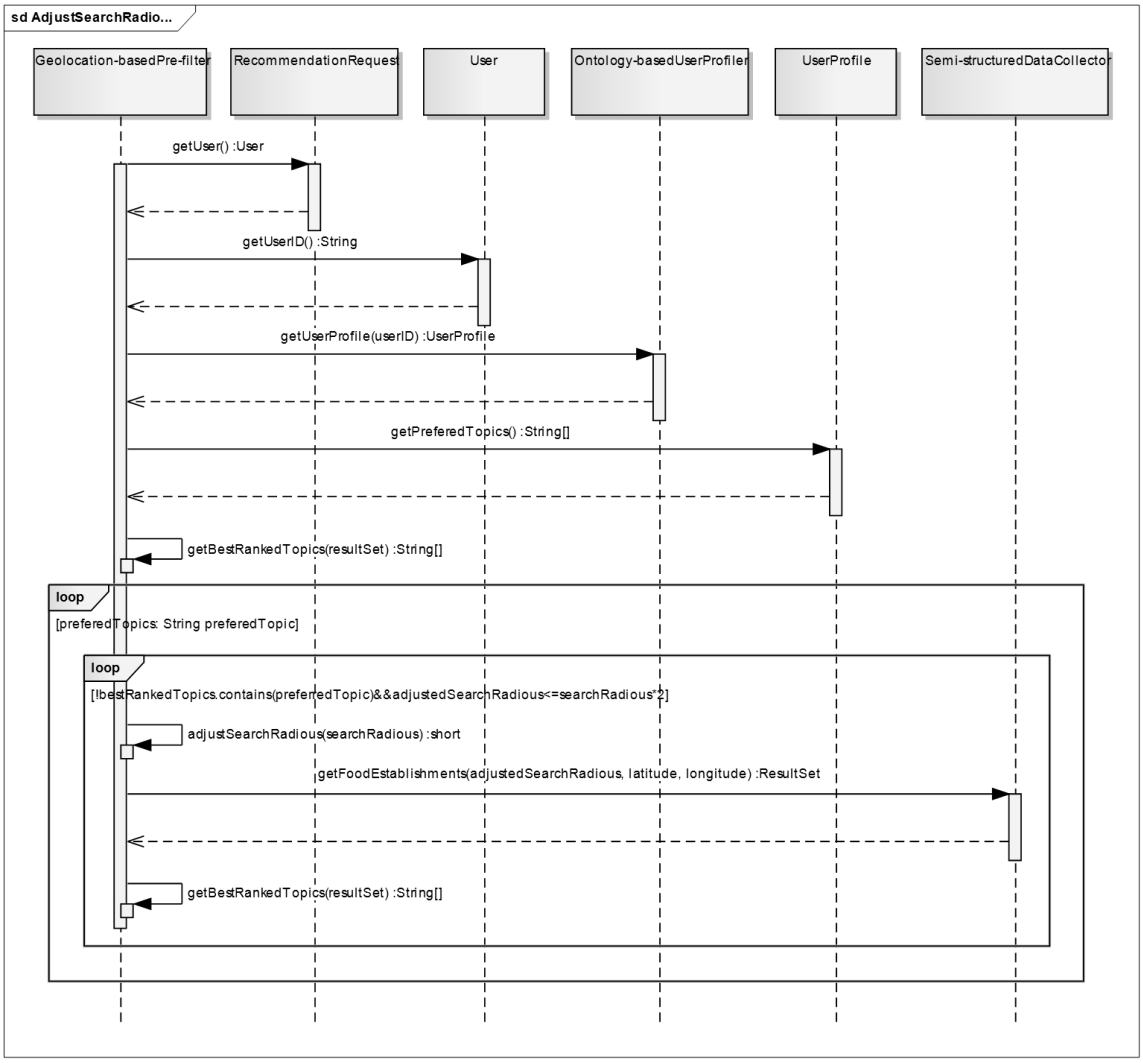


Figura 3.8. Diagrama de secuencia de algoritmo para el ajuste iterativo del radio de búsqueda de establecimientos de alimentos y bebidas.

Como se puede observar en el diagrama mostrado en la Figura 3.8, si por cada tópico de interés para el usuario activo (tópicos para los cuales el componente de la arquitectura propuesta encargado del perfilamiento de usuarios ha resuelto la preferencia, respecto al usuario activo, a un valor decimal mayor a cero) no existe al menos una instancia de la clase "FoodEstablishment" en el conjunto de resultados de la consulta *SPARQL* de recuperación de establecimientos asociada a dicho tópico (a un valor de un literal del tipo de dato "string" de *XML Schema* que representa al identificador del tópico) a través de la propiedad de tipo de dato "hasBestRankedTopic", entonces el valor de la longitud del radio de búsqueda se incrementa en un 10% respecto a su valor anterior (el método *adjustSearchRadious*). Este proceso se repite mientras el valor acumulado sea menor al doble del valor de la longitud del radio de búsqueda establecida por el usuario activo en la solicitud de recomendación.

Por cada incremento en la longitud del radio de búsqueda se ejecuta una nueva llamada al servicio o recurso Web para la búsqueda de los establecimientos existentes dentro de un área relativa a una posición geográfica dada de la *API* de la red social o servicio basado en localización subyacente (el método *getFoodEstablishments*), a fin de identificar todos los establecimientos de alimentos y bebidas dentro del área geográfica extendida relativa a la posición geográfica del usuario activo, los cuales se consideran la salida

tentativa de la operación de pre-filtrado basado en geolocalización. Sucesivamente, se ejecuta la consulta *SPARQL* antes mencionada. Cabe mencionar que los detalles de la llamada a la *API* de la red social o servicio basado en localización subyacente y la sucesiva ejecución de la consulta *SPARQL* de recuperación de establecimientos (instancias de la clase "FoodEstablishment" de la ontología del dominio) no se incluyen en la Figura 4 ya que corresponden conceptualmente a funcionalidades de los componentes de la arquitectura propuesta encargados de la recolección de datos semi-estructurados y la instanciación automática de la ontología del dominio.

3.9. Pre-filtrado basado en Datos Sociales/Temporales

En lo que respecta al procesamiento de los datos sociales/temporales, es necesario, primeramente, inferir la situación social del usuario activo, a partir de las relaciones sociales que tiene con las personas indicadas en la solicitud de recomendación (es decir, sus acompañantes). Asimismo, es necesario inferir el día de la semana y el periodo del día durante los cuales el usuario y sus acompañantes pretenden visitar los establecimientos de alimentos y bebidas, a partir del *timestamp* de la solicitud.

Este escenario de la funcionalidad del componente de la arquitectura propuesta encargado del pre-filtrado basado en datos sociales/temporales se basa en la suposición de que el usuario está buscando establecimientos para visitar en el momento de la solicitud de recomendación. En este contexto es importante mencionar que, como se verá más adelante en este capítulo (sección titulada "Interfaz de Usuario"), opcionalmente, el usuario puede indicar el día de la semana y el periodo del día en el que pretende visitar los establecimientos explícitamente en la solicitud de recomendación. En ese caso, evidentemente, el proceso de inferencia de dichos parámetros de la solicitud no es necesario.

En detalle, en las instancias de la clase "RecommendationRequest" de la ontología del dominio, el valor de la propiedad de tipo de dato "hasSocialSituation" se infiere utilizando una regla SPIN similar a la regla empleada para inferir el valor de la propiedad de tipo de dato "visitedInSocialSituation", cuyo dominio es la clase "CheckIn". Asimismo, los valores de las propiedades de tipo de dato "hasIntendedDayOfWeek" y "hasIntendedPeriodOfDay", cuyo dominio es la clase "RecommendationRequest", se infieren utilizando un enfoque similar al empleado para la inferencia de las propiedades de tipo de dato "visitedAtPeriodOfDay" y "visitedAtDayOfWeek", cuyo dominio es la clase "CheckIn" (ver sección 3.6 de este capítulo).

Una vez inferida la información contextual de alto nivel a partir de la información contextual de bajo nivel asociada a las instancias de la clase "RecommendationRequest", es posible llevar a cabo el pre-filtrado basado en datos sociales/temporales propiamente dicho. En detalle, esto corresponde, por un lado, a la estimación de la probabilidad de cada establecimiento de alimentos y bebidas en la salida de la operación de pre-filtrado basado en geolocalización, dada la distribución de probabilidad del documento que lo representa en el modelo *LDA*, respecto a la situación social inferida y el día de la semana y el periodo del día pretendidos (palabras que componen los documentos en el modelo *LDA*).

Para ello, es necesario: (1) seleccionar previamente el tópico con la mayor probabilidad de ser asignado al documento que representa al establecimiento (una instancia de la clase "FoodEstablishment" de la ontología del dominio), (2) seleccionar tanto el valor del atributo contextual "SocialSituation" obtenido a partir de la situación social inferida, como los valores de los atributos contextuales "DayOfWeek" y "PeriodOfDay" obtenidos respectivamente a partir del día de la semana y el periodo del día pretendidos, y calcular la probabilidad de generar, en el tópico seleccionado, cada uno de esos valores interpretados como palabras y (3) calcular el producto de la probabilidad de asignar el tópico seleccionado al documento que representa al establecimiento y la suma de las probabilidades antes calculadas, tal como se muestra en la Fórmula 3.7.

$$likelihood_{p,ss,pd,dw,t} = topicProbability(p,t) * \\ \left(topicWordProb(t,ss) + topicWordProb(t,pd) + topicWordProb(t,dw)\right) \quad (3.7)$$

Donde:

- *p* es el documento que representa al establecimiento de alimentos y bebidas para el que se calcula la probabilidad (una instancia de la clase "FoodEstablishment" de la ontología del dominio).
- *ss* es el valor del atributo contextual "SocialSituation" obtenido a partir de la situación social inferida.
- *pd* es el valor del tributo contextual "DayOfWeek" obtenido a partir del día de la semana pretendido.
- *dw* es el valor del atributo contextual "PeriodOfDay" obtenido a partir del periodo del día pretendido.
- *t* es el identificador del tópico con mayor probabilidad de ser asignado al documento *p*, esto es, el valor asociado a la instancia de la clase "FoodEstablishment" representada por el documento *p* a través de la propiedad de tipo de dato "hasBestRankedTopic".
- *topicProbability(p,t)* es la probabilidad de asignar el tópico *t* al documento *p*, la cual es calculada por el componente de la arquitectura propuesta encargado del descubrimiento de los tópicos (ver sección 3.6 de este capítulo).
- *topicWordProb(x,y)* es la probabilidad de generar la palabra *y* en el tópico *x*.

De acuerdo con las medidas destinadas a promover la diversidad en las recomendaciones, las cuales fueron descritas en la sección anterior, la probabilidad de cualquier establecimiento nuevo en la salida de la operación de pre-filtrado basado en geolocalización, así como la probabilidad de las instancias de la clase "FoodEstablishment" para las cuales aún no exista un tópico asociado a través de la propiedad de tipo de dato "hasBestRankedTopic", deberá calcularse como la media aritmética de la probabilidad calculada respecto a cada tópico en el modelo *LDA*.

Como en el caso del número de veces que un tópico se asignó a un documento de un modelo *LDA*, la probabilidad de generar una palabra en un tópico se puede obtener fácilmente a partir de un objeto de la clase estática "LatentDirichletAllocation.GibbsSample" (tipo de retorno del método "gibbsSampler" de la clase "LatentDirichletAllocation") de la biblioteca *LingPipe*, una vez que se ha construido un modelo *LDA*. Específicamente, utilizando el método "topicWordProb", el cual debe recibir como parámetros los identificadores del tópico y la palabra de interés, y devuelve como resultado un valor del tipo de dato *double* de Java.

Una vez que se ha estimado la probabilidad de cada establecimiento de alimentos y bebidas en la salida de la operación de pre-filtrado basado en geolocalización, se deben filtrar todos aquellos establecimientos en dicha salida para los cuales el valor de la probabilidad estimada es menor que el valor de un umbral calculado mediante la medida estadística de percentiles, de modo que el conjunto de establecimientos resultante se considera la salida de la operación de pre-filtrado basado en datos sociales/temporales. En esta investigación, se ha propuesto establecer el valor del umbral en el valor que separa al 50% de las probabilidades estimadas (ordenadas de menor a mayor), esto es, en el valor del percentil 50 (P_{50}). Para ello, se ha propuesto emplear el método conocido como *nearest rank*, uno de los métodos más populares para el cálculo de percentiles en una lista de valores ordenados, el cual consiste en calcular la posición correspondiente al valor que separa la lista de valores en el percentil buscado utilizando la Fórmula 3.8 y tomar el valor que ocupa dicha posición en la lista de valores (Ferguson, 1959).

$$n = \frac{P}{100} xN \quad (3.8)$$

Donde:

- *p* representa el percentil buscado.
- *N* es el número de valores en la lista.

3.10. Calculo de Similitudes Semánticas basado en Ontologías

La recomendación propiamente dicha de establecimientos de alimentos y bebidas de posible interés para el usuario activo requiere una fase previa de cálculo de similitudes semánticas entre todos los establecimientos en el espacio de recomendación (instancias de la clase de referencia de la ontología del dominio en el repositorio de conocimiento), a fin de identificar aquellos establecimientos similares semánticamente a los establecimientos del gusto del usuario, esto es, los establecimientos frecuentados por él/ella, así como los establecimientos para los cuales él/ella ha indicado su preferencia en el pasado.

Para el cálculo de las similitudes semánticas se ha propuesto una nueva métrica de similitud basada en ontologías que utiliza un enfoque híbrido taxonómico-no taxonómico basado en características. Esta métrica pretende evitar el problema del "arranque en frío", específicamente, la modalidad del "elemento nuevo" al depender de conocimiento semántico acerca de los elementos a recomendar y no de los *ratings* dados por los usuarios a los elementos (Su & Khoshgoftaar, 2009). Los valores resultantes de dicho cálculo se representan en memoria en una matriz cuadrada de orden *n* -la matriz de similitudes generales, donde *n* representa el número de instancias de la clase de referencia en el repositorio de conocimiento, y cada elemento x_{ij} en dicha matriz es un valor decimal en el rango [0,1] que mide la similitud entre las instancias *i* y *j*.

En términos generales, la métrica propuesta toma en cuenta no solo el número de terceras instancias/valores asociadas al mismo tiempo a los pares de instancias de la clase de referencia a comparar sino también el número de pares de terceras instancias que no están asociadas simultáneamente al par de instancias a comparar, pero que son similares en un sentido taxonómico. Esto se ha logrado integrando el concepto de similitud semántica taxonómica propuesto en (Sánchez et al., 2012) al concepto de similitud semántica propuesto en (Carrer-Neto et al., 2012), el cual a su vez se basa en el concepto de similitud semántica inferencial propuesto en (Blanco-Fernández et al., 2008b), y permite capturar conocimiento no taxonómico explícito. Por lo tanto, el cálculo de las similitudes generales requiere el cálculo previo de las similitudes taxonómicas, a fin de determinar el número de pares de terceras instancias que son similares en un sentido taxonómico; cabe mencionar que esto es aprovechado para capturar conocimiento no taxonómico implícito de manera similar a como lo hace la métrica propuesta por Blanco-Fernández y colaboradores (Blanco-Fernández et al., 2008b).

Los valores resultantes de dicho cálculo se representan en memoria en una serie de matrices de orden *n* (una por cada propiedad de objeto de referencia), donde *n* es el número de instancias de la clase de la ontología del dominio en el rango de la propiedad de objeto correspondiente -matrices de similitudes taxonómicas.

Se ha propuesto almacenar en una base de datos relacional tanto la matriz de similitudes generales como las matrices de similitudes taxonómicas, a fin de evitar el recalculo de los valores de similitud con cada solicitud de recomendación. No obstante, dado que los establecimientos de alimentos y bebidas se recuperan bajo demanda desde la *API* de red social o servicio basado en localización subyacente, dichas matrices se deben actualizar con cada solicitud de recomendación que resulte en al menos una nueva instancia de la clase "FoodEstablishment" de la ontología del dominio o de las clases de la ontología en el rango de las propiedades de objeto de referencia en el cálculo de las similitudes. En aras de la brevedad, en este documento no se abordan los detalles del procedimiento de actualización de las matrices de similitudes.

En esta fase del método de recomendación propuesta, la interpretación dada a la métrica de similitud semántica taxonómica propuesta por Blanco-Fernández et al. (2008) es distinta a la interpretación dada en la fase de perfilamiento de usuarios (ver sección 3.7 de este capítulo). En este caso, la interpretación es la siguiente (ver Fórmula 3.9). Cabe resaltar que, como se mencionó en la sección citada de este capítulo, en esta investigación, la similitud taxonómica se calcula como el complemento del valor racional (en el rango [0, 1]) resultante de la métrica propuesta por Blanco-Fernández et al. (2008), ya que, formalmente, esta permite medir la distancia semántica, no la similitud. Por último, es importante tener en cuenta que, a diferencia del cálculo de las similitudes generales, el cálculo de las similitudes taxonómicas aplica únicamente a propiedades de objeto.

Asimismo, dado el enfoque utilizado en esta investigación para el modelado del dominio, en el cual los establecimientos de alimentos y bebidas se pueden caracterizar, o bien por sus estilos de cocina, o bien por los platillos que sirven, puede darse el caso de que dos instancias de dicha clase a comparar no se encuentren involucradas, tanto en afirmaciones de propiedad de objeto en las que el predicado (tripletas *RDF*) sea la propiedad de objeto "servesCuisine", como en afirmaciones en las que el predicado sea la propiedad "servesDish", y que, sin embargo, ambas sean propiedades de referencia para el cálculo de las similitudes. En dicho caso, se deberá recurrir a las propiedades de objeto "isDishOf" y "hasDish" (propiedades inversas una de la otra), las cuales permiten asociar los platillos a las cocinas de las que forman parte, y viceversa. De hecho, la propiedad "servesCuisine" se ha definido en la ontología del dominio como super-propiedad de la secuencia formada por las expresiones de propiedad de objeto "servesDish" y "isDishOf" (servesDish o isDishOf -> servesCuisine); de igual manera, se ha añadido una propiedad de objeto extra denominada "servesCuisineDish", y se ha añadido un axioma de sub-propiedad de objeto para definirla como super-propiedad de la secuencia formada por las propiedades de objeto "servesCuisine" y "hasDish" (servesCuisine o hasDish -> servesCuisineDish).

La adición de esta propiedad de objeto se debe a que, definir directamente la propiedad "servesDish" como super-propiedad de la cadena antes mencionada crearía un ciclo que causaría la desestimación del axioma de sub-propiedad, generando un error en el razonador empleado para inferir las afirmaciones de propiedad correspondientes. De este modo, en los casos en los que no existan afirmaciones de propiedad de objeto declaradas en las que el predicado (tripletas *RDF*) sea la super-propiedad "servesCuisine", se deberán considerar las afirmaciones inferidas en las que el predicado sea dicha super-propiedad; mientras que en los casos en los que no existan afirmaciones de propiedad declaradas en las que el predicado sea la propiedad "servesDish", se deberán considerar las afirmaciones inferidas en las que el predicado sea la super-propiedad "servesCuisineDish".

$$tSimilarity(a,b,p) = 1 - \left(\log_2\left(1 + \frac{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)|}{|\emptyset(A)|\backslash|\emptyset(B)| + |\emptyset(B)|\backslash|\emptyset(A)| + |\emptyset(A) \cap \emptyset(B)|}\right)\right) \quad (3.9)$$

Donde:

- *p* es una de las propiedades de objeto de referencia.
- *a*, *b* es un par de instancias de clases (denominadas *A* y *B*) en la jerarquía de clases de la ontología del dominio cuya clase raíz es la clase en el rango de la propiedad de objeto representada por *p*.
- *Ø(A)* representa al conjunto de características taxonómicas que describen a la clase *A* en términos de subsunción de conceptos.
- *Ø(B)* representa al conjunto de características taxonómicas que describen a la clase *B* en términos de subsunción de conceptos.
- *Ø(X)\ Ø(Y)* es la cardinalidad del conjunto de características taxonómicas diferenciales de la clase *X* respecto a la clase *Y*.

- *Ø(X) ∩ Ø(Y)* es la cardinalidad del conjunto de características taxonómicas comunes a las clases *X* y *Y*.

Una vez que se han construido las matrices de similitudes taxonómicas necesarias, es posible establecer el valor de un umbral que permita considerar solo lo pares de terceras instancias cuyos valores de similitud taxonómica sean significativos, esto es, solo los pares de terceras instancias cuyos valores de similitud taxonómica sean mayores a un valor predeterminado. En esta investigación, se ha propuesto establecer dicho valor en el valor que separa al 50% de los valores de similitud calculados (ordenados de menor a mayor), esto es, en el valor del percentil 50 (P_{50}). Como en el caso del umbral para las probabilidades de los establecimientos de alimentos y bebidas en la operación de pre-filtrado basado en datos sociales/temporales, el método propuesto para el cálculo del percentil antes mencionado es el método *nearest rank*, cuya fórmula fue presentada en la sección anterior (Fórmula 3.8).

Una vez que se ha establecido el valor del umbral para las similitudes taxonómicas, es posible calcular las similitudes generales. Para ello se ha propuesto un algoritmo que engloba el cálculo de las similitudes taxonómicas (y, evidentemente, la construcción de las matrices de similitudes taxonómicas), así como el cálculo del umbral antes mencionado (ver Figura 3.9). El núcleo de este algoritmo es representado por la Fórmula 3.10. Es importante tener en cuenta que, una vez que se han calculado las similitudes generales, es necesario aplicar la función de normalización "L2-norm" (normalización Euclidiana) para normalizar cada vector en la matriz de similitudes generales y obtener valores racionales en el rango [0,1].

$$oSimilarity(a, b, P) = \sum_{i=1}^{\#P} \left(\frac{count(a\ P[i]\ ?\ p\ \&\&\ b\ P[i]\ ?\ p) + notCommonTS(a, b, P[i])}{max\big(count(a\ P[i]\ ?\ p), count(b\ P[i]\ ?\ p)\big)} \right) * weight(P[i]) \quad (3.10)$$

Donde:

- *a*, *b* son las instancias de la clase de referencia de la ontología del dominio a comparar.
- *P* es un *array* de las propiedades de tipo de dato/objeto respecto a las cuales se debe calcular la similitud entre las instancias de la clase de referencia, es decir, las propiedades de referencia para el cálculo de las similitudes, y *P[i]* representa a cada una de esas propiedades.
- *#P* representa el tamaño del array *P*, es decir, el número de propiedades de referencia a considerar en el cálculo de las similitudes.
- *weight(P[i])* representa la relevancia o peso de cada una de las propiedades de referencia respecto a las demás.
- *count(a P[i] ?p && b P[i] ?p)* es el número de terceras instancias o valores asociadas simultáneamente a las instancias *a* y *b* a través de la propiedad de objeto o tipo de dato *P[i]*, esto es, el número de terceras instancias o valores en común.
- *max(count(a P[i] ?p), count(b P[i] ?p))* es el número máximo entre el número de instancias/valores asociadas a la instancia *a* a través de la propiedad de objeto/tipo de dato *P[i]* y el número de instancias/valores asociadas a la instancia *b* a través de la misma propiedad de objeto/tipo de dato.
- *notCommonTS(a,b,P[i])* es el número de pares de terceras instancias que son similares taxonómicamente (instancias únicas), en donde una de las instancias está asociada a la instancia *a* a través de la propiedad de objeto *P[i]*, mientras que la otra está asociada a la instancia *b* a través de la misma propiedad de objeto. Este valor se debe obtener a su vez mediante el uso de la Fórmula 3.11, la cual se presenta a continuación.

$$notCommonTS(a, P[i], b) = \sum_{j=0}^{\#APB} \sum_{k=0}^{APB[j]} [APB(a, P[i], b)[j][k] > hSthreshold] \quad (3.11)$$

Donde:

- *a*, *b* y *P[i]* se interpretan como en el caso de la Fórmula 3.10.
- *APB(a,P[i],b)* es una matriz de las similitudes taxonómicas entre cada tercera instancia asociada a la instancia *a* a través de la propiedad *P[i]* y cada instancia única asociada a la instancia *b* a través de la misma propiedad, esto es, una sub-matriz de la matriz de similitudes taxonómicas calculadas respecto a la propiedad de objeto *P[i]*.
- *#APB* es el tamaño de la primera dimensión de la matriz *APB*, esto es, el número de terceras instancias asociadas a la instancia *a* pero no a la instancia *b* a través de la propiedad de objeto *P[i]*.
- *#APB[i][j]* es el tamaño de la segunda dimensión de la matriz *APB*, esto es, el número de terceras instancias asociadas a la instancia *b* pero no a la instancia *a* a través de la propiedad de objeto *P[i]*.
- *hSthreshold* es el valor establecido para el umbral de los valores de similitud taxonómica.

Como se puede inferir de la Fórmula 3.10, la notación utilizada para representar los términos *count(a P[i] ?p && b P[i] ?p)* y *max(count(a P[i] ?p), count(b P[i] ?p))* hace referencia a la sintaxis y semántica de las consultas *SPARQL* que, evidentemente subyacen al cálculo de las similitudes, específicamente a las funciones de agregación COUNT y MAX. En lo que respecta al segundo término, se ha diseñado una consulta *SPARQL* que permite aplicar la función MAX al conjunto de la solución de en un patrón de unión (el operador UNION) que combina los resultados de un par de sub-consultas, cada una de las cuales permite aplicar la función COUNT al conjunto de la solución de uno de los patrones indicados, a saber, *my:a my:P[i] ?p* y *my:b my:P[i] ?p*. En dichos patrones, "*a*" y "*b*" se remplazan con las partes locales de las *IRIs* de las instancias de la clase "FoodEstablishment" para las cuales se va a calcular la similitud, "*P[i]"* se remplaza con la parte local del *IRI* de la propiedad de tipo de dato/objeto de referencia correspondiente a la iteración en el cálculo de la similitud entre "*a"* y "*b"* y "*my*" es el prefijo definido para el espacio de nombres de la ontología del dominio.

```
public double[][] fillTSMatrix(ObjectProperty p){

    OntModel model=OntologyModel.getOntModel();
    Query query = new Query();
    ResultSet results = query.queryAllInstancesOfSubclassesOfRangeClass(model, p);
    List<QuerySolution> list=new ArrayList<>();

    while(results.hasNext()){
       list.add(results.next());
    }

    double[][] tSMatrix = new double[list.size()][list.size()];

    int i, j;
    Resource resource;
    Individual a, b;

    for(i=0; i<tSMatrix.length; i++){

       resource = list.get(i).getResource("subject");
```

```
        a = model.getResource(resource.getURI()).as(Individual.class);

        for(j=i+1; j<tSMatrix[i].length; j++){

            resource = list.get(j).getResource("subject");
            b = model.getResource(resource.getURI()).as(Individual.class);

            tSMatrix[i][j]=tSimilarity(a, b, p);
        }
    }

    return tSMatrix;
}

public double[][] fillOSMatrix(double tST, OntProperty[] ps, double[] ws){

    OntModel model=OntologyModel.getOntModel();
    Query query = new Query();
    ResultSet results = query.queryAllFoodEstablishmennts(model);
    List<QuerySolution> list=new ArrayList<>();

    while(results.hasNext()){
        list.add(results.next());
    }

    double[][] oSMatrix = new double[list.size()][list.size()];

    int i, j;
    Resource resource;
    Individual a, b;

    for(i=0; i<oSMatrix.length; i++){

        resource = list.get(i).getResource("subject");
        a = model.getResource(resource.getURI()).as(Individual.class);

        for(j=i+1; j<oSMatrix[i].length; j++){

            resource = list.get(j).getResource("subject");
            b = model.getResource(resource.getURI()).as(Individual.class);

            oSMatrix[i][j]=oSimilarity(a, b, ps, ws, tST);
        }
    }
    return oSMatrix;
}

public double[][] calculateSimilarities(OntProperty[] ps, double[] ws){

    Object[] tSMatrixes =new Object[ps.length];
```

```
    for(int i=0; i<ps.length; i++)
       if(ps[i].isObjectProperty())
          tSMatrixes[i] = fillTSMatrix(ps[i].asObjectProperty());

    double tSThreshold = calculateTSThreshold(tSMatrixes);
    double[][] oSMatrix = fillOSMatrix(tSThreshold, ps, ws);
    double[][] nOSMatrix = normalizeOSMatrix(oSMatrix);
    return nOSMatrix;
 }
```

Figura 3.9. Algoritmo de cálculo de similitudes semánticas.

Es importante tener en cuenta que, el extracto del algoritmo de cálculo de similitudes semánticas mostrado en la Figura 5 no incluye los detalles de los métodos que implementan las métricas representadas por las Fórmulas 3.9 y 3.10, esto es, la métrica de similitud semántica taxonómica y la métrica de similitud semántica general (taxonómica e inferencial), ni tampoco los detalles de la implementación de la Fórmula 3.11. De hecho, este extracto tiene como objetivo mostrar más bien el procedimiento de construcción inicial de las matrices de similitudes (ver los métodos "fillTSMatrix" y "fillOSMatrix"). Tampoco se presentan los detalles de la clase estática "OntologyModel" y la clase "Query"; no obstante, es importante saber que estas forman parte de la implementación del componente de la arquitectura propuesta encargado de la instanciación automática de la ontología del dominio; mientras que la primera permite gestionar los distintos modelos de ontología Jena empleados por el sistema de recomendación sensible al contexto de alimentos y bebidas como un todo, la segunda permite recuperar datos del repositorio de conocimiento, a saber, las instancias de la clase "FoodEstablishment", en el caso del cálculo de las similitudes generales, y las instancias de las subclases de las clases en el rango de las propiedades de referencia, en el caso del cálculo de las similitudes taxonómicas.

3.11. Filtrado Colaborativo basado en Modelos de Tópicos

Una vez que se ha completado la matriz de similitudes generales es posible calcular las recomendaciones propiamente dichas. Para ello es necesario, primero, buscar en dicha matriz los elementos que representan valores de similitud entre instancias de la clase de referencia que representan las preferencias del usuario activo e instancias de la clase de referencia en la salida de la operación de pre-filtrado basado en datos sociales/temporales (de aquí en adelante llamado conjunto candidato de establecimientos), y construir una nueva matriz para representar en memoria dichos elementos -la matriz reducida de similitudes generales.

La matriz reducida de similitudes generales se debe reducir sucesivamente a aquellos elementos que representan valores de similitud significativos para el cálculo de las recomendaciones, esto es, a aquellos elementos que representan valores de similitud mayores a un umbral prestablecido o, en otras palabras, a los elementos que corresponden a las *k* instancias que representan las preferencias del usuario activo más similares a las instancias que representan al conjunto candidato de establecimientos.

Como en el caso del umbral para las probabilidades de los establecimientos de alimentos y bebidas en la operación de pre-filtrado basado en datos sociales/temporales y el umbral para las similitudes taxonómicas en el cálculo de las similitudes generales, se ha propuesto establecer el valor del umbral para las similitudes generales en el cálculo de las recomendaciones en el valor que separa al 50% de los valores de similitud general calculados (ordenados de menor a mayor), esto es, en el valor del percentil 50 (P_{50}); para ello se ha utilizado, de igual manera, el método *nearest rank*, cuya fórmula fue presentada en la sección 3.9 de este capítulo (Fórmula 3.8).

Finalmente, se deben inferir los valores que representan los *ratings* que el usuario activo daría a los establecimientos en el conjunto candidato de establecimientos en la matriz reducida de similitudes. Para ello se

ha propuesto el uso del popular método basado en la media aritmética ponderada (ver Fórmula 3.12). Los valores resultantes de dicho cálculo, se representan en memoria en un *array* de orden *n* -el *array* de recomendaciones, en donde *n* es el número de establecimientos en el conjunto candidato de establecimientos. Cabe mencionar que los *ratings* estimados no se deben confundir con los *ratings* dados explícita o implícitamente por los usuarios a los establecimientos, es decir, a las instancias de la clase de referencia de la ontología del dominio.

$$recommendation(a, B, u) = \frac{1}{\#B}\sum_{j=1}^{\#B}\left(cRating(u, B[j]) * oSimilarity(a, B[j])\right) \quad (3.12)$$

Donde:

- *u* es el usuario activo para quien se van a calcular las recomendaciones.
- *a* es una instancia de la clase "FoodEstablishment" de la ontología del dominio en el conjunto candidato de establecimientos.
- *B* es un *array* de las instancias de la clase "FoodEstablishment" que representan las preferencias del usuario *u*, esto es, las instancias de dicha clase valoradas implícita o explícitamente por el usuario *u* en el pasado, que están emparejadas con la instancia *a* en la matriz reducida de similitudes generales.
- *oSimilarity(a,B[j])* es el valor de similitud obtenido para las instancias *a* y *B[j]* mediante el uso de la métrica de similitud representada por la Fórmula 3.10.
- *cRating(u,B[j])* es el *rating* dado en el pasado por el usuario *u* a la instancia *B[j]*, el cual es calculado por el componente de la arquitectura propuesta encargado del perfilamiento de usuarios (ver sección de 3.7 este capítulo).

De acuerdo con la Fórmula 3.12), el *rating* que el usuario *u* daría a una instancia *a* de la clase de referencia de la ontología del dominio se estima a partir de la media aritmética ponderada de los *ratings* dados por él a todas las instancias de la misma clase emparejadas a la instancia *a* en la matriz reducida de similitudes generales, es decir, a las instancias en el *array B*. Cada *rating* en dicha media se pondera con el valor de la similitud calculada para la instancia *a* y la correspondiente instancia en *B*.

Finalmente, se ha propuesto ajustar cada valor en el *array* de recomendaciones a partir de las preferencias del usuario activo por los tópicos en el modelo *LDA* (ver Fórmula 3.13). De acuerdo con esto, es necesario ajustar cada *rating* en dicho *array* añadiendo el valor de la preferencia del usuario por el tópico con mayor probabilidad de ser asignado al documento que representa al establecimiento para el cual se estimó el *rating* (la instancia *a*); de modo que resulta necesario seleccionar previamente dicho tópico. Como se puede deducir, se ha empleado un método de hibridación por ponderación a fin de combinar una técnica de recomendación basada en conocimiento, esto es, la métrica de similitud semántica basada en ontologías, y una técnica de filtrado colaborativo basado en modelos, esto es, la técnica de recomendación de filtrado colaborativo basado en modelos probabilísticos generativos de tópicos que se describe en esta subsección.

$$fRecommendation(a, u, r, t) = r + topicProbability(a, t) * preference(u, t) \quad (3.13)$$

Donde:

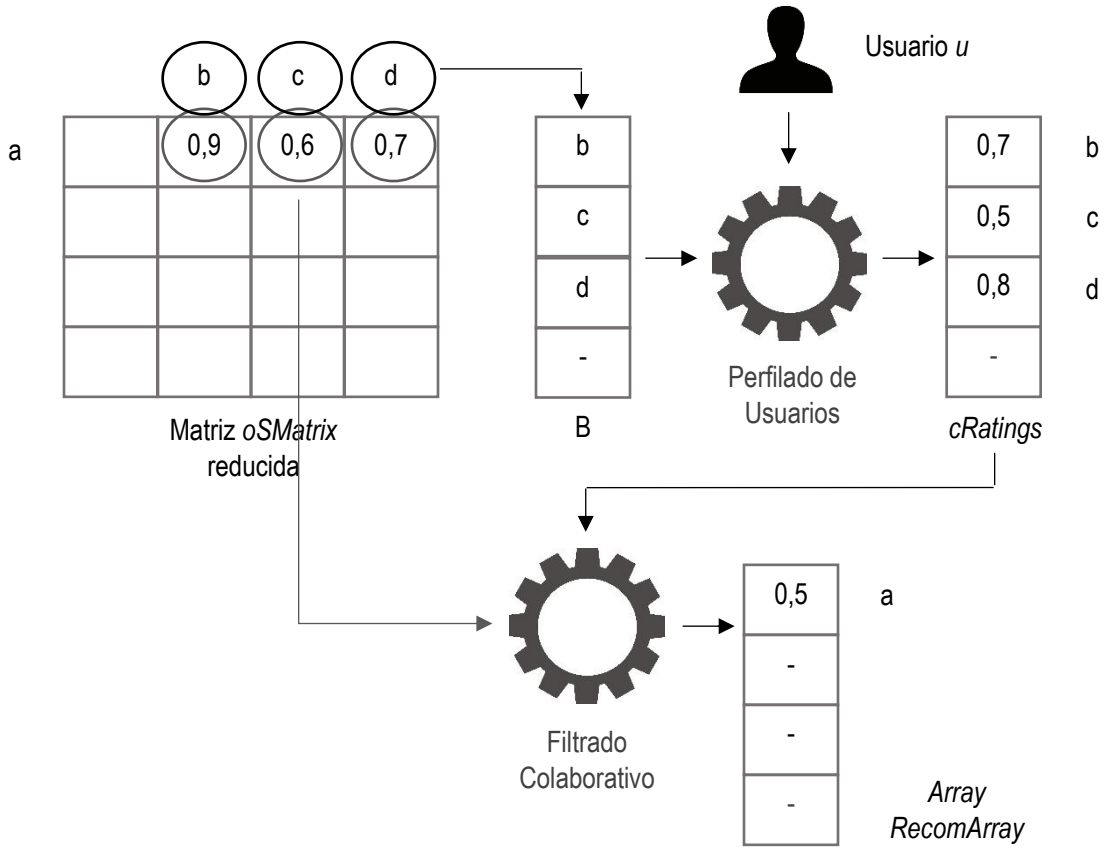
- *a* es el documento en el modelo *LDA* que representa a la instancia de la clase de referencia de la ontología del dominio para la cual se va a ajustar el *rating* previamente estimado (es decir, *r*).
- *u* es el usuario activo para quien se calcularon las recomendaciones.

- *t* es el identificador del tópico en el modelo *LDA* con mayor probabilidad de ser asignado al documento *a*, esto es, el valor asociado a la instancia de la clase "FoodEstablishment" representada por el documento *a* a través de la propiedad de tipo de dato "hasBestRankedTopic".
- *topicProbability(a,t)* es la probabilidad de asignar el tópico *t* al documento *a*, la cual es calculada por el componente de la arquitectura propuesta encargado del descubrimiento de los tópicos (ver sección 3.6 de este capítulo).
- *preference(u,t)* es la preferencia del usuario activo *u* por el tópico *t*, la cual es calculada por el componente de la arquitectura encargado del perfilamiento de usuarios (ver sección 3.7 de este capítulo).

Como en el caso de los vectores en la matriz de similitudes generales, es necesario aplicar la función de normalización "L2-norm" (normalización Euclidiana) para normalizar el *array* de recomendaciones una vez que sus valores (*ratings* estimados) han sido ajustados y obtener así valores racionales en el rango [0,1]. Una vez llevado a cabo el proceso de normalización antes descrito, es posible ordenar los *ratings* normalizados de mayor a menor para, finalmente, presentar al usuario activo, como resultado de la solicitud de recomendación, los establecimientos para los cuales dichos *ratings* fueron estimados y ajustados.

Como se verá en el siguiente capítulo de este documento, el número de establecimientos a presentar al usuario activo como resultado de la solicitud de recomendación se puede configurar, al igual que las propiedades de referencia a utilizar (con sus respectivos pesos) en el cálculo de las similitudes.

Con el objetivo de facilitar el entendimiento de la técnica de recomendación propuesta, en la Figura 3.10 se presenta un ejemplo de cálculo de recomendaciones.



$$recommendation(a, B, u) = \frac{1}{3}(0{,}7 * 0{,}9 + 0{,}5 * 0{,}6 + 0{,}8 * 0{,}7) = \frac{1}{3}(1{,}49) = 0{,}496$$

Figura 3.10. Ejemplo de cálculo de recomendaciones.

En la Figura 3.10 se ejemplifica la estimación del *rating* (valor de recomendación) para la instancia *a* respecto al usuario activo *u*, por parte del componente de la arquitectura propuesta encargado del filtrado colaborativo probabilístico basado en conocimiento. En dicho ejemplo, se asume que existen solo tres instancias (que representan las preferencias del usuario *u*) significativamente similares a la instancia *a* en la matriz reducida de similitudes generales, a saber, las instancias *b*, *c* y *d*. Por lo tanto, la estimación del valor de recomendación se reduce a la media aritmética ponderada de tres valores, esto es, los *ratings* compuestos calculados por el componente perfilador de usuarios para las instancias *b*, *c* y *d* respecto al usuario *u*, los cuales, como se puede apreciar, se representan en memoria en un *array* denominado *cRatings*. El valor de recomendación resultante redondeado a un decimal (0,5) se muestra en el *array* de recomendaciones.

Como se puede inferir del algoritmo de cálculo de similitudes y el algoritmo de recomendaciones presentados anteriormente en este documento, el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas propuesto en esta investigación corresponde a un método basado en conocimiento de filtrado colaborativo basado en memoria y probabilístico (Badrul Sarwar et al., 2001), dada la métrica de similitud semántica basada en ontologías y la técnica de recomendación basada en modelos de tópicos.

3.12. Interfaz de Usuario

Los usuarios (usuarios activos) pueden interactuar con el sistema de recomendación de establecimientos de alimentos y bebidas, a través de un único punto: la interfaz de usuario. Concretamente, existen tres escenarios

principales de interacción: (1) la creación de perfiles de usuarios, (2) la solicitud de recomendaciones de establecimientos de alimentos y bebidas y (3) la provisión de retroalimentación respecto a las preferencias personales, ya sea en forma de *ratings* explícitos o en forma de *check-ins* en establecimientos. Como ya se mencionó en este capítulo, la interfaz de usuario, junto con la funcionalidad de bajo nivel en la que descansa la funcionalidad del componente de la arquitectura propuesta encargado del pre-filtrado de establecimientos basado en geolocalización, se ha implementado como parte de la aplicación cliente.

En este contexto, y con el objetivo de abrir el abanico de posibilidades a una variedad de plataformas móviles, a saber, Android, iOS y Windows Phone 8, se ha empleado para ello el *framework* de desarrollo de aplicaciones móviles multiplataforma, Apache Córdova. Apache Cordova es un *framework* gratuito y *open source* que permite construir aplicaciones hibridas (aplicaciones nativas basadas en tecnologías Web) para múltiples plataformas móviles empleando el mismo código de base escrito en lenguajes de programación Web, es decir, *HTML5* (del inglés *HyperText Markup Language*), CSS3 y JavaScript. En detalle, bajo este enfoque, las aplicaciones se ejecutan dentro de un componente denominado *WebView*, el cual hace las veces de un *wrapper* para cada plataforma móvil específica (motor de renderizado de *HTML*), y descansa en *APIs* basadas en estándares que permiten acceder a las prestaciones nativas de los dispositivos móviles, como los sensores y los datos.

Como resultado, el usuario activo es libre de interactuar con el sistema utilizando prácticamente cualquier dispositivo móvil equipado con al menos una conexión a Internet, ya sea vía redes de comunicación celular, esto es *GSM* y *UMTS* (del inglés *Universal Mobile Telecommunications System*) (3G o 4G), o vía redes inalámbricas de área local (*Wireless Local Area Network*, *WLAN*), por ejemplo, Wi-Fi®. Como se explicó detalladamente en la sección 3.8 de este capítulo, las tarjetas SIM GSM/UTMS, al igual que los receptores Wi-Fi, se emplean en esta investigación, no solo para habilitar la comunicación entre la aplicación cliente y la capa del motor de recomendación de la arquitectura propuesta, sino también como fuentes de información de ubicación. No obstante, con el objetivo de proveer la información de ubicación más precisa posible, los dispositivos móviles a utilizar deben contar, preferiblemente, con sensores GPS.

Entrando en materia de los escenarios principales de interacción con el sistema, más allá de lo que se explicó en la sección 3.7 de este capítulo, aquí no se ahondará en los detalles sobre el primer escenario de interacción, esto es, la creación de perfiles de usuario; no obstante, es importante mencionar que el cuestionario de preferencias es mostrado en un componente *modal box*, y que esta vista se ha denominado “vista de preferencias”.

Como se mencionó en la sección 3.8 de este capítulo, en el segundo escenario principal de interacción, es decir, en la solicitud de recomendaciones de establecimientos de alimentos y bebidas, el usuario activo debe indicar: (1) la gente de su círculo social con la que se encuentra o con la que planea visitar los establecimientos; esto lo debe hacer escribiendo los nombres de las personas de la forma como se hace en Facebook al etiquetar amigos en publicaciones, (2) el radio en metros para acotar la búsqueda de establecimientos al área relativa a su posición geográfica. La información contextual temporal, esto es, el día de la semana y el periodo del día durante el cual el usuario activo, junto con sus acompañantes, pretende visitar los establecimientos de alimentos y bebidas, se infiere a partir del *timestamp* de la solicitud (utilizando la *API* Date de JavaScript). El tipo restante de información contextual considerado en esta investigación, es decir, la información de ubicación, es automáticamente obtenida del dispositivo móvil desde el que se solicitan las recomendaciones (utilizando la *API* de Geolocalización de Apache Cordova). La Figura 3.11 presenta el diseño de la vista de la interfaz de usuario a través de la cual el usuario puede solicitar recomendaciones al sistema.

Alternativamente, el usuario activo puede indicar una ubicación de destino seleccionando un punto directamente en un mapa, lo que se interpreta como que él/ella no está interesado en los establecimientos cercanos a su ubicación geográfica, sino en establecimientos cercanos a una ubicación específica. De manera similar, él

puede indicar el día de la semana y el periodo del día en el que pretende visitar los establecimientos, esto seleccionando las opciones correspondientes de un par de listas desplegables.

Vale la pena aclarar que, estos escenarios alternativos y el escenario principal de interacción descrito anteriormente conforman la funcionalidad de la Interfaz de Usuario correspondiente a la solicitud de recomendaciones sensibles al contexto, la cual es, formalmente, una especialización de la funcionalidad correspondiente a la solicitud de recomendaciones durante la fase de entrenamiento de la puesta en marcha del método de recomendación propuesto. En detalle, en el escenario principal de interacción de dicha funcionalidad más general, los usuarios no se ven obligados a indicar personas de su círculo social a fin de permitir al sistema inferir información social de alto nivel, a saber, situaciones sociales; igualmente, los *timestamps* de las solicitudes, si bien son recolectados, no son utilizados para inferir información temporal de alto nivel, a saber, días de la semana y periodos del día. Esto se debe a que durante dicha fase de la puesta en marcha del método de recomendación propuesto no están disponibles las características de sensibilidad a la información contextual social y temporal, por lo que la inferencia de información contextual de alto nivel no es necesaria.

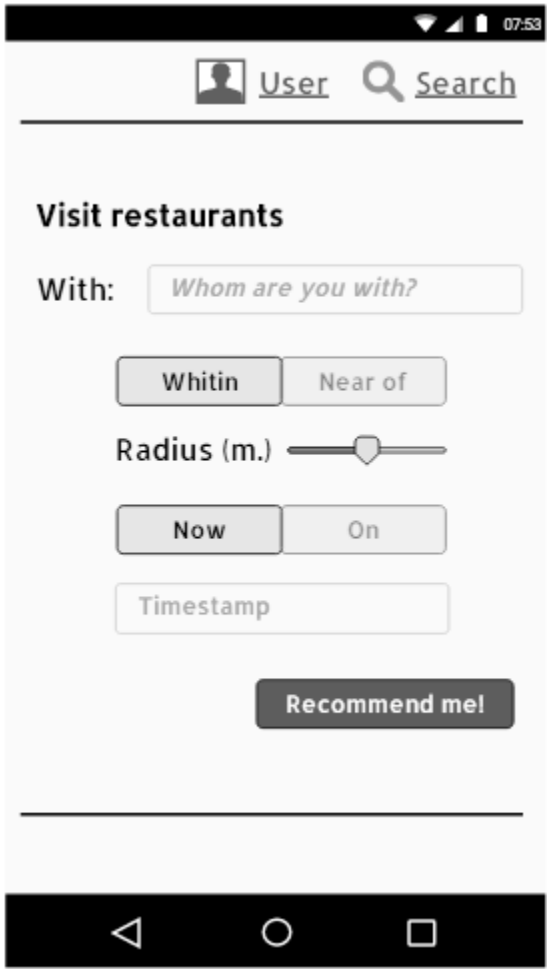


Figura 3.11. Diseño de la vista de la interfaz de usuario para la solicitud de recomendaciones.

Dados los distintos tipos de información contextual y demás parámetros requeridos, las recomendaciones calculadas como resultado de la solicitud de recomendaciones sensibles al contexto, se presentan finalmente al usuario activo a través de una vista distinta de la misma interfaz de usuario; para ello, se ha intentado

aprovechar las capacidades de *framework* de Aplicaciones Enriquecidas de Internet (*Rich Internet Applications*, *RIAs*) (Alor-Hernández, Rosales-Morales, & Colombo-Mendoza, 2015) ofrecidas por Apache Cordova. Concretamente, la interfaz de usuario se ha diseñado como una composición de diferentes patrones de interacción relativos al diseño de experiencias de navegación de contenido, en otras palabras, esta se ha diseñado como una interfaz de usuario enriquecida cuyo objetivo es permitir al usuario activo vivir experiencias enriquecidas al navegar a través de las recomendaciones resultantes para encontrar información útil sobre los establecimientos recomendados. De hecho, las recomendaciones se presentan usando principalmente un mapa -la vista de recomendaciones (ver Figura 3.12); cuando el usuario activo selecciona una recomendación del mapa, los detalles acerca del establecimiento correspondiente se muestran usando un *modal box* -la vista de detalles del establecimiento.

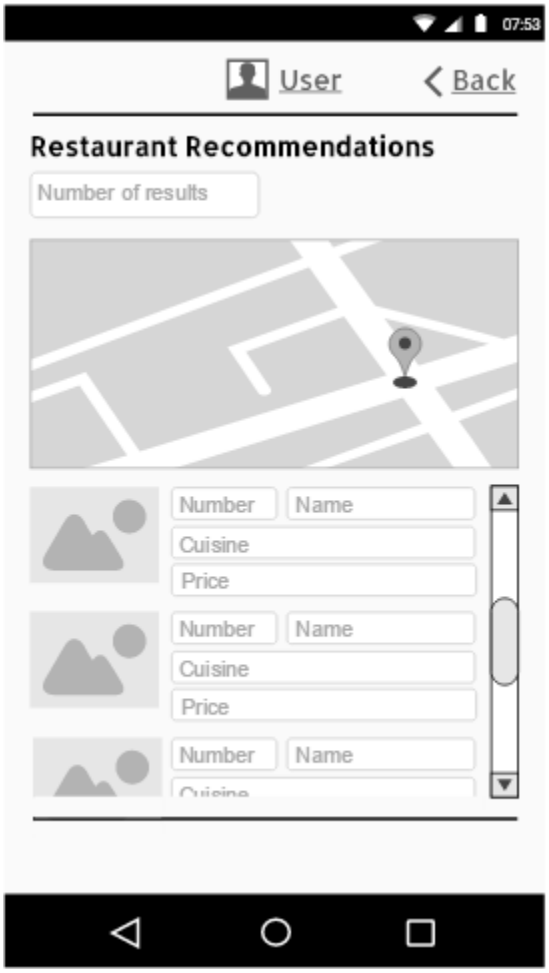


Figura 3.12. Diseño de la vista de la interfaz de usuario denominada “vista de recomendaciones”.

En lo que respecta al tercer escenario principal de interacción mencionado al inicio de esta sección, esto es, la provisión de retroalimentación respecto a las preferencias personales, es importante enfatizar que, además de la información contextual, las preferencias personales del usuario activo representan la entrada del proceso de recomendación; no obstante, estas no son establecidas por el usuario en la solicitud de recomendación, sino aprendidas por el sistema a partir de *ratings* explícitos e historiales de *check-ins*. En este contexto, los usuarios pueden proveer *ratings* explícitos para los establecimientos a través de una vista denominada “vista de exploración”, la cual es similar a la vista “Explore” en la aplicación cliente del servicio de *streaming* de música y video, Spotify®.

En la "vista de exploración" de la Interfaz de Usuario, los establecimientos se muestran agrupados por tópico, de modo que los detalles sobre un establecimiento particular se pueden observar al seleccionar un establecimiento del grupo (y, alternativamente, utilizando un componente *search box*, el cual implementa el patrón de interacción *live search*), lo que despliega la vista de detalles sobre el establecimiento. En dicha vista, el usuario puede interactuar con un componente de puntuación de cinco estrellas para proveer un *rating* explícito para el establecimiento correspondiente. De manera similar, desde dicha vista, el usuario puede hacer *check-in* en el establecimiento haciendo clic en el botón "check-in". Por otro lado, se ha aprovechado la jerarquía de clases equivalentes que representa categorías de establecimientos de alimentos y bebidas de la ontología del dominio (ver la sección 3.3 de este capítulo) para mostrar en la "vista de exploración" de la interfaz de usuario los establecimientos agrupados por estilo de cocina y por platillo, además de por tópico.

3.13. Conclusión

A primera vista, es posible, efectivamente, integrar las tecnologías de la Web Semántica y los modelos estadísticos de clases latentes a fin de definir técnicas híbridas de modelado y representación de información contextual, así como técnicas híbridas de recomendación. No obstante, la eficacia y efectividad de dichas técnicas deberá ser valorada mediante la ejecución del método de evaluación que permitirá la validación de la propuesta de esta tesis doctoral.

Gracias a la ejecución de las subtareas correspondientes a la tarea de formalización de la metodología definida en esta investigación ha sido posible, efectivamente, el descubrimiento de los puntos de convergencia entre las tecnologías de la Web Semántica y los modelos estadísticos de clases latentes, lo cuales tienen lugar, principalmente, en el modelo del contexto (distintos tipos de información contextual de bajo y alto nivel) y en el perfil de usuario basado en ontologías (preferencias por establecimientos y preferencias por tópicos) propuestos.

Asimismo, vale la pena mencionar algunas conclusiones relacionadas con contribuciones de esta investigación consideradas secundarias. Por un lado, la técnica de propagación de preferencias en jerarquías de clases en perfiles de usuario basados en ontologías, aunque limitada en funcionalidad, se prevé que, en efecto, contribuya a aliviar el problema del arranque en frío en sus modalidades de nuevo usuario e ítem nuevo, problema que tienen lugar en el método de recomendación propuesto debido al enfoque de filtrado colaborativo empleado. Por otro lado, la técnica de perfilamiento de usuarios basado en ontologías corresponde a una combinación de técnicas de retroalimentación explícita e implícita, y es, evidentemente, complementaria a la técnica de propagación de preferencias, o más bien dicho, la técnica de propagación de preferencias es complementaria a la técnica de perfilamiento de usuarios.

Finalmente, vale la pena mencionar algunas conclusiones relacionadas con la contribución general de esta tesis doctoral:

- El método de hibridación de técnicas de recomendación empleado por el método de recomendación propuesto corresponde al método de hibridación por ponderación.
- El paradigma de integración de información contextual empleado por el método de recomendación propuesto corresponde al paradigma de pre-filtrado de información contextual.

Capítulo 4 . Evaluación

4.1. Introducción

Una de las tareas primordiales del desarrollo de los sistemas de recomendación es la evaluación. Distintas técnicas principalmente procedentes de las áreas de recuperación de información y de los sistemas de soporte a las decisiones han sido propuestas a la fecha en la literatura de sistemas de recomendación. El objetivo de las evaluaciones bajo la perspectiva del área de recuperación de información es, generalmente, medir la exactitud de los algoritmos bajo distintas interpretaciones, mientras que el objetivo de las evaluaciones bajo la perspectiva del área de los sistemas de soporte a las decisiones es, generalmente, medir la calidad percibida por los usuarios. De ahí que, en la mayoría de los casos, las primeras estén dadas en forma de evaluaciones cuantitativas y las segundas en forma de evaluaciones cualitativas.

En el área de recuperación de información se han empleado tradicionalmente las métricas *precision*, *recall* y *f-measure* para medir la exactitud de la predicción del uso de los ítems, es decir la exactitud de la predicción de la probabilidad de que los ítems sean usados por los usuarios. Estas métricas están destinadas a evaluar colectivamente la eficacia y efectividad de los sistemas de recuperación de información, enfocándose principalmente en la calidad de su salida. En este contexto, *recall* se define formalmente como la fracción de los documentos relevantes para un usuario que son recuperados por un sistema; mientras que *precision* mide la fracción de los documentos recuperados por un sistema que son relevantes para un usuario.

En el contexto particular de los sistemas de recomendación, además, se considera universalmente aceptada la existencia de tres formas principales de procedimientos de evaluación, esto es, formas de organización de los métodos de evaluación (Shani & Gunawardana, 2011): (1) experimentos *offline*, (2) estudios de usuario y (3) experimentos *online*. Concretamente, los experimentos *offline* corresponden a un tipo de caso de estudio que se lleva a cabo utilizando *datasets* de preferencias de usuarios (comúnmente *ratings*) recolectados previamente en el contexto del uso de sistemas de recomendación en producción, mientras que los estudios de usuario permiten analizar el comportamiento de los usuarios al interactuar con los sistemas de recomendación, y consisten en solicitar a grupos de usuarios la realización de distintas tareas que suponen la interacción con los sistemas.

En esta tesis doctoral se ha propuesto un método de evaluación doble en forma de análisis comparativo bajo un enfoque de recuperación de información. Dicho método de evaluación se basa en las métricas *recall*, *precisión* y *f-measure*, así como en una métrica de calidad de *ratings*, a saber, la métrica *Normalized Distance-based Performance Measure* (*NDPM*), la cual permite medir la exactitud de los algoritmos de recomendación en un contexto de ordenamiento de ítems. La hipótesis detrás del método de evaluación propuesto es que la combinación de las métricas seleccionadas permitirá evaluar de una manera más exacta la eficacia y la efectividad del método de recomendación que representa la contribución de esta investigación, o más bien, del prototipo del sistema de recomendación implementado como prueba de concepto a partir de la arquitectura propuesta.

Además, con el objetivo dar lugar al análisis comparativo, se ha implementado un método de recomendación alternativo, a partir, por un lado, de una métrica de similitud de línea de base, a saber, la métrica de similitud basada en coseno ajustado (Badrul Sarwar et al., 2001) y, por otro lado, de la técnica de recomendación basada en modelos de tópicos que representa una parte integral de la contribución de esta tesis doctoral. Este método de recomendación corresponde a un método de recomendación de filtrado colaborativo híbrido basado en memoria y basado en modelos.

De la misma manera, se ha diseñado y conducido tanto un estudio de usuario como un experimento *offline* con el objetivo de llevar a cabo la evaluación, los cuales permiten evaluar, respectivamente, el método de evaluación

propuesto bajo dos escenarios distintos: un escenario de escasez de ratings y un escenario de suficiencia de ratings.

En este capítulo se realiza una extensa revisión de los enfoques de evaluación existentes más allá del área de recuperación de información, a saber, enfoques de evaluación en las Ciencias de la Computación y enfoques de evaluación en las Ciencias de la Información, así como de las técnicas o métricas de evaluación empleadas popularmente bajo dichos enfoques, diferenciando entre métricas orientadas a la medición de la exactitud bajo distintas interpretaciones y métricas orientadas a la evaluación de otros aspectos de los sistemas de recomendación.

Asimismo, se describe detalladamente el método de evaluación propuesto, incluyendo el método de recomendación de línea base, y haciendo énfasis en la contextualización de las métricas seleccionadas y en los detalles del diseño y ejecución del estudio de usuario y del experimento *offline*. Finalmente, se discuten los resultados obtenidos a partir del análisis comparativo, en términos de medidas de las métricas seleccionadas, de las listas de recomendaciones producidas por los métodos de recomendación involucrados en la evaluación, y se presentan las medidas propiamente dichas.

4.2. Enfoques de Evaluación de Sistemas de Recomendación

Partiendo de la suposición de que los sistemas de recomendación preferidos por los usuarios son aquellos que aportan a estos las recomendaciones más exactas o las más útiles, una de las tareas del ciclo de vida de los sistemas de recomendación que más interés ha recibido por parte de los investigadores en el área es la evaluación.

Debido a la naturaleza híbrida de los sistemas de recomendación, es posible hacer una primera diferenciación entre enfoques de evaluación en las Ciencias de la Computación y enfoques de evaluación en las Ciencias de la Información; más concretamente, entre enfoques en el campo del aprendizaje computacional, y enfoques en los campos de la recuperación de información y los sistemas de soporte a las decisiones (Jannach, Zanker, Ge, & Gröning, 2012).

Los enfoques de evaluación en el campo del aprendizaje computacional se han centrado, primordialmente, en la evaluación de aspectos de la calidad técnica de los sistemas, los cuales comúnmente están relacionados con el rendimiento computacional de los mismos.

Por otro lado, los enfoques de evaluación en el campo de la recuperación de información se han encaminado tradicionalmente a la evaluación de la exactitud, bien de la predicción, bien del ordenamiento de los ítems respecto a las preferencias de los usuarios; el último caso corresponde a los sistemas de recomendación en los que el objetivo no es la generación de predicciones de *ratings* sobre ítems desconocidos, sino la generación de listas ordenadas de top-n ítems desconocidos, como los enfoques de filtrado colaborativo de top-n recomendaciones.

Por último, los enfoques en el campo de los sistemas de soporte a las decisiones, se han enfocado en la medición de aspectos de calidad relacionados con las capacidades de soporte a las decisiones, los cuales generalmente determinan la calidad percibida por los usuarios.

Además, se puede pensar en los dos primeros grupos de enfoques de evaluación como en una suerte de evaluaciones cuantitativas, esto es, evaluaciones basadas en mediciones; mientras que el tercer grupo representa evaluaciones más bien cualitativas o subjetivas. De acuerdo con Kitchenham, Linkman y Law (Kitchenham, Linkman, & Law, 1997), se puede clasificar a los métodos de evaluación de herramientas o métodos de software, en un sentido genérico, en las dos categorías antes mencionadas; de modo que, mientras el objetivo de los primeros es establecer los efectos medibles del uso de las herramientas o métodos, el objetivo

de los segundos es establecer la pertinencia de las herramientas o métodos o, en otras palabras, determinar que tan bien las herramientas o métodos (en términos de sus características) cubren las necesidades de sus usuarios.

Además, dichos autores definen tres formas distintas de procedimientos de evaluación, esto es, formas de organización de los métodos de evaluación, añadiendo una dimensión a las evaluaciones como concepto: (1) experimentos formales, (2) casos de estudio y (3) encuestas.

Así, en los experimentos formales un grupo de evaluadores solicita a un grupo de sujetos que lleven a cabo determinada tarea usando las herramientas o métodos de software bajo investigación, por lo que existe una separación de roles entre los evaluadores y los sujetos de la investigación; no obstante, en el caso de los métodos de evaluación cualitativos, los sujetos (además de los evaluadores propiamente dichos) pueden llegar a actuar como evaluadores. Por otra parte, los casos de estudio consisten en la evaluación de las herramientas o métodos de software en proyectos reales por parte de miembros de dichos proyectos; existe la misma separación de roles que en el caso anterior. Las encuestas, al igual que los experimentos formales, consisten en solicitar información sobre las herramientas y métodos de software bajo investigación a un grupo de sujetos, pero en este caso se trata de sujetos con cierta experiencia en el uso de los/las mismos/mismas en proyectos pasados.

En el contexto particular de los sistemas de recomendación, sin embargo, se considera universalmente aceptada la existencia de tres formas principales de procedimientos de evaluación (Shani & Gunawardana, 2011): (1) experimentos *offline*, (2) estudios de usuario y (3) experimentos *online*.

En detalle, los experimentos *offline* corresponden a un tipo de caso de estudio que se lleva a cabo utilizando *datasets* de preferencias de usuarios (comúnmente *ratings*) recolectados previamente en el contexto del uso de sistemas de recomendación en producción; por lo tanto, no existe interacción directa con sujetos de ningún tipo, y el objetivo es simular el comportamiento de los usuarios potenciales de los sistemas a partir de los *datasets* disponibles. Si bien se trata del tipo de procedimiento de evaluación menos costoso (respecto al tiempo y otros recursos) de implementar, su aplicación está más bien restringida a la medición de la exactitud de la predicción de *ratings* o del uso de los ítems.

Más arriba en el nivel de complejidad respecto al costo de implementación, los estudios de usuario permiten obtener más información e información más valiosa acerca del rendimiento de los sistemas de recomendación, posiblemente más que los experimentos *online*, los cuales se discutirán a continuación. Concretamente, los estudios de usuario permiten analizar el comportamiento de los usuarios al interactuar con los sistemas de recomendación, y consisten en solicitar a grupos de usuarios la realización de distintas tareas que suponen la interacción con los sistemas, a fin de permitir la recolección, tanto de mediciones cuantitativas, como de mediciones cualitativas, por parte de los evaluadores. Como se puede deducir, este tipo de procedimiento de evaluación se corresponde con el concepto de “experimento formal” propuesto por Kitchenham y colaboradores.

Los experimentos *online* son el único tipo de procedimiento de evaluación que posibilita la obtención de evidencias fuertes del efecto real de los sistemas de recomendación; sin embardo, el costo de su implementación suele ser mayor aún que el de los estudios de usuario. En detalle, los experimentos *online* permiten medir el cambio que provoca en el comportamiento de los usuarios la interacción con los sistemas, y consisten en el uso de los mismos para la realización de tareas específicas por parte de usuarios reales; más allá, los experimentos *online* representan un mecanismo para medir de manera directa los objetivos globales a largo plazo de los sistemas, como, por ejemplo, los beneficios o la capacidad de retención de usuarios.

4.3. Métricas de Evaluación

Entrando en la materia de las técnicas empleadas popularmente en la evaluación de sistemas de recomendación bajo los enfoques descritos al inicio de la subsección anterior, es necesario mencionar que, los investigadores en el área de sistemas de recomendación se han enfocado tradicionalmente en la medición de la "exactitud" de los algoritmos bajo distintas interpretaciones (Herlocker, Konstan, Terveen, & Riedl, 2004). Por lo tanto, es natural hacer una primera discriminación entre métricas orientadas a la medición de la exactitud y métricas orientadas a la evaluación de otros aspectos de los sistemas de recomendación.

En lo que respecta a la categoría de las métricas orientadas a la medición de la exactitud de los algoritmos de recomendación, es posible diferenciar además entre: (1) métricas de evaluación de la exactitud en un contexto de predicción de *ratings*, (2) métricas de evaluación de la exactitud en un contexto de clasificación de ítems y (3) métricas de evaluación de la exactitud en un contexto de ordenamiento de ítems.

Si bien las dos primeras subcategorías se corresponden, respectivamente, con las primera forma previamente descritas del enfoque de evaluación de sistemas de recomendación en el área de la recuperación de información: evaluación de la exactitud de la predicción, la segunda categoría de métricas está orientada a la medición de la exactitud de la predicción del uso de los ítems, esto es, de la predicción de la probabilidad de que los ítems sean usados por los usuarios, y no a la medición de la exactitud de la predicción de *ratings*.

En lo que respecta a la categoría de las métricas orientadas a la medición de aspectos distintos a la exactitud es evidente que se pueden considerar una gran variedad de métricas de distinta naturaleza. No obstante, en esta investigación se ha considerado particularmente relevante el concepto de "satisfacción" tal y como se define genéricamente en *frameworks* de calidad de software como *Systems and software Quality Requirements and Evaluation* (*SQuaRE*). Con esto se pretende abordar el tópico de los enfoques de evaluación en el campo de los sistemas de soporte a las decisiones (ver subsección anterior). En este sentido, es importante mencionar que el tópico de los enfoques de evaluación en el campo del aprendizaje computacional está fuera del alcance de esta tesis doctoral y, por lo tanto, no se discute en este documento.

SQuaRE es un estándar internacional (ISO/IEC 25010:2011) para la evaluación genérica de la calidad de los sistemas y del software desde dos perspectivas distintas pero complementarias: calidad de desarrollo y calidad de uso; define dos modelos de calidad compuestos por características generales del software que se refinan en sub-características generales y atributos específicos. En detalle, en *SQuaRE* el concepto de "satisfacción" se relaciona con el grado en el que se cubren las necesidades de un usuario cuando el software es utilizado por este en un contexto determinado; concretamente, se considera como una de las características del software que componen el modelo de calidad de uso, y a su vez se compone de las sub-características: confianza, placer, conformidad y utilidad.

A continuación, se describen brevemente las métricas más representativas en cada categoría y subcategoría mencionada arriba.

4.3.1. Métricas de Exactitud en un Contexto de Predicción

En términos generales, el uso de este tipo de métricas consiste en la medición de la lejanía entre los *ratings* calculados por un sistema de recomendación respecto a un usuario y los *ratings* otorgados realmente por ese usuario.

En particular, la métrica *Mean Absolute Error* (*MAE*) ha sido ampliamente utilizada para la medición de la exactitud en la predicción de *ratings* en sistemas de recomendación, principalmente en sistemas de recomendación basada en filtrado colaborativo, a lo largo de una gran variedad de dominios (K. Goldberg et al., 2001). En detalle, esta métrica mide la desviación absoluta media entre los *ratings* calculados y los *ratings* reales de un usuario (ver Fórmula 4.1).

$$|\bar{E}| = \frac{\sum_{I=1}^{N} |r_{ij} - p_{ij}|}{N} \quad (4.1)$$

Donde:

- *i* representa al usuario objetivo.
- *N* es el número total de ítems que ha evaluado el usuario *i*.
- p_i es el *rating* calculado para el ítem *j* respecto al usuario *i*.
- r_i es el *rating* real otorgado por el usuario *i* al ítem *j*.

En la literatura de sistemas de recomendación se ha propuesto el uso de distintas variantes de esta métrica, por ejemplo, *Normalized Mean Absolute Error* (*NMAE*), *Mean Squared Error* (*MSE*) y *Root Mean Squared Error* (*RMSE*). Esta última, a diferencia de la métrica descrita arriba, mide la raíz cuadrada de la media aritmética de los cuadrados de los errores o desviaciones, y ha adquirido reciente popularidad gracias a la competición de algoritmos de filtrado colaborativo de la compañía Netflix, "Netflix Prize", la cual tuvo lugar en el periodo 2006-2009.

4.3.2. Métricas de Exactitud en un Contexto de Clasificación

En general, este tipo de métricas miden la frecuencia con la que un sistema de recomendación realiza predicciones correctas o incorrectas acerca de la idoneidad de un ítem (de su uso) para un usuario; por lo tanto, su aplicación resulta particularmente relevante en el contexto de los procesos de recomendación en los que las preferencias de los usuarios se expresan en términos de valores binarios, por ejemplo, en sistemas tradicionales de recomendación basada en contenido y sistemas de filtrado colaborativo basado en técnicas de clasificación binaria.

Tradicionalmente, las métricas *Precision* y *Recall* han sido utilizadas colectivamente para la evaluación de los sistemas de recuperación de información. Formalmente, estas métricas están orientadas a la evaluación de la eficacia y efectividad de estos sistemas respecto a la calidad del resultado de la tarea de recuperación de información (Salton & McGill, 1986). En el contexto de la evaluación de sistemas de recomendación su utilización se remonta al final de la década de 1990 (Basu et al., 1998); (Billsus & Pazzani, 1998).

En detalle, *Recall* mide la fracción de los documentos relevantes para un usuario que son recuperados por un sistema; mientras *Precision* mide la fracción de los documentos recuperados por un sistema que son relevantes para un usuario (ver Fórmulas 4.2 y 4.3). Evidentemente, este planteamiento asume que el usuario es capaz de juzgar los documentos como "relevantes" o "no relevantes" de acuerdo a sus preferencias.

Dado que las medidas de *precision* y *recall* adquieren mayor relevancia cuando se interpretan en conjunto, existen métricas que permiten resumir estas medidas en una sola medida. Por ejemplo, la métrica *F-measure* o *F-score* mide la media armónica de las medidas de *precision* y *recall* obtenidas para una misma consulta. Formalmente, la media armónica de dichas medidas representa solo uno de los posibles casos de la métrica F_β*-measure*, donde β representa un número real no negativo (ver Fórmula 4.4); no obstante, debido a que se trata del caso tradicional o balanceado, se suele emplear el término "*F-measure"* para designarlo, siendo que el término correcto es "F_1*-measure*" o "F_1*-score*".

$$Precision = \frac{|\{docRel\} \cap \{docRec\}|}{|\{docRec\}|} \quad (4.2)$$

$$Recall = \frac{|\{docRel\} \cap \{docRec\}|}{|docRel|} \quad (4.3)$$

$$F_1 - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

Donde:

- *docRel* es el número de documentos que son juzgados por el usuario como relevantes en el contexto de una consulta.
- *docRec* es el número de documentos recuperados por el sistema en el contexto de la misma consulta.

4.3.3. Métricas de Exactitud en un Contexto de Ordenamiento

En términos generales, la utilización de este tipo de métricas consiste en la medición del grado de discrepancia entre el *ranking* de ítems producido por un sistema de recomendación respecto a un usuario y el *ranking* real de preferencias de dicho usuario. Vale la pena resaltar que, este planteamiento asume que es posible obtener un *ranking* de referencia, ya sea a partir de *ratings* proporcionados explícitamente por usuarios o a partir de *ratings* implícitos derivados de datos de uso del sistema.

A diferencia de las métricas en la categoría anterior, las métricas en esta categoría son especialmente idóneas en el contexto de los procesos de recomendación en los que las preferencias de los usuarios no se expresan en términos de valores binarios y la tarea de recomendación propiamente dicha consiste en la generación de listas ordenadas de top-n ítems; por ejemplo, en el contexto de los enfoques de filtrado colaborativo de top-n recomendaciones.

Dependiendo de si se debe penalizar al sistema o no por dar a uno de dos ítems un rango más alto cuando ambos tienen el mismo rango en el *ranking* de referencia (*rankings* de referencia con rangos repetidos), es posible emplear dos tipos distintos de métricas para la medición de la exactitud del *ranking* resultante respecto a dicho *ranking*.

En el primer caso, las métricas de correlación, especialmente métricas de correlación por rangos como el *coeficiente de correlación de Spearman* y el *coeficiente de correlación de Kendall* (Celma, 2010), pueden ser útiles; no obstante, su uso en este contexto no se ha extendido demasiado, a diferencia de lo que sucede particularmente con la primera métrica en el contexto del cálculo de similitudes en procesos de recomendación basada en filtrado colaborativo.

En el segundo caso destaca el uso de la métrica *Normalized Distance-based Performance Measure* (*NDPM*), que fue propuesta originalmente por Yao en el área de recuperación de información (Yao, 1995), y fue introducida al área de sistemas de recomendación por Balabanović & Shoham un par de años después (Balabanović & Shoham, 1997). Esta métrica es similar a las métricas nombradas anteriormente; sin embargo, permite analizar más fielmente el efecto de los *rankings* de preferencia con rangos repetidos (ver Fórmula 4.5).

No penalizar al sistema por dar prioridad a un ítem sobre otro cuando ambos tienen el mismo rango en el *ranking* de referencia parece ser el funcionamiento deseado de las métricas de medición de exactitud de *rankings* en el escenario particular de la evaluación de sistemas de recomendación que presentan al usuario *rankings* de ítems con rangos únicos. Considerando que este tipo de sistemas de recomendación son una herramienta recurrente a lo largo de un sin número de sitios Web populares en la actualidad, por ejemplo, sitios Web de servicios de *streaming* de contenido multimedia como Google Play Music y YouTube Gaming, y sitios Web de redes sociales como ResearchGate y SoundCloud, es posible pensar que la aplicabilidad de esta métrica es mayor que la de las métricas *coeficiente de correlación de Spearman* y *coeficiente de correlación de Kendall*.

$$NDPM = \frac{2C^{-} + C^{u}}{2C^{i}} \quad (4.5)$$

Donde:

- C^{-} es el número de relaciones de preferencia contradictorias entre el *ranking* producido por el sistema y el *ranking* de referencia, esto es, el número de relaciones en las que, según el *ranking* producido por el sistema, el ítem 1 tiene un rango mayor que el ítem 2, pero según el *ranking* de referencia, el ítem 2 debe tener un rango más alto que el ítem 1.
- C^{u} es el número de relaciones de compatibilidad, esto es, el número de relaciones en las que, según el *ranking* de referencia, el ítem 1 debe tener un rango mayor que el ítem 2, pero según el *ranking* producido por el sistema, ambos ítems tienen el mismo rango.
- C^{i} es el número de relaciones de prioridad, esto es, pares de ítems en los que uno de los ítems tiene un rango más alto que el otro.

4.3.4. Otras Métricas (Satisfacción)

En aras de la brevedad, a continuación se discuten únicamente las sub-características de calidad "utilidad" y "confianza", las cuales forman parte de la característica "satisfacción" según el estándar internacional de evaluación de la calidad de sistemas y software, *SQuaRE*. En este sentido, si bien *SQuaRE* es lo suficientemente genérico como para usarse en la evaluación de prácticamente cualquier tipo de sistema o software, existen *frameworks* relacionados enfocados especialmente en la evaluación de sistemas de recomendación que consideran los factores antes mencionados.

Recommender systems' Quality of user experience (*ResQue*) es un *framework* de evaluación de usabilidad basado en modelos y centrado en el usuario que captura una serie de características consideradas esenciales en los sistemas de recomendación. En detalle, *ResQue* comprende una serie de construcciones agrupadas en cuatro dimensiones de evaluación: (1) cualidades del sistema percibidas por el usuario, (2) creencias del usuario como resultado de las cualidades percibidas, (3) actitudes subjetivas del usuario y (4) intenciones de comportamiento del usuario; cada construcción es un conjunto de escalas de medición y preguntas procedentes de un cuestionario de plantilla que actúan como instrumento de medición.

En *ResQue*, la sub-característica de calidad "utilidad" corresponde a una de las construcciones de la dimensión "creencias del usuario", y comprende las escalas "soporte a la decisión" y "calidad de la decisión". En la Figura 4.1 se muestran las preguntas propuestas en *ResQue* para la medición de los factores correspondientes a dichas escalas.

- The recommended items effectively helped me find the ideal product.
- The recommended items influence my selection of products.
- I feel supported to find what I like with the help of the recommender.
- I feel supported in selecting the items to buy with the help of the recommender.

Figura 4.1. Preguntas para la medición del "soporte a la decisión" y la "calidad de la decisión" en *ResQue*.

Por otra parte, la sub-característica "confianza" corresponde a una construcción de la dimensión "actitudes del usuario", la cual comprende una única escala del mismo nombre. La Figura 4.2 muestra las preguntas propuestas en *ResQue* para la medición del factor correspondiente a dicha escala.

- The recommended items made me confused about my choice (reverse scale).
- The recommender can be trusted.

Figura 4.2. Preguntas para la medición de la "confianza" según *ResQue*.

Inspirados en el concepto homónimo de la literatura de redes sociales, los investigadores del campo de sistemas de recomendación introdujeron y desarrollaron el concepto de "confianza" (o reputación) durante la primera mitad de los años 2000 bajo el argumento de que, como en el mundo real, las personas suelen confiar más en las sugerencias de unas personas que en las de otras al momento de tomar decisiones acerca de productos o servicios desconocidos (Montaner, López, & de la Rosa, 2002); (Massa & Bhattacharjee, 2004); (O'Donovan & Smyth, 2005).

Originalmente, ellos propusieron extender las técnicas de recomendación basada en filtrado colaborativo de tal manera que, además de las similitudes entre usuarios o ítems, estás considerasen el grado de confianza entre usuarios. Como resultado, los sistemas de recomendación serían capaces de capturar fielmente aquellos casos en los que un usuario "x" comparte preferencias generales con un usuario "y" y, sin embargo, no se puede considerar un predictor confiable de la preferencia del usuario "y" por un ítem específico.

Esta interpretación del concepto de "confianza" no debe ser confundida con la interpretación dada en el contexto de la evaluación de los sistemas de recomendación, la cual está en el punto de mira de la presente subsección. En este contexto, la confianza tiene que ver con la seguridad del usuario de cara a las recomendaciones provistas por el sistema; por lo que no se trata de un atributo técnico de la calidad del sistema, sino más bien de un atributo de la calidad percibida por el usuario.

En la literatura de sistemas de recomendación se han propuesto diversas estrategias para la "construcción" de la confianza de los usuarios respecto a los sistemas; una de las más populares es la provisión de explicaciones acerca de los ítems recomendados (Tintarev & Masthoff, 2011); (Scheel, Castellanos, Lee, & Luca, 2012); (Sharma & Ray, 2016), que parece estar directamente relacionada con otro factor recurrente de la calidad de los sistemas desde la perspectiva de los usuarios: la "transparencia" respecto a su funcionamiento interno. Este factor es de hecho considerado en *ResQue*; específicamente, corresponde a una de las escalas de una construcción paralela a la construcción que corresponde al factor "utilidad" dentro de la dimensión "creencias del usuario".

En (Cramer et al., 2008) los autores presentaron un estudio de usuario destinado a la evaluación de los efectos de la transparencia en la confianza y la aceptación como factores de la calidad percibida por los usuarios de los sistemas de recomendación. En este contexto, la aceptación fue medida en dos dimensiones diferentes: (1) la aceptación de las recomendaciones propiamente dichas y (2) la aceptación o idoneidad del sistema como un todo en el escenario específico de la sobrecarga de información. En detalle, los autores identificaron una serie de medidas o escalas (entre ellas medidas correspondientes a los factores de calidad antes mencionados) e instrumentos de medición correspondientes (entre ellos entrevistas y cuestionarios). En la Figura 4.3 se presentan las preguntas propuestas en este trabajo para la medición del factor de calidad "confianza".

- I trust the system.
- I can depend on the system.

Figura 4.3. Preguntas para la medición de la "confianza" según (Cramer et al., 2008).

4.4. Método de Evaluación Propuesto

En esta investigación se ha propuesto un método de evaluación doble basado en un enfoque de Ciencias de la Información; específicamente, se trata de un enfoque de recuperación de información basado en: (1) métricas de exactitud en un contexto de clasificación de ítems y (2) métricas de exactitud en un contexto de ordenamiento de ítems.

En detalle, por un lado, se han seleccionado las métricas de evaluación tradicionales del campo de la recuperación de información, *recall*, *precision* y f_1*-measure*; por otro lado, se ha seleccionado la métrica de calidad de *rankings*, *NDPM*, la cual procede también de dicho campo.

La hipótesis detrás del método de evaluación propuesto es que la combinación de las métricas seleccionadas permitirá evaluar de una manera más exacta la eficacia y la efectividad del método de recomendación sensible al contexto de establecimientos de alimentos y bebidas que representa la contribución de esta investigación. Asimismo, con el objetivo de llevar a cabo dicha evaluación de la manera más confiable posible, se ha implementado un método de recomendación de filtrado colaborativo híbrido, basado en memoria y basado en modelos (método de recomendación de línea base); dicho método radica tanto en una métrica de similitud de línea de base, a saber, la métrica de similitud basada en coseno ajustado (Badrul Sarwar et al., 2001), como en la técnica de recomendación basada en modelos de tópicos, que representa una parte integral de la contribución de esta tesis doctoral.

La métrica de similitud basada en coseno permite determinar el grado de similitud entre dos usuarios o ítems representándolos como vectores de *ratings* y calculando el coseno del ángulo entre dichos vectores. En el contexto de los enfoques de filtrado colaborativo de vecindario o de top-n recomendaciones basados en ítems, esta métrica requiere que se identifiquen primero todos los usuarios que hayan evaluado ambos ítems en el par de ítems *i*, *j* a comparar; no obstante, durante el cálculo propiamente dicho no se consideran las diferencias en la escala de *ratings* entre los distintos usuarios. La métrica basada en coseno ajustado, tal y como fue propuesta originalmente en (Badrul Sarwar et al., 2001), resuelve este inconveniente de la siguiente manera: por cada usuario *u* identificado, sustrae el correspondiente *rating* promedio $\bar{R}_u$ a cada *rating* individual $R_{u,i}$ y $R_{u,j}$ (ver Fórmula 4.6).

$$sim(i,j) = \frac{\sum_{u \in U}\left(R_{u,i} - \bar{R}_u\right)\left(R_{u,j} - \bar{R}_u\right)}{\sqrt{\sum_{u \in U}\left(R_{u,i} - \bar{R}_u\right)^2}\sqrt{\sum_{u \in U}\left(R_{u,j} - \bar{R}_u\right)^2}} \quad (4.6)$$

Donde:

- *i, j* es el par de ítems a comparar.
- *u* representa a cada uno de los usuarios que ha evaluado ambos ítems en el par de ítems *i*, *j*.
- $\bar{R}_u$ es el promedio de todos los *ratings* otorgados por el usuario *u*.
- $R_{u,i}$ y $R_{u,j}$ son los *ratings* otorgados por el usuario *u* a los ítems *i* y *j*, respectivamente.

La métrica de similitud basada en coseno ajustado es sin duda una de las métricas de similitud más populares en la categoría de técnicas de filtrado colaborativo basado en memoria; en distintos escenarios se ha demostrado su rendimiento competitivo (en términos de distintos tipos de métricas de evaluación) frente a otras métricas populares en la misma categoría de técnicas de filtrado colaborativo, como la métrica *coeficiente de correlación de Pearson* (Candillier, Meyer, & Boullé, 2007); (Ekstrand, Ludwig, Konstan, & Riedl, 2011).

4.4.1. Contextualización de las Métricas Seleccionadas

En lo que respecta a las métricas tradicionales del campo de la recuperación de información, en este método de evaluación, *recall* (como medida) se ha interpretado como la fracción de los establecimientos de alimentos

y bebidas relevantes para un usuario que son recomendados por el método de recomendación propuesto y por el enfoque de recomendación de línea base seleccionado. Como se explicará claramente en la siguiente subsección, un establecimiento de alimentos y bebidas se considera relevante para un usuario, si este lo juzga como tal respecto a sus preferencias en un contexto dado, a saber, en una posición geográfica, una situación social, un día de la semana y un periodo del día determinados.

Por otro lado, *Precision* (como medida) se ha interpretado como la fracción de los establecimientos de alimentos y bebidas recomendados (por el método de recomendación propuesto y por el método de recomendación de línea base seleccionado) que son relevantes para un usuario.

Las Fórmulas 4.7 y 4.8 representan, respectivamente, las fórmulas que permiten calcular las medidas de *recall* y *precision* bajo las interpretaciones dadas.

$$Recall = \frac{correctRecomEstablishments}{totalRelevantEstablishments} \quad (4.7)$$

Donde:

- *correctRecomEstablishments* es el número de establecimientos de alimentos y bebidas considerados relevantes para un usuario que son efectivamente recomendados por el método de recomendación propuesto y por el enfoque de recomendación de línea base seleccionado.
- *totalRelevantEstablishments* es el número total de establecimientos de alimentos y bebidas considerados relevantes para el usuario.

$$Precision = \frac{correctRecomEstablishments}{totalRecomEstablishments} \quad (4.8)$$

Donde:

- *correctRecomEstablishments* tiene la misma interpretación dada en la Fórmula 4.7.
- *totalRecomEstablishments* es el número total de establecimientos de alimentos y bebidas recomendados al usuario por el método de recomendación propuesto y por el enfoque de recomendación de línea base seleccionado.

Por último, la medida de f_1*-measure* se puede calcular utilizando la Fórmula 4.4.

En lo que respecta a la métrica de calidad de *rankings*, *NDPM*, en esta investigación se ha propuesto su uso respecto a listas de top-5 establecimientos (*rankings* de referencia). Los detalles acerca de la generación de estas listas de establecimientos se presentan en el contexto de los detalles sobre el experimento *offline* que ha sido diseñado y ejecutado a fin de llevar a cabo la evaluación propuesta.

4.4.2. Diseño y Ejecución del Estudio de Usuario

Durante la fase final de esta tesis doctoral, se invitó a estudiantes del grado en Informática de la Universidad de Murcia a participar en la prueba del prototipo del sistema de recomendación en el que se implementa el método de recomendación propuesto; para ello se distribuyó una convocatoria de voluntariado a través de las listas de correo electrónico de la universidad. Como requisito, los estudiantes debían ser residentes de la ciudad de Murcia, así como visitantes frecuentes de establecimientos de alimentos y bebidas en dicha ciudad.

Finalmente se seleccionó un total de 48 estudiantes (de aquí en adelante llamados “participantes”) que cumplían con dicho requisito. Seguidamente, se conformó un entorno de pruebas acorde a las necesidades de los preparativos de la prueba; para ello se habilitaron dos laboratorios de cómputo de la Facultad de Informática de la Universidad de Murcia, cada uno de ellos equipado con un total de 24 ordenadores capaces de acceder a un

prototipo Web de la aplicación cliente de la arquitectura de software a partir de la cual se ha pretendido implementar el método de recomendación propuesto -prototipo de aplicación cliente.

Una vez listo el entorno de pruebas, se solicitó a los participantes que se reuniesen en el mismo para interactuar con el prototipo de aplicación cliente y crear las correspondientes cuentas de usuario para la aplicación cliente. De acuerdo con el procedimiento de construcción de perfiles de usuario explicado en la sección 3.7 en el capítulo anterior, se solicitó a los participantes que respondiesen al cuestionario de preferencias presentado por el prototipo de aplicación cliente. Evidentemente, para el caso específico de esta prueba, la residencia de todos los participantes (usuarios) se fijó en la ciudad de Murcia, España.

De acuerdo con lo explicado en la sección previamente referida de este documento, las preferencias de los participantes por los establecimientos, esto es, los perfiles de usuario, se aprendieron automáticamente a partir de la propagación de los *ratings* explícitos otorgados por los participantes, a través del cuestionario de preferencias, a los estilos de cocina y platos. En detalle, debido a la inexistencia de historiales de *check-ins* asociados a los participantes, en esta fase de la evaluación las preferencias se obtuvieron únicamente a partir de *ratings* explícitos propagados a los establecimientos desde los estilos de cocina y platos (formalmente desde las clases que los representan) a través de las jerarquías de clases de la ontología del dominio. Como resultado, el componente de la arquitectura propuesta encargado del perfilamiento de usuarios recolectó un total de 121 establecimientos de alimentos y bebidas de la ciudad de Murcia, España, así como un total de 912 *ratings* explícitos.

Obsérvese que, los *ratings* recolectados fueron utilizados por el componente de la arquitectura propuesta encargado de calcular las similitudes semánticas para calcular las similitudes entre los establecimientos en el caso específico del enfoque de recomendación de línea base; así se aseguró que dicho enfoque no sería evaluado en condiciones de escasez de datos, lo que le habría representado una desventaja considerable a efectos del análisis comparativo a llevar a cabo entre el método de recomendación propuesto y el enfoque de recomendación de línea base.

En lo que respecta a las características de sensibilidad a la información contextual, específicamente a la información social y temporal, del método de recomendación propuesto, debido a la inexistencia de información histórica (historiales de *check-ins*) que sirviera de base para la construcción del modelo *LDA* de información contextual de alto nivel durante esta fase de la evaluación, estas no pudieron ser evaluadas sino en la fase posterior de la misma.

Esto se debe a que, como se menciona en la sección 3.7 del capítulo anterior, esta fase de la evaluación corresponde a la fase de entrenamiento de la puesta en marcha del método de recomendación propuesto. De hecho, durante dicha fase de entrenamiento, no están disponibles las características de sensibilidad a la información contextual social y temporal (por lo que, de acuerdo con el procedimiento de creación de perfiles de usuario, no se solicitó a los participantes el establecimiento de sus relaciones sociales después de la presentación del cuestionario de preferencias); sin embargo, no sucede lo mismo con la característica de sensibilidad a la información contextual geográfica, la cual está parcialmente disponible aún en las condiciones de la fase de entrenamiento.

Con el objetivo de permitir a los participantes interactuar con el prototipo de aplicación cliente mientras se habilitaba la característica de sensibilidad a la información contextual geográfica del prototipo del sistema, se seleccionaron dos ubicaciones geográficas de la ciudad de Murcia de posible relevancia para la prueba: (1) el edificio “Real Casino” en la calle “Trapería” del centro de la ciudad y (2) el centro comercial “Zig Zag” de la avenida “Juan Carlos I”, el cual se encuentra aproximadamente a 2 kms. del centro de la ciudad. Asimismo, se seleccionaron dos conjuntos de establecimientos desconocidos por los participantes, cada uno compuesto por 30 establecimientos disponibles en un área geográfica relativa a una de las ubicaciones antes mencionados,

en donde las áreas geográficas estuvieron dadas por un casquete esférico cuya base tenía un radio de longitud igual a 750 m. Las Figuras 4.4 y 4.5 muestran respectivamente los 30 establecimientos de alimentos y bebidas mejor calificados dentro del área geográfica relativa al edificio “Real Casino” (Calle Trapería 18, Murcia, España) según Yelp y Foursquare, respectivamente.

Figura 4.4. Los 30 establecimientos de alimentos y bebidas mejor calificados en un radio de 750m. alrededor del edificio “Real Casino” de Murcia según Yelp.

Figura 4.5. Los 30 establecimientos de alimentos y bebidas mejor calificados en un radio de 750m. alrededor del edificio “Real Casino” de Murcia según Foursquare.

De acuerdo con lo anterior, se formaron dos grupos de 24 participantes cada uno; uno de los grupos fue asignado aleatoriamente a una de las ubicaciones geográficas previamente seleccionadas, de modo que el segundo grupo fue asignado a la ubicación geográfica restante. Para finalizar con los preparativos de la prueba, se pidió a cada participante que juzgase cada uno de los establecimientos en el conjunto de establecimientos correspondiente a la ubicación geográfica que le fue asignada con anterioridad como relevante o no relevante para sus preferencias personales.

Una vez hecho esto, se pudo dar inicio a la prueba propiamente dicha. En este contexto, es necesario mencionar que, a efectos prácticos, no se solicitó a los participantes que se dirigiesen a las ubicaciones geográficas predefinidas; en cambio, estas solo se simularon en el prototipo de aplicación cliente. En concreto, se requirió a los participantes que interactuasen con el prototipo de aplicación cliente a fin de solicitar recomendaciones; para ello fue necesario únicamente que, de acuerdo con los preparativos de la prueba, el radio de búsqueda de establecimientos se fijase en 750 m.

Tal y como lo dicta el escenario principal de interacción con la aplicación cliente correspondiente a la solicitud de recomendaciones al margen del enfoque de recomendación sensible al contexto propuesto en esta investigación, no se requirió a los participantes que, como parte de las solicitudes, indicasen nombres de personas de su círculo social, ya que no era necesario que el prototipo del sistema infiriese información social de alto nivel, a saber, situaciones sociales; de igual manera, no era relevante la inferencia de información temporal de alto nivel, a saber, días de la semana y periodos del día, a partir de los *timestamps* de las solicitudes, los cuales, no obstante, son recolectados invariablemente (ver sección 3.12 en el capítulo anterior).

En este punto de la prueba, el método de recomendación propuesto generó una lista de top-n recomendaciones por cada participante, a partir de la predicción, basada en el correspondiente perfil de usuario, de *ratings* para los ítems juzgados por él como relevantes o no relevantes para sus preferencias personales durante los preparativos de la prueba. Igualmente, se generó una segunda lista de recomendaciones mediante el uso del enfoque de recomendación de línea base descrito en la subsección anterior. Los resultados del método de recomendación propuesto fueron finalmente comparados en términos de medidas de *recall*, *precisión* y f_1-*measure* con los resultados del enfoque de recomendación de línea base; los resultados de dicho análisis comparativo, así como las medidas propiamente dichas, se discuten detalladamente en la sección 4.5 de este capítulo.

En la Tabla 4.1 se muestra la lista de las propiedades de objeto consideradas en el cálculo de las similitudes semánticas entre los establecimientos durante la prueba, así como la relevancia o peso dado a cada una de ellas, tal y como lo requiere la métrica de similitud basada en ontologías que representa una parte sustancial de la contribución de esta tesis doctoral (ver Fórmula 3.10 del Capítulo 3).

Nombre	Tipo	Relevancia
servesCuisine	Propiedad de objeto	0,45
servesCuisineDish	Propiedad de objeto	0,45
priceTier	Propiedad de tipo de dato	0,10

Tabla 4.1. Propiedades de objeto seleccionadas para el cálculo de las similitudes semánticas entre los establecimientos de alimentos y bebidas.

4.4.3. Diseño y Ejecución del Experimento Offline

Con el objetivo de evaluar las características de sensibilidad a la información contextual social y temporal, se llevó a cabo un experimento *offline* que permitió construir un modelo *LDA* de información contextual de alto nivel "sintético". Este experimento representó la segunda fase de la ejecución del método de evaluación propuesto en el sentido de que, se basó en el *dataset* obtenido gracias a la interacción de los participantes con el prototipo de aplicación cliente durante el estudio de usuario reseñado anteriormente.

En detalle, se establecieron relaciones sociales aleatorias para cada uno de los perfiles de usuario en dicho *dataset* de modo que fuera posible crear *check-ins* que involucraran aleatoriamente a los usuarios representados por dichos perfiles (48 usuarios), a las personas en sus relaciones sociales y, evidentemente, a los establecimientos de alimentos y bebidas en el mismo *dataset* (121 establecimientos existentes en la ciudad de Murcia). Se creó un total de 726 *check-ins*, esto es, un promedio de seis *check-ins* por establecimiento. El *dataset* extendido resultante fue utilizado a modo de *dataset* de entrenamiento por el componente de la arquitectura propuesta encargado del perfilamiento de usuarios para actualizar las preferencias de estos por los establecimientos, es decir, para calcular *ratings* compuestos para los establecimientos respecto a los usuarios.

Por otro lado, el conjunto de los 60 establecimientos existentes en las dos áreas (de radio igual 750 m.) relativas a las dos posiciones geográficas seleccionadas para los propósitos del estudio de usuario fue usado a modo de *dataset* de prueba en este experimento *offline*. Para ello, se simularon ocho contextos socio-temporales a partir de todas las combinaciones posibles entre los valores "Friends" y "Family" para la variable contextual "SocialSituation", los valores "Tuesday" y "Thursday" para la variable contextual "DayOfWeek" y los valores "Afternoon" y "Night" para la variable contextual "PeriodOfDay" (ver Tabla 4.2). De acuerdo con esto, se formaron ocho grupos de seis usuarios cada uno; cada grupo fue asignado a uno de los contextos simulados; además, de cada grupo, tres usuarios fueron asignados aleatoriamente a una de las áreas geográficas antes mencionadas, y la otra mitad fue asignada al área geográfica restante. De este modo, cada área geográfica quedó finalmente asociada a 24 usuarios del total de 48 usuarios en el experimento, y cada contexto simulado quedó asociado a ambas áreas geográficas.

Id.	**Variables Contextuales**		
	SocialSituation	**DayOfWeek**	**PeriodOfDay**
1	Friends	Tuesday	Afternoon
2	Family	"	"
3	Friends	"	Night
4	Family	"	"
5	Friends	Thursday	Afternoon
6	Family	"	"
7	Friends	"	Night
8	Family	"	"

Tabla 4.2. Contextos socio-temporales simulados en el experimento offline.

Asimismo, a partir del *dataset* extendido fue posible a dicho componente de la arquitectura aprender las preferencias de los usuarios por los tópicos del modelo *LDA* sintético empleando las Fórmulas 3.2 y 3.5 descritas en el capítulo anterior. En este sentido, cabe mencionar que para que fuera posible al componente de la arquitectura propuesta encargado del descubrimiento de tópicos la construcción del modelo *LDA* sintético, se tomaron las siguientes medidas en lo que respecta al uso del algoritmo para generación de modelos *LDA* para análisis de estabilidad (ver Figura 2 en el capítulo anterior).

- El rango de valores para la estimación del número de tópicos *k*, el cual a su vez indica el número de modelos de referencia a generar, se estableció en [2,10].
- El número de modelos de prueba a generar se estableció en 50
- El porcentaje de documentos en el corpus subyacente (los historiales de *check-ins* ocurridos en los establecimientos derivados del *dataset* extendido) a usar para la generación de dichos modelos, se estableció en 81,8181% (99 historiales).

Posteriormente, gracias al uso del algoritmo representado por la Fórmula 3.1 (ver capítulo anterior), el valor óptimo de *k* fue resuelto en 5 por dicho componente de la arquitectura; por lo tanto, el modelo *LDA* de referencia para el que el número de tópicos *k* se estableció en 5 se tomó como el modelo *LDA* definitivo en este experimento.

Una vez que el modelo *LDA* sintético fue construido, y los perfiles de usuario fueron actualizados, el método de recomendación propuesto fue capaz de producir una lista de top-n recomendaciones por cada usuario, a partir de la predicción, basada en el correspondiente perfil de usuario, de *ratings* para los establecimientos en el área geográfica a la que fue asignado, según el correspondiente contexto (socio-temporal) simulado. Como en el caso del estudio de usuario, se empleó además el enfoque de recomendación de línea base a fin de generar una segunda lista de recomendaciones que diese pie al análisis comparativo propuesto. De hecho, por cada usuario en el experimento se simuló una solicitud al prototipo del sistema; en ella se incluyó explícitamente la información contextual de alto nivel y la ubicación geográfica correspondientes, además del radio de búsqueda predefinido (750 m.).

Una vez generadas las listas de recomendaciones, se pidió a un grupo de 3 estudiantes del grado en Informática de la Universidad de Murcia (estudiantes ajenos al estudio de usuario antes reseñado y referidos de aquí en adelante como "expertos") que juzgasen cada par de estas correspondiente a un usuario distinto con el objetivo de clasificar manualmente cada establecimiento recomendado como relevante o no relevante para sus preferencias dado el contexto socio-temporal simulado al que fue asignado. Además, a fin de identificar establecimientos que, si bien eran relevantes para la correspondiente solicitud de recomendación, no fueron recomendados por el prototipo del sistema, se analizó, el conjunto de establecimientos en el área geográfica a la que cada usuario fue asignado; asimismo, a partir del análisis de los conjuntos predefinidos de establecimientos se identificaron y ordenaron en orden decreciente de prioridad, independientemente de si estaban o no presentes en las listas de recomendaciones producidas, los cinco establecimientos más relevantes para cada solicitud de recomendación, esto es, las listas de top-5 establecimientos (ver subsección anterior). Finalmente, las listas de recomendaciones fueron comparadas en términos de medidas de *recall*, *precision*, f_1-*measure* y *NDPM*; las medidas propiamente dichas y los resultados del análisis comparativo se discuten en detalle a continuación.

4.5. Resultados y Discusión

Con el objetivo de calcular las medidas de *recall*, *precision*, f_1-*measure* y *NDPM* para el método de recomendación propuesto y para el método de recomendación de línea base, se obtuvieron los valores para las variables de las Fórmulas 4.4, 4.5, 4.7 y 4.8; para ello se recolectaron ciertos datos relacionados con las recomendaciones producidas por ambos métodos de recomendación durante el estudio de usuario y el experimento *offline*.

Las Tablas 4 y 5 muestran los resultados de dichos cálculos correspondientes respectivamente al estudio de usuario y el experimento *offline* para el caso del método de recomendación propuesto; los resultados para el caso del método de recomendación de línea base se discuten más adelante en esta subsección. Además, los valores de las variables antes mencionadas obtenidos para cada participante y usuario simulado se muestran respectivamente en las Tablas 4.3 y 4.4. En detalle, la columna "Relevantes - Total" representa la variable "totalRelevantEstablishments"; la columna "No Relevantes - Total" representa lo opuesto, es decir, el número de establecimientos juzgados como no relevantes por los participantes o expertos. La columna "Relevantes - Correctos" representa la variable "correctRecommEstablishments"; la columna "No Relevantes - Errores" representa lo opuesto, esto es, el número de establecimientos juzgados como no relevantes por los participantes o expertos que fueron recomendados por los métodos de recomendación. Por último, la columna "Recomendados" representa la variable "totalRecommEstablishments". En el caso particular de la Tabla 4.4, la columna "Id. Contexto" representa los identificadores de los contextos socio-temporales simulados (ver Tabla

4.2 en la subsección 4.4.3 de esta sección); la columna "Área Geo." representa los identificadores de las áreas geográficas predefinidas en el estudio de usuario y retomadas en el experimento *offline*.

Área Geo.	Relevantes		No Relevantes		Recomen.	*Precision*	*Recall*	*F-measure*
	Tot.	Correct.	Tot.	Err.				
1	14	9	16	4	13	0.692307692	0.642857143	0.666666667
	17	12	13	5	17	0.705882353	0.705882353	0.705882353
	20	15	10	5	20	0.75	0.75	0.75
	18	13	12	5	18	0.722222222	0.722222222	0.722222222
	13	11	17	4	15	0.733333333	0.846153846	0.785714286
	16	12	14	5	17	0.705882353	0.75	0.727272727
	11	8	19	2	10	0.8	0.727272727	0.761904762
	10	8	20	2	10	0.8	0.8	0.8
	11	7	19	3	10	0.7	0.636363636	0.666666667
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	9	7	21	2	9	0.777777778	0.777777778	0.777777778
	9	7	21	2	9	0.777777778	0.777777778	0.777777778
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	15	11	15	5	16	0.6875	0.733333333	0.709677419
	16	11	14	5	16	0.6875	0.6875	0.6875
	11	7	19	3	10	0.7	0.636363636	0.666666667
	13	9	17	2	11	0.818181818	0.692307692	0.75
	8	6	22	2	8	0.75	0.75	0.75
	11	8	19	2	10	0.8	0.727272727	0.761904762
	14	9	16	5	14	0.642857143	0.642857143	0.642857143
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	20	15	10	5	20	0.75	0.75	0.75
	8	6	22	2	8	0.75	0.75	0.75
	13	10	17	3	13	0.769230769	0.769230769	0.769230769
2	18	13	12	5	18	0.722222222	0.722222222	0.722222222
	15	10	15	4	14	0.714285714	0.666666667	0.689655172
	10	8	20	2	10	0.8	0.8	0.8
	15	11	15	5	16	0.6875	0.733333333	0.709677419
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	8	6	22	2	8	0.75	0.75	0.75
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	15	10	15	3	13	0.769230769	0.666666667	0.714285714
	14	9	16	5	14	0.642857143	0.642857143	0.642857143
	11	7	19	2	9	0.777777778	0.636363636	0.7
	9	7	21	2	9	0.777777778	0.777777778	0.777777778
	11	7	3	3	10	0.7	0.636363636	0.666666667
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	18	13	12	5	18	0.722222222	0.722222222	0.722222222
	8	6	22	2	8	0.75	0.75	0.75
	15	11	15	4	15	0.733333333	0.733333333	0.733333333
	15	10	15	3	13	0.769230769	0.666666667	0.714285714
	13	9	17	4	13	0.692307692	0.692307692	0.692307692
	15	10	15	3	13	0.769230769	0.666666667	0.714285714
	8	6	22	2	8	0.75	0.75	0.75
	20	15	10	5	20	0.75	0.75	0.75
	19	14	11	5	19	0.736842105	0.736842105	0.736842105
	15	10	15	3	13	0.769230769	0.666666667	0.714285714
	16	12	14	3	15	0.8	0.75	0.774193548
	Promedios					0.740115728	0.721107955	0.729701516

Tabla 4.3. Resultados del cálculo de las medidas de *recall*, *precision* y *f1-measure* correspondientes al estudio de usuario para el caso del método de recomendación propuesto.

Id. Contexto	Área Geo.	Relevantes		No Relevantes		Recomen.	*Precision*	*Recall*	*F-measure*
		Tot.	Correct.	Tot.	Err.				
1	1	15	12	16	2	14	0.857142857	0.8	0.827586207
		20	16	13	4	20	0.8	0.8	0.8
		18	14	10	4	18	0.777777778	0.777777778	0.777777778
	2	10	8	12	2	10	0.8	0.8	0.8
		14	11	17	2	13	0.846153846	0.785714286	0.814814815
		14	10	14	3	13	0.769230769	0.714285714	0.740740741
2	1	20	16	19	4	20	0.8	0.8	0.8
		17	13	20	2	15	0.866666667	0.764705882	0.8125
		8	7	19	1	8	0.875	0.875	0.875
	2	11	9	11	2	11	0.818181818	0.818181818	0.818181818
		18	14	21	4	18	0.777777778	0.777777778	0.777777778
		16	13	21	4	17	0.764705882	0.8125	0.787878788
3	1	12	9	11	3	12	0.75	0.75	0.75
		13	10	15	3	13	0.769230769	0.769230769	0.769230769
		14	11	14	3	14	0.785714286	0.785714286	0.785714286
	2	9	7	19	2	9	0.777777778	0.777777778	0.777777778
		19	15	17	4	19	0.789473684	0.789473684	0.789473684
		13	10	22	2	12	0.833333333	0.769230769	0.8
4	1	16	12	19	2	14	0.857142857	0.75	0.8
		9	7	16	1	8	0.875	0.777777778	0.823529412
		20	16	11	4	20	0.8	0.8	0.8
	2	8	7	10	1	8	0.875	0.875	0.875
		18	14	22	4	18	0.777777778	0.777777778	0.777777778
		9	7	17	1	8	0.875	0.777777778	0.823529412
5	1	11	8	12	2	10	0.8	0.727272727	0.761904762
		15	12	15	3	15	0.8	0.8	0.8
		15	11	20	3	14	0.785714286	0.733333333	0.75862069
	2	19	15	15	4	19	0.789473684	0.789473684	0.789473684
		17	13	11	2	15	0.866666667	0.764705882	0.8125
		11	9	22	3	12	0.75	0.818181818	0.782608696
6	1	15	11	11	3	14	0.785714286	0.733333333	0.75862069
		9	7	15	1	8	0.875	0.777777778	0.823529412
		16	13	16	2	15	0.866666667	0.8125	0.838709677
	2	10	8	19	2	10	0.8	0.8	0.8
		18	14	21	4	18	0.777777778	0.777777778	0.777777778
		16	13	19	4	17	0.764705882	0.8125	0.787878788
7	1	9	7	11	1	8	0.875	0.777777778	0.823529412
		10	8	12	2	10	0.8	0.8	0.8
		17	13	22	2	15	0.866666667	0.764705882	0.8125
	2	20	16	15	4	20	0.8	0.8	0.8
		20	16	15	4	20	0.8	0.8	0.8
		20	16	17	4	20	0.8	0.8	0.8
8	1	14	11	15	3	14	0.785714286	0.785714286	0.785714286
		17	13	22	4	17	0.764705882	0.764705882	0.764705882
		18	14	10	4	18	0.777777778	0.777777778	0.777777778
	2	15	11	11	4	15	0.733333333	0.733333333	0.733333333
		8	7	15	1	8	0.875	0.875	0.875
		9	7	14	1	8	0.875	0.777777778	0.823529412
		Promedios					0.811104272	0.786028186	0.798369373

Tabla 4.4. Resultados del cálculo de las medidas de *recall*, *precision* y *f1-measure* correspondientes al experimento offline para el caso del método de recomendación propuesto.

De acuerdo con los resultados del estudio de usuario (Tabla 4.3), durante la fase de entrenamiento de la puesta en marcha, esto es, en un escenario de escasez de *ratings*, el método de recomendación propuesto es capaz de recomendar correctamente 11 de cada 13 establecimientos de alimentos y bebidas juzgados como relevantes por un usuario en el mejor de los casos. Esto es equivalente a un valor de *recall* de 0,85 (85%). Por el contrario, en las circunstancias antes mencionadas el método de recomendación propuesto es capaz de recomendar correctamente solo 7 de cada 11 establecimientos en el peor de los casos; esto corresponde a un valor de *recall* de 0,64 (64%). El valor promedio de recall en este caso es 0,72 (72%).

De manera similar, en un escenario de escasez de *ratings*, 9 de cada 11 establecimientos de alimentos y bebidas recomendados por el método de recomendación propuesto son establecimientos relevantes, es decir, establecimientos juzgados como relevantes por un usuario. Este representa el mejor de los casos, y equivale a un valor de *precision* de 0,82 (82%). Por el contrario, en las circunstancias antes mencionadas, solo 9 de cada 14 establecimientos recomendados por el método de recomendación propuesto son establecimientos juzgados como relevantes por un usuario; este representa el peor de los casos, y corresponde a un valor de *precision* de 0,64 (64%). El valor promedio de *precision* en este caso es 0,74 (74%).

Las figuras 4.6 y 4.7 respectivamente representan los resultados del cálculo de las medidas de *recall* y *precision* correspondientes al estudio de usuario, tanto para el caso del método de recomendación propuesto, como para el caso método de recomendación de línea base. A efectos prácticos, los valores de las variables de las Fórmulas 4.5, 4.7 y 4.8 (f_1-*mesure*, *recall* contextualizada y *precision* contextualizada) obtenidos por cada participante y usuario simulado para el caso del método de línea base no se muestran en este documento.

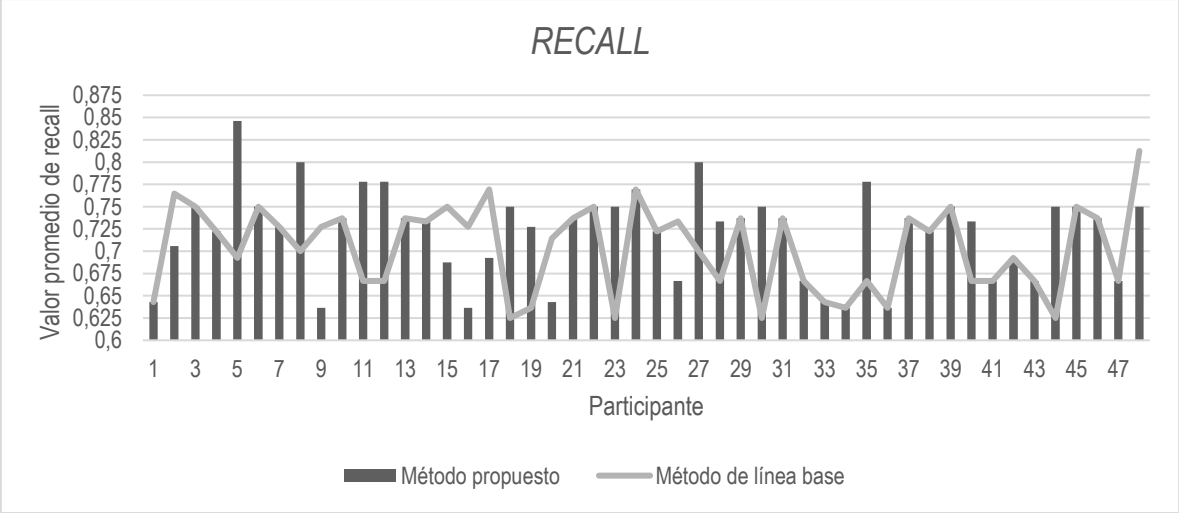


Figura 4.6. Comparación de las medidas de recall calculadas para el método de recomendación propuesto y para el método de recomendación de línea base (estudio de usuario).

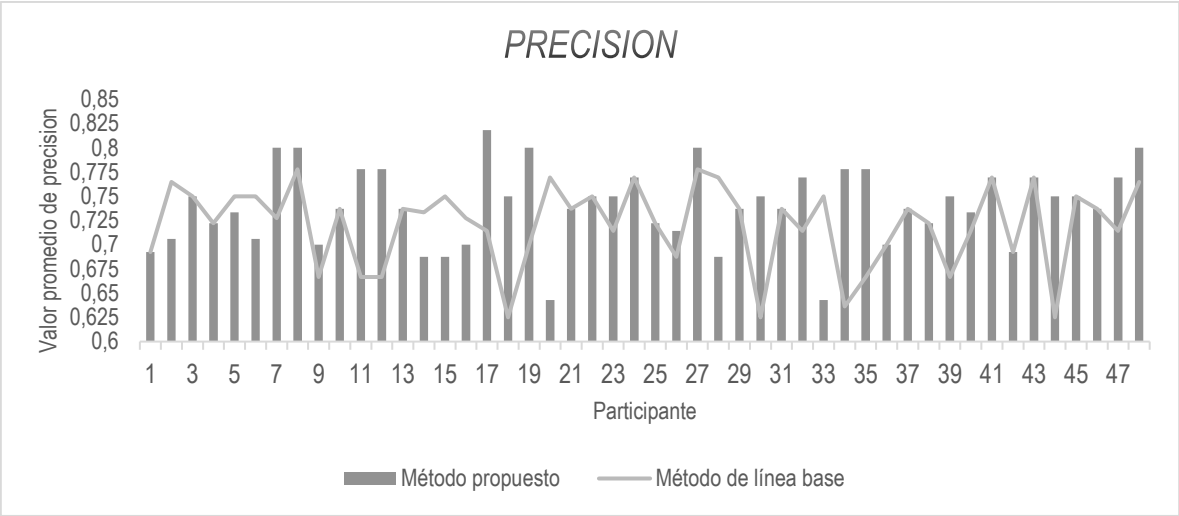


Figura 4.7. Comparación de las medidas de precision calculadas para el método de recomendación propuesto y para el método de recomendación de línea base (estudio de usuario).

A partir de las figuras 4.6 y 4.7 se puede concluir que en un escenario de escasez de *ratings* el método de recomendación propuesto es capaz de superar al método de recomendación del estado del arte seleccionado en un 22% tanto en términos de *recall* como en términos de *precision* en el mejor de los casos (participante 5 y participante 34, respectivamente). No obstante, en las circunstancias antes mencionadas, el método de recomendación propuesto es superado por el método del estado del arte seleccionado en un 14% en términos de *recall* y en un 20% en términos de *precision* en el peor de los casos (participante 9/participante 16 y participante 20, respectivamente). De hecho, en un escenario de escasez de *ratings*, el método propuesto es capaz de superar al método del estado del arte en solo un 2,46% en términos de valores promedio de *recall* y un 2,73% en términos de valores promedio de *precision*. Evidentemente, esta mejora, aunque modesta, se atribuye al uso de la métrica de similitud semántica basada en ontologías, la cual representa una parte integral del método propuesto. En este sentido, es esperable un comportamiento similar frente a otras métricas de similitud "sintáctica" populares entre las técnicas de filtrado colaborativo basado en memoria, como la métrica *coeficiente de correlación de Pearson*.

Por otra parte, de acuerdo con los resultados del experimento *offline* (Tabla 4.4), durante la fase de "pruebas" de la puesta en marcha del método de recomendación propuesto, esto es, en condiciones normales de disponibilidad de *ratings*, el método de recomendación propuesto es capaz de recomendar correctamente 7 de cada 8 establecimientos de alimentos y bebidas juzgados como relevantes por un usuario en el mejor de los casos. Esto es equivalente a un valor de *recall* de 0,88 (88%). Por el contrario, en las circunstancias antes mencionadas el método de recomendación propuesto es capaz de recomendar correctamente solo 10 de cada 14 establecimientos en el peor de los casos; esto corresponde a un valor de *recall* de 0,71 (71%). El valor promedio de *recall* en este caso es 0,79 (79%).

De manera similar, en condiciones normales de disponibilidad de *ratings*, 7 de cada 8 establecimientos de alimentos y bebidas recomendados por el método de recomendación propuesto son establecimientos relevantes, es decir, establecimientos juzgados como relevantes por un usuario. Este representa el mejor de los casos, y equivale a un valor de *precision* de 0,88 (88%). Por el contrario, en las circunstancias antes mencionadas, solo 11 de cada 15 establecimientos recomendados por el método de recomendación propuestos son establecimientos juzgados como relevantes por un usuario; este representa el peor de los casos, y corresponde a un valor de *precision* de 0,73 (73%). El valor promedio de *precision* en este caso es 0,81 (81%).

Las figuras 4.8 y 4.9 respectivamente representan los resultados del cálculo de las medidas de *recall* y *precision* correspondientes al experimento *offline*, tanto para el caso del método de propuesto, como para el caso método de línea base. A efectos prácticos, en estas figuras se representan los valores promedio por contexto socio-temporal simulado.

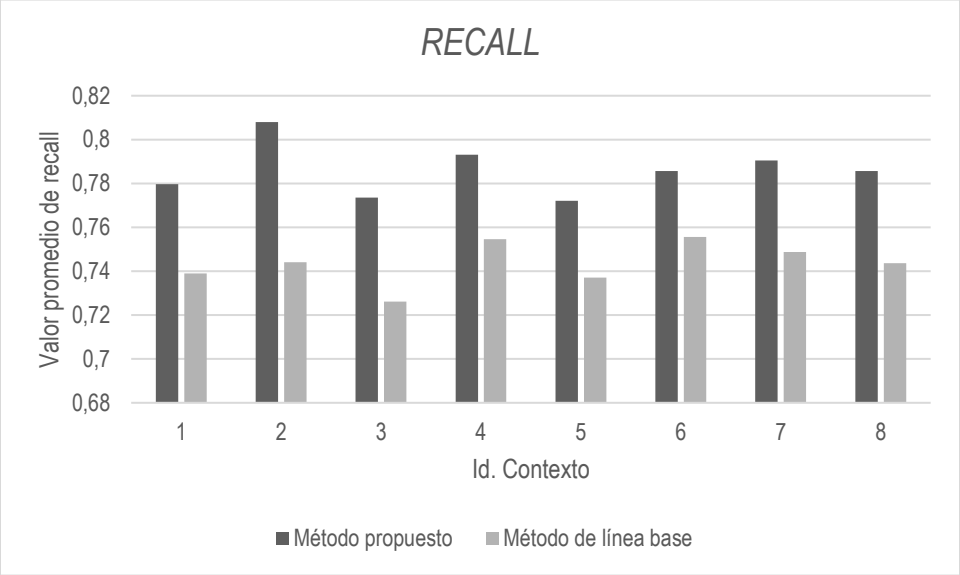


Figura 4.8. Comparación de las medidas de recall calculadas para el método de recomendación propuesto y para el método de recomendación de línea base (experimento *offline*).

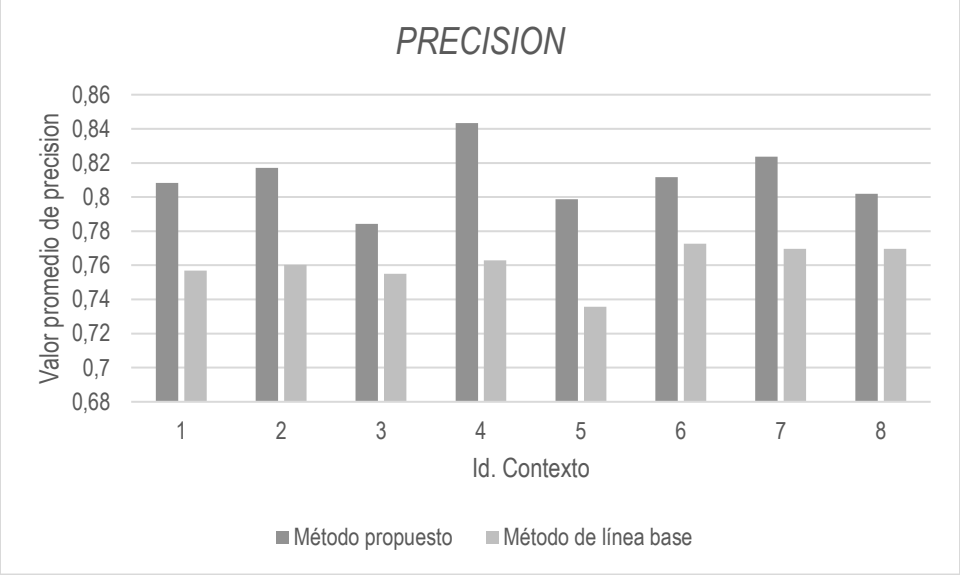


Figura 4.9. Comparación de las medidas de precision calculadas para el método de recomendación propuesto y para el método de recomendación de línea base (experimento *offline*).

Como se puede concluir a partir de las figuras 4.8 y 4.9, los valores de *recall* y *precision*, y especialmente los valores de *precision*, varían ligeramente entre contextos socio-temporales, tanto en el caso del método propuesto, como en el caso del método de línea base en condiciones normales de disponibilidad de *ratings*. Esto podría considerarse un claro indicativo de que los tipos de información contextual considerados por la técnica de modelado de información contextual presentada en este documento como parte integral del método

de recomendación propuesto son realmente relevantes para la provisión, lo más realista posible, de recomendaciones de establecimientos de alimentos y bebidas.

En detalle, el segundo valor promedio más alto de *precision* es alcanzado en el intervalo correspondiente al contexto 7 (“Thursday”, “Night”, “Friends”) en ambos casos (el caso del método propuesto y el caso del método de línea base). En el caso del método propuesto el valor promedio más alto de *precision* es alcanzado en el intervalo correspondiente al contexto 4 (“Tuesday”, “Night”, “Family”); este incremento en las medidas de *precision* corresponde con el valor promedio más bajo de número de establecimientos recomendados. El hecho de que varíe tanto la información contextual temporal (al menos el día de la semana) como la información contextual social entre los dos intervalos con los dos mayores valores de *precision* puede significar que ambos tipos de información contextual son igualmente importantes para la predicción de los establecimientos que mejor se ajustan a las preferencias de los usuarios en un contexto socio-temporal específico.

Asimismo, el valor promedio más bajo de *precision* es alcanzado en el intervalo correspondiente al contexto 3 (“Tuesday”, “Night”, “Friends”) en prácticamente ambos casos (exceptuando el intervalo correspondiente al contexto 5 en el caso del método de línea base). El valor promedio más bajo de *precision* en el caso del método de línea base (el intervalo correspondiente al contexto 5) corresponde con el tercer valor promedio más alto de número de establecimientos recomendados.

En lo que respecta al cálculo de las medidas de *recall* cabe mencionar los siguientes hallazgos. El valor promedio alcanzado en el intervalo correspondiente al contexto 7 representa un decremento mínimo en las medidas de *recall* en ambos casos (el caso del método propuesto y el caso del método de línea base); de hecho, se trata del quinto valor promedio más bajo en ambos casos. Asimismo, el valor promedio alcanzado en el intervalo correspondiente al contexto 3 es, de hecho, el valor promedio más bajo en prácticamente ambos casos (exceptuando el intervalo correspondiente al contexto 5 en el caso del método propuesto). Una posible explicación para este comportamiento es el tamaño de la matriz de *ratings*, concretamente el número de *ítems* en la misma; de hecho, tanto para el estudio de usuario, como para el experimento *offline*, el espacio de recomendación se redujo a solo 30 establecimientos en dos áreas geográficas predefinidas.

A manera de resumen, es importante mencionar que, en condiciones normales de disponibilidad de *ratings*, el método propuesto es capaz de superar al método de línea base en un 8,60% en términos de valores promedio de *recall* y en un 10,53% en términos de valores promedio de *precision* en el mejor de los casos (intervalos correspondientes a los contextos 2 y 4, respectivamente). Asimismo, el método propuesto es capaz de superar al método de línea base en un 3,98% en términos de valores promedio de *recall* y en un 3,87% en términos de valores promedio de *precision* en el peor de los casos (intervalos correspondientes a los contextos 6 y 3 respectivamente). La mejora considerable en el rendimiento respecto al estudio de usuario se puede atribuir al uso conjunto de la métrica de similitud semántica basada en ontologías y el modelo *LDA* de información contextual de alto nivel, el cual representa otra parte integral del método propuesto (la técnica de modelado de información contextual, formalmente). Como se vio a lo largo del capítulo anterior, esto tiene un impacto directo no solo en modelado propiamente dicho de la información contextual sino en el perfilamiento de los usuarios.

Finalmente, a fin de comparar las listas de top-n recomendaciones producidas por ambos métodos de recomendación, se calcularon las medidas de *NDPM* para todos los usuarios simulados en el experimento *offline* respecto a las listas de top-5 establecimientos de alimentos y bebidas relevantes compiladas por los expertos durante dicho experimento (*rankings* de referencia). Para ello, se consideraron, evidentemente, solo los cinco establecimientos mejor posicionados en las listas de recomendaciones producidas por ambos métodos de recomendación -las top-5 recomendaciones. En este contexto cabe mencionar que, a efectos prácticos, los valores de las variables de la fórmula de la métrica antes mencionada (Fórmula 4.5) obtenidos por cada usuario simulado no se muestran en este documento.

La Figura 4.10 representa los resultados del cálculo de las medidas de *NDPM*, tanto para el caso del método propuesto, como para el caso método de línea base. Continuando con la temática manejada a lo largo de esta discusión de resultados, en esta figura se representan los valores promedio por contexto socio-temporal simulado. No obstante, en el caso particular de la métrica *NDPM*, entre más se acerca a la unidad el valor promedio de las medidas, peor es el rendimiento del sistema de recuperación de información (el sistema de recomendación, en este caso) en la reducción de la distancia entre los *rankings* de referencia y los *rankings* producidos por este; de hecho, en este caso la puntuación perfecta es cero.

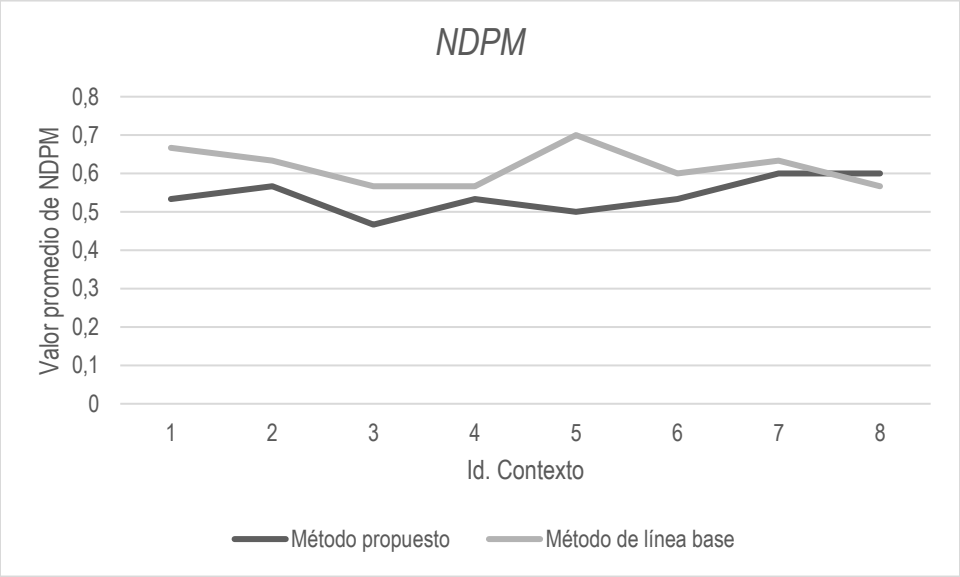


Figura 4.10. Comparación de las medidas de *NDPM* calculadas para el método de recomendación propuesto y para el método de recomendación de línea base.

Como se puede observar en la Figura 4.10, las medidas de *NDPM* varían ligeramente entre contextos socio-temporales, tanto en el caso del método propuesto, como en el caso del método de línea base, tal y como sucede especialmente con las medidas de *precision*. Obsérvese que, además, las medidas de *NDPM* son inversamente proporcionales a las medidas de *precision*. Una posible explicación para esto es que, entre más establecimientos juzgados como relevantes por un usuario son recomendados por el método de recomendación propuesto (o por el método de línea base), menos probabilidades hay de que existan relaciones de preferencia contradictorias (C^-) entre el *ranking* producido por este y el *ranking* de referencia, y viceversa. En este contexto cabe resaltar que, en este experimento, ni los *rankings* producidos por los métodos de recomendación, ni los *rankings* de referencia contenían rangos repetidos, por lo que el número de relaciones de compatibilidad (C^u) entre los *rankings* fue fijado en cero para todos los usuarios simulados; como consecuencia, dado que el tamaño de los *rankings* fue reducido a cinco, el número de relaciones de prioridad (C^i) entre los *rankings* fue fijado en cinco para todos los usuarios simulados.

En detalle, el valor promedio más bajo de *NDPM* es alcanzado en el intervalo correspondiente al contexto 3 en ambos casos (además de en los intervalos correspondiente a los contextos 4 y 8 en el caso del método de línea base). Nótese que, como en el caso de los dos intervalos en los que se alcanzan los dos valores más altos de *precision* (análisis comparativo respecto a las métricas *recall* y *precision* correspondiente al experimento *offline*), en este caso, entre los intervalos correspondientes a los contextos 3 y 8 varía tanto la información contextual temporal (al menos el día de la semana) como la información contextual social.

Por otro lado, el segundo valor promedio más alto de *NDPM* es alcanzado en el intervalo correspondiente al contexto 2 en ambos casos (exceptuando el intervalo correspondiente al contexto 1 en el caso del método de línea base). De hecho, en el caso específico del método propuesto, el valor promedio más alto de *NDPM* es alcanzado en el intervalo correspondiente al contexto 7. Nótese que, una vez más, entre los contextos 2 y 7 varía tanto la información temporal como la información social; estos hallazgos podrían significar que ambos tipos de información contextual son igualmente importantes para la generación, lo más realista posible, de recomendaciones de establecimientos de alimentos y bebidas que mejor se ajusten a las preferencias de los usuarios en un contexto socio-temporal dado.

A manera de resumen, vale la pena mencionar que, en el mejor de los casos (el intervalo correspondiente al contexto 5), el método propuesto es capaz de superar al método de línea base en un 28,57% en términos de valores promedio de *NDPM* (0,5 vs. 0,7). Sin embargo, en el peor de los casos (el intervalo correspondiente al contexto 8), el método propuesto es superado por el método de línea base en un 5,56% en los mismos términos (0,57 vs. 0,6). El valor promedio acumulado de *NDPM* es 0,54 (54%) en el caso del método propuesto y 0,62 (62%) en el caso del método de línea base. Esto puede ser un claro indicativo de que la técnica de recomendación basada en modelos de tópicos, la cual es una parte integral del método propuesto, permite generar recomendaciones de establecimientos de una manera más realista al considerar las preferencias de los usuarios por los tópicos en el modelo *LDA* subyacente, y no solo las preferencias por los establecimientos propiamente dichos.

4.6. Conclusión

Existen una gran variedad de técnicas o métricas de evaluación de sistemas de recomendación bajo enfoques de evaluación en las Ciencias de la Computación y en las Ciencias de la Información, por lo que la elección de las más adecuadas depende de los objetivos de la evaluación a llevar a cabo. Asimismo, cada métrica puede aportar una perspectiva distinta de las características o de los sistemas de recomendación, incluyendo técnicas o algoritmos y herramientas de software, como un todo. Por lo tanto, siempre se debe considerar la posibilidad de emplear más de una métrica de naturaleza diferente en una misma evaluación a fin de abordar el proceso de evaluación desde distintas perspectivas.

Otro aspecto crítico de la evaluación de los sistemas de recomendación es la interpretación de los resultados derivados de la utilización de las métricas seleccionadas, ya que la interpretación errónea o poco adecuada de los mismos puede dar lugar a conclusiones poco realistas acerca del rendimiento de los sistemas y ello derivar, a su vez, en la toma de decisiones poco acertadas, sobre todo en escenarios de puesta en marcha de sistemas comerciales de cara a usuarios reales. En este sentido, y tal vez aún más en el ámbito académico, resulta particularmente crucial la presentación de los resultados de las evaluaciones; la elección de las técnicas de presentación más adecuadas es fundamental para la comunicación y difusión efectiva de los mismos de cara al aprovechamiento y extensión por parte de otros desarrolladores y académicos del área de las técnicas y herramientas de software evaluadas. Para ello es igualmente importante la formalización, en la medida de lo posible, no solo de las propuestas propiamente dichas, sino también de las evaluaciones, a fin de proveer métodos de evaluación replicables.

Gracias a los resultados del método de evaluación propuesto en esta tesis doctoral, se ha podido comprobar la eficacia y efectividad superior del método de recomendación sensible al contexto de establecimientos de alimentos y bebidas que representa la contribución de esta investigación, específicamente de la técnica de recomendación y de la técnica de representación y modelado de información contextual basadas en tecnologías de la Web Semántica y modelos estadísticos de clases latentes, respecto a un método de evaluación del estado del arte. Esto ha llevado, finalmente, a la validación de la propuesta de esta tesis doctoral y esto a su vez a la demostración de la tesis de la investigación.

Capítulo 5 . Conclusiones, Contribuciones y Trabajo Futuro

5.1. Conclusiones y Contribuciones

El uso de las tecnologías de la Web Semántica en el desarrollo de sistemas de recomendación se ha convertido a finales de la década pasada en tendencia dentro del campo de la investigación en sistemas de recomendación, y esto ha sido motivado en gran medida por la necesidad histórica de dar solución al problema de la sobrecarga de información. Como resultado, los denominados sistemas híbridos de recomendación basada en conocimiento han adquirido cierta popularidad en dicho campo. En detalle, estos sistemas combinan técnicas de recomendación basadas en tecnologías de la Web Semántica con técnicas de recomendación tradicionales, especialmente técnicas de recomendación basada en filtrado colaborativo, de modo que se compensan las desventajas de las últimas al mismo tiempo que se aprovechan las ventajas de ambas.

Al mismo tiempo, la idea de un tipo especial de sistema de recomendación destinado a la generación lo más realista posible de recomendaciones útiles dentro de circunstancias determinadas, es decir, contextos determinados, se ha consolidado como línea de investigación en el campo de la investigación en sistemas de recomendación a finales de la década pasada. Este tipo de sistema de recomendación surgió de la hipótesis de que el proceso de recomendación involucra información potencialmente útil para la predicción de *ratings*, la cual debe ser preservada a fin de generar recomendaciones más relevantes desde el punto de vista del usuario.

A partir del análisis del estado del arte realizado en esta tesis doctoral se pudo determinar que los dominios del turismo y del ocio representan dos áreas de aplicación muy populares de los sistemas de recomendación sensibles al contexto. Esta investigación pretendió contribuir con soporte tecnológico a un área que depende parcialmente de la afluencia de turistas, a saber, la industria de los servicios de alimentos y bebidas, mediante la aplicación de un sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas. Por otro lado, las contribuciones particulares de esta investigación están destinadas a servir como punto de partida para otros académicos y desarrolladores del área de sistemas de recomendación en el afán de construir un ecosistema abierto de datos y aplicaciones semánticas homogenizadas e integradas bajo un enfoque de *Linked Data*.

A continuación, se discuten brevemente las contribuciones puntuales de esta tesis doctoral, el alcance o las limitaciones principales de estas, así como las posibles líneas de investigación futura a las que dichas limitaciones pueden dar pie considerando el objetivo a largo plazo de la investigación.

En esta investigación se han aprovechado las tecnologías de la Web Semántica, principalmente los lenguajes de descripción y definición de vocabularios, *OWL* y *RDFS*, el *framework* de propósito general para la representación de datos en la Web, *RDF*, el lenguaje de consulta para *RDF*, *SPARQL*, y la notación para la definición basada en *SPARQL* de reglas de inferencia y restricciones, *SPIN*, así como el modelo probabilístico generativo de tópicos, *LDA*, con dos objetivos principales: la representación de conocimiento y el razonamiento en el dominio de la restauración.

En lo que respecta a la representación de conocimiento, se ha propuesto un modelo ontológico basado en *OWL* del dominio antes mencionado, el cual pretende además enlazar y, finalmente integrar, los vocabularios de las *APIs* de redes sociales basados en localización, como sitios Web de opiniones de usuarios, que proveen contenido heterogéneo en el dominio de la restauración, utilizando un enfoque de *Linked Data*.

Dicho modelo es empleado asimismo con fines de modelado de las preferencias y el contexto de los usuarios. En este sentido, realmente se ha propuesto una técnica híbrida de modelado de información contextual basada en modelos probabilísticos generativos de tópicos, específicamente modelos *LDA*. Esta técnica aprovecha el modelo ontológico del dominio, así como una base de reglas de inferencia representadas bajo la notación *SPIN*, para la inferencia de información contextual temporal y social de alto nivel a partir de información relacionada

de bajo nivel. De hecho, con el modelo del contexto propuesto en esta investigación se pretende integrar dicha información contextual de alto nivel con información contextual de ubicación de bajo nivel, esto es, datos de geolocalización obtenidos desde dispositivos móviles, con el objetivo primordial de explorar la importancia de las situaciones sociales de los usuarios al recomendar a estos establecimientos de alimentos y bebidas que mejor cubren sus necesidades en contextos determinados.

Como se puede deducir, esta técnica representa una contribución de esta tesis doctoral en el ámbito del razonamiento. Junto con esta, se ha propuesto en el mismo ámbito una métrica de similitud semántica basada en ontologías, la cual emplea un enfoque híbrido taxonómico/no taxonómico de conjuntos de características ontológicas, y permite capturar, además de conocimiento taxonómico, conocimiento no taxonómico tanto explícito como inferido. En este contexto, es importante aclarar que, si bien esta métrica formalmente no emplea un enfoque híbrido de caminos en grafos y de conjuntos de características ontológicas, si permite capturar los tipos de conocimiento semántico a los que comúnmente están destinados dichos enfoques: conocimiento taxonómico y conocimiento no taxonómico, respectivamente. En conjunto con una técnica de propagación de preferencias a través de jerarquías de clases en perfiles de preferencias basados en ontologías, le métrica de similitud semántica propuesta pretende aliviar el problema por el cual los sistemas puros de recomendación basada en filtrado colaborativo son susceptibles de verse afectados: el arranque en frío en sus modalidades de nuevo usuario e ítem nuevo.

En el ámbito de la recomendación propiamente dicha, cabe mencionar que el modelo *LDA* subyacente a la técnica de modelado contextual es además explotado para los fines del perfilamiento de usuarios (modelado de preferencias) de modo que los establecimientos de alimentos y bebidas son recomendados a los usuarios a partir de, por un lado, las similitudes semánticas entre los establecimientos y, por otro lado, dos tipos distintos de preferencias: preferencias por establecimientos y preferencias por tópicos en el modelo *LDA*. En este sentido, a manera de *framework* se ha propuesto una técnica híbrida de filtrado colaborativo basado en modelos y de filtrado colaborativo basado en ítems bajo el enfoque de top-n recomendaciones, en donde el componente basado en modelos lo representa el modelo *LDA* de información contextual de alto nivel (modelo probabilístico generativo de tópicos), y el componente basado en ítems lo representa la métrica de similitud semántica basada en ontologías.

El método de recomendación sensible al contexto de establecimientos de alimentos y bebidas resultante se puede considerar un método híbrido basado en conocimiento de filtrado colaborativo probabilístico y basado en memoria. Finalmente, se ha propuesto en esta tesis doctoral una arquitectura de software destinada a la implementación del método de recomendación resultante; dicha arquitectura se basa en un diseño lógico cliente/servidor de tres capas, en el que la capa de presentación cosiste en una aplicación móvil nativa basada en tecnologías Web -la aplicación cliente. Asimismo, como prueba de concepto se ha implementado un prototipo de sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas a partir de dicha arquitectura.

5.1.1. Demostración de la Tesis de la Investigación

Por un lado, mediante el análisis del estado del arte se ha podido dar respuesta a las preguntas definidas a fin de posibilitar la demostración de la tesis de esta investigación a partir de la **Sub-hipótesis 1**.

En detalle, según los resultados del estudio del estado del arte discutidos en la sección 2.6 del Capítulo 2 de este documento, prácticamente no existen propuestas en la literatura de sistemas de recomendación sensible al contexto que apliquen, tanto tecnologías de la Web Semántica, como modelos estadísticos de clases latentes, al dominio particular de la restauración con dos fines distintos: (1) la definición de técnicas híbridas de minería, modelado y representación de información contextual y (2) la definición de técnicas híbridas de recomendación.

De ninguna manera, esto puede considerarse indicio de que no es posible combinar estos tipos de tecnologías y técnicas computacionales para dichos fines. De hecho, según los resultados de la ejecución del método de evaluación propuesto en esta investigación, los cuales se discuten en profundidad en la sección 4.5 del Capítulo 4 de este documento, en un contexto de clasificación de ítems, la exactitud de las recomendaciones generadas por el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas que representa la contribución de esta investigación se ve mejorada considerablemente con la integración del modelo *LDA* de información contextual de alto nivel (formalmente, la técnica de modelado de información contextual) a la métrica de similitud semántica basada en ontologías. Concretamente, esto se puede observar al comparar los resultados del experimento *offline* con los resultados del estudio de usuario.

Además, como se mencionó en la sección 2.6 del Capítulo 2 de este documento, las propuestas existentes en la aplicación de técnicas y herramientas de recomendación al dominio de la restauración emplean, bien tecnologías de la Web semántica, comúnmente el lenguaje de definición y descripción de vocabularios, *OWL*, y el lenguaje de reglas, *SWRL*, bien modelos estadísticos de clases latentes, comúnmente el modelo probabilístico generativo de tópicos, *LDA*, para los fines antes mencionados.

Por otro lado, en lo que respecta a la demostración de la tesis de esta investigación a partir de las preguntas asociadas a la **Sub-hipótesis 2**, el estudio del estado del arte ha permitido responder, al menos parcialmente, a dichas preguntas, tal como en el caso anterior.

En este caso, según los resultados de dicho estudio, las propuestas en el campo de la investigación en sistemas de recomendación sensible al contexto que aplican, bien tecnologías de la Web Semántica, bien modelos estadísticos de clases latentes, a fin de definir técnicas de minería, modelado y representación de información contextual más potentes, comúnmente no contemplan información contextual de alto nivel como parte de los modelos del contexto; sin embargo, consideran distintos tipos de información de bajo nivel, principalmente información de ubicación e información temporal. Asimismo, dichas contribuciones no contemplan información contextual de tipo social a ningún nivel de abstracción.

En este sentido, según los hallazgos derivados de la ejecución del método de evaluación propuesto en esta investigación, la información contextual temporal de alto nivel, a la par de la información contextual social de alto nivel, parece ser relevante en la generación, tanto más exacta en el contexto tradicional de la recuperación de información, como más realista desde el punto de vista del usuario (exactas en un contexto de ordenamiento de ítems), de recomendaciones de establecimientos de alimentos y bebidas que mejor se ajusten a las preferencias del usuario en un contexto socio-temporal dado.

Finalmente, mediante el método de evaluación destinado a la validación de la propuesta de esta investigación, ha sido posible dar respuesta a las preguntas definidas a fin de poder demostrar la veracidad de la tesis de investigación a partir de la **Sub-hipótesis 3**.

En detalle, según los resultados de la ejecución del método de evaluación propuesto en esta investigación, en un contexto de clasificación de ítems, la exactitud de las recomendaciones generadas por el método de recomendación sensible al contexto de establecimientos de alimentos y bebidas que representa la contribución de esta investigación se ve mejorada en un 6,67% en términos de valores promedio de *recall* y en un 5,70% en términos de valores promedio de *precision* respecto a las recomendaciones generada por un método de recomendación de línea base (métrica de similitud sintáctica). Concretamente, esto se puede deducir de los resultados del experimento *offline* llevado a cabo en esta investigación, el cual, en contraposición al estudio de usuario, representa un escenario de suficiencia de *ratings*.

Asimismo, en un contexto de ordenamiento de ítems, la exactitud de las recomendaciones generadas por el método de recomendación propuesta se ve mejorada en un 12,16% en términos de valores acumulados de

NDPM respecto a las recomendaciones generadas por un método de recomendación de línea base (métrica de similitud sintáctica). Esto se puede interpretar como que las recomendaciones generadas a partir de técnicas de recomendación basadas en tecnologías de la Web Semántica y modelos estadísticos de clases latentes son 12% más realistas desde el punto de vista del usuario que las recomendaciones generadas a partir de técnicas de recomendación del estado del arte.

5.2. Trabajo Futuro

Como se explicó a lo largo de los dos capítulos anteriores al presente capítulo, durante la puesta en marcha del método propuesto, o más bien del prototipo del sistema de recomendación sensible al contexto de establecimientos de alimentos y bebidas implementado a partir de la arquitectura de software propuesta, se requiere de una fase de recolección de historiales de *check-ins* de usuarios. Dicha fase denominada "de entrenamiento" permite la construcción del modelo *LDA* de información contextual de alto nivel; por lo tanto, mientras no se recolecte información histórica suficiente como para construir un modelo *LDA* robusto, todas las características del método que dependen de la existencia de este no estarán disponibles; principalmente se trata de la parte del método correspondiente al enfoque de recomendación sensible al contexto.

Esto se debe, evidentemente, a que las *APIs* de dos de los servicios basados en localización que se intentan integrar y enlazar con la ontología del dominio y el correspondiente repositorio de conocimiento (datos *RDF*) empleando un enfoque de *Linked Data* no ponen a disposición a través de sus *APIs* datos sobre los *check-ins* asociados a los establecimientos existentes en sus bases de datos. En este sentido, es imprescindible la definición de algún mecanismo basado en una fuente de datos externa alternativa (por ejemplo, Facebook) que permita no depender de los datos de uso del sistema a fin de posibilitar la construcción del modelo *LDA* de información contextual de alto nivel. Para más detalles acerca de esta aproximación a esta posible solución, favor de referirse a los productos académicos de esta investigación, específicamente al trabajo (Colombo-Mendoza, Valencia-García, Rodríguez-González, Colomo-Palacios, & Alor-Hernández, 2017).

Por otro lado, cabe mencionar que la técnica de propagación de preferencias a través de jerarquías de clases en perfiles de preferencias basados en ontologías, si bien, como se discute detalladamente en la sección anterior, ha demostrado jugar un papel fundamental en la mitigación del problema del arranque en frio, específicamente de la modalidad del nuevo usuario, está limitada al conocimiento taxonómico, es decir, no considera relaciones no taxonómicas, ni explícitas ni inferidas. Por lo tanto, la aplicación de esta se ve limitada a casos particulares, como el caso planteado en esta tesis doctoral. No obstante, resultaría sumamente relevante su extensión en el sentido mencionado, a fin de ampliar su aplicabilidad de cara al mantenimiento futuro de la ontología del dominio, o de su adaptación y uso en otros subdominios del dominio del turismo.

Asimismo, la métrica de similitud semántica no taxonómica basada en ontologías, si bien permite capturar conocimiento no taxonómico inferido (además del conocimiento no taxonómico explícito) empleando un enfoque de características, lo hace solo a partir del concepto de característica común del modelo de similitud de Tversky, es decir, no considera el concepto de característica no común. De hecho, en esta investigación, la métrica de similitud taxonómica está destinada a capturar las características no comunes entre establecimientos, aunque, evidentemente, en un sentido taxonómico. Ciertamente, esto limita el potencial de la métrica de similitud como un todo, por lo que resulta necesario el estudio a futuro de otras interpretaciones del concepto de conocimiento no taxonómico que permitan capturar fielmente tantos tipos de relaciones no taxonómicas en la ontología del dominio como sea posible.

Como parte del trabajo a futuro evidente de esta tesis doctoral, se planea implementar en su totalidad el prototipo del sistema de recomendación implementado como prueba de concepto a partir de la arquitectura de software propuesta, lo que corresponde, principalmente, a la implementación de la aplicación cliente como una aplicación móvil multiplataforma completamente funcional. A mediano plazo, resultaría muy interesante poner

dicha aplicación a disposición de usuarios reales en distintas plataformas de distribución digital de aplicaciones móviles comerciales, como Google Play Store y App Store (Mac App Store).

Esto además posibilitaría la realización de otros tipos de evaluación complementarios a la evaluación propuesta en esta investigación, a saber, evaluaciones dentro del campo de las Ciencias de la Información, específicamente evaluaciones bajo el enfoque de los sistemas de soporte a las decisiones. Esto permitiría la medición de aspectos de calidad del sistema de recomendación relacionados con las capacidades de soporte a las decisiones y, finalmente, la determinación de la calidad del sistema, tal y como es percibida por sus usuarios. En este sentido, es importante tener en cuenta que, como se vio en el capítulo anterior a este capítulo, existe un sin número de métricas de distinta naturaleza orientadas a la medición de aspectos diferentes a la exactitud de los algoritmos de recomendación, las cuales son susceptibles de ser utilizadas en dicha tarea.

En el mismo tópico de la evaluación del sistema, resultaría igualmente relevante la realización de una segunda evaluación dentro del campo de las Ciencias de la Computación (la evaluación propuesta en esta investigación corresponde a esta categoría), en este caso bajo un enfoque de aprendizaje computacional. Comúnmente, este tipo de evaluaciones están enfocadas en la evaluación de aspectos de la calidad técnica de los sistemas, los cuales están más bien relacionados con el rendimiento computacional de los mismos. Teniendo en cuenta que el rendimiento computacional óptimo no fue considerado como prioritario en esta tesis doctoral, dicha evaluación debería estar destinada a la medición de la demanda computacional del algoritmo que implementa la métrica de similitud semántica basada en ontologías y del algoritmo de generación de modelos *LDA* para análisis de estabilidad, los cuales si bien a primera vista no demostraron ser un problema para el rendimiento computacional del sistema de recomendación, se presupone que en determinadas circunstancias podrían serlo.

5.3. Publicaciones en Revistas JCR y Congresos

Publicaciones en revistas JCR:

- Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2015). RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, *42*(3), 1202–1222. https://doi.org/10.1016/j.eswa.2014.09.016
- Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Colomo-Palacios, R., & Alor-Hernández, G. (2017). Towards a knowledge-based probabilistic and context-aware social recommender system. *Journal of Information Science*, 165551517698787. https://doi.org/10.1177/0165551517698787

Publicaciones en congresos:

- Rodríguez-García, M. Á., Colombo-Mendoza, L. O., Valencia-García, R., Lopez-Lorca, A. A., & Beydoun, G. (2015). Ontology-Based Music Recommender System. In S. Omatu, Q. M. Malluhi, S. R. Gonzalez, G. Bocewicz, E. Bucciarelli, G. Giulioni, & F. Iqba (Eds.), Distributed Computing and Artificial Intelligence, 12th International Conference (pp. 39–46). Springer International Publishing. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-19638-1_5
- Colomo-Palacios, R., Colombo-Mendoza, L. O., & Valencia-García, R. (2016). Towards Supporting International Standard-Based Software Engineering Approaches Using Semántica Web Technologies: A Systematic Literature Review. In R. Valencia-García, K. Lagos-Ortiz, G. Alcaraz-Mármol, J. del Cioppo, & N. Vera-Lucio (Eds.), Technologies and Innovation: Second International Conference, CITI 2016, Guayaquil, Ecuador, November 23-25, 2016, Proceedings (pp. 169–183). Cham: Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-319-48024-4_14

Capítulo 6 . Resumen en Inglés

6.1. Introduction

Personalized recommendation systems (or recommender systems, for short) are software techniques and tools whose objective is to make predictions and give users suggestions about the items in a given domain (Ricci et al., 2011). They can be formally considered a mixture of information retrieval systems and Decision Support Systems (DSS) whose objective is to provide personalized information recommendations.

The growing popularity of location-based social networks and user opinion websites across domains is making it possible for companies advertising services through those medium to gain an increasingly broad and accurate understanding of users' preferences; at the same time, this is allowing users to make more and more informed decisions about which services meet their requirements in specific circumstances when considering the suggestions from other users regarding their personal experiences with the services. However, it is essential for companies and users to use techniques and software tools that provide solutions to the problem of information overload in the context of location-based social networks and user opinion websites.

Moreover, a type of recommendation system with the ability to suggest items of possible interest to the user in certain circumstances emerged in the early 2000s: context-aware recommender system (Adomavicius & Tuzhilin, 2008). The objective of this kind of recommender system is to provide the user with more relevant recommendations, and it is based on the assumption that it is crucial to preserve potentially useful information about the context in which the requests for recommendations occur.

Given the suitability of Semantic Web technologies for the development of knowledge-based recommender systems, the use of technologies such as the vocabulary definition and description languages, OWL (Web Ontology Language) and RDFS (Resource Description Framework Schema), the framework for representing information in the Web, RDF (Resource Description Framework), the SPARQL Query Language for RDF and SWRL (Semantic Web Rule Language), in the development of this type of techniques and software tools has recently become a trend in the research field in recommender systems, and this has been motivated by the need to look for new ways of dealing with the problem mentioned above.

As one of the main building blocks of the Semantic Web, ontologies -complex and formal vocabularies- are harnessed with two primordial purposes: knowledge representation and reasoning. Moreover, the creation of recommendation techniques based on Semantic Web technologies, that is, recommendation techniques that take advantage of ontology-based reasoning mechanisms or rule-based inference mechanisms has greatly contributed to the popularity of the hybrid knowledge-based recommender systems (Martín-Vicente et al., 2014); (L.-C. Chen et al., 2015); (Movahedian & Khayyambashi, 2014); (Al-Hassan et al., 2015). In those recommender systems, knowledge-based recommendation techniques are commonly used in conjunction with the remaining types of recommendation techniques in order to leverage each other's strengths while overcoming each other's weaknesses.

Nowadays, collaborative filtering-based recommendation techniques could be considered the most widely used type of techniques along recommender systems in a variety of disparate domains. Furthermore, one of the most popular approaches in the category of model-based collaborative filtering techniques is the approach based on statistical models, besides the approaches based on dimensionality reduction models and matrix decomposition models. Those approaches formally come from the intersection of the research filed of machine learning (and, in general, artificial intelligence) and statistics as a discipline (Cacheda et al., 2011). On the other hand, different models at the intersection of machine learning and data mining research fields have been widely used in the construction of model-based collaborative filtering techniques (Amatriain et al., 2011).

In the context of the statistical modeling approaches, latent class statistical models, particularly the Latent Semantic Analysis (LSA) model, which is based on a statistical modeling technique that introduces latent class variables into a mixed model with the aim of discovering user communities and prototypical user profiles, have been shown to have greater precision and scalability when compared to memory-based collaborative filtering techniques (Hofmann, 2004). Furthermore, in the context of the discovery of latent topic structures from collections of documents or historical usage data, the Latent Dirichlet Allocation (LDA) generative probabilistic topic model has traditionally been considered a reasonable alternative to the Probabilistic Latent Semantic Analysis (PLSA) model, which can be considered a specialization of the LSA model.

Based on the hypothesis that the combination of Semantic Web technologies and latent class statistical models will enable the design and implementation of more powerful techniques for representing and modeling contextual information as well as more powerful context-aware recommendation techniques, in this research it is proposed a hybrid knowledge-based and collaborative filtering context-aware recommendation method using a statistical latent class model-based approach for the restaurant domain in the context of location-based social network APIs (Application Programming Interfaces) and user opinion website APIs.

The existing proposals on the application of context-aware recommendation techniques and tools to the restaurant domain employ, either semantic Web technologies, commonly the vocabulary definition and description language, OWL, and the SWRL rule language, or latent class statistical models, commonly the LDA probabilistic generative topic model, with two different purposes: (1) the definition of hybrid techniques for mining, modeling and representation of contextual information and (2) the definition of hybrid recommendation techniques. There are, however, practically no proposals that take advantage of the possible conjunctures between those types of computational technologies and techniques.

In fact, leaving aside the tourism domain, unlike the leisure domain, which represents an economic sector related to activities not necessarily supported by tourists, third industries related to activities partially supported as a result of the tourist influx, for example the restaurant Industry, do not currently have the required support. In this sense, this research intends to serve as a starting point for other academics and developers in the research field in recommender systems in an attempt to build an open ecosystem of semantic data and applications in the restaurant domain homogenized and integrated under a Linked Data approach, and eventually to help close the aforementioned gap in the long term.

6.2. State of the art

6.2.1. Recommender Systems

The first advances in the research field of model-based collaborative filtering recommender systems were based on techniques such as Bayesian clustering (Breese et al., 1998), Bayesian networks (Y. Chen & George, 1999) and dependency networks. Nonetheless, the work by Hofmann on probabilistic Latent Semantic Analysis (PLSA)-based collaborative filtering and user data mining algorithms (Hofmann, 2004) marked a turning point in the research on model-based collaborative filtering recommender systems in an attempt to improve the accuracy of the predictions of recommender systems. Difference with previous works lies in the fact that the work by Hofmann is based on a model of latent variables that introduces the concepts of user communities and item groups, whereas Bayesian-based techniques and dependency networks build a dependency structure directly from observed variables.

With the emergence of the LDA model in 2003 (Blei et al., 2003), research on model-based collaborative filtering systems took a new direction in the context of topic modeling in the field of text mining. This model is considered an improvement over PLSA, and it in turn can be interpreted as a generalization of the latter in certain circumstances. Formally, LDA is a probabilistic generative model of a corpus in which documents are represented as random mixtures of latent topics; each topic is characterized by a distribution over words in

documents, and there is an a priori Dirichlet distribution for both the document-topic distribution and the topic-word distribution.

Collaborative filtering techniques can be classified into two main groups: (1) memory-based techniques and (2) model-based techniques. In general, memory-based techniques involve the use of a sample or even a complete rating database in generating predictions for items unknown to users, while model-based techniques involve using the ratings database to first learn a model that allows the subsequent generation of predictions. At this point, it is important to mention that the domain of the information of the collaborative filtering recommendation techniques usually corresponds to a database of ratings assigned by a set of users to a set of items; such ratings can be considered a form of low-level user preferences.

Among memory-based collaborative filtering techniques, the primitive type of collaborative filtering techniques, two main predominant approaches can be distinguished: (1) neighborhood-based collaborative filtering and (2) collaborative filtering of top-n recommendations. The first approach is based on the principle known as "word-of-mouth" according to which people are susceptible to be influenced by the opinions of other people with similar thoughts to in assessing the value of an item regarding their own interests. Hence, the original memory-based collaborative filtering approach is known as user-based neighborhood. In this context, it should be mentioned that collaborative filtering techniques can also be classified into two different categories depending on whether the similarity is calculated with respect to the users or with respect to the elements or, in other words, depending on whether groups of similar users or groups of similar items are constructed (1) user-based collaborative filtering and (2) item-based collaborative filtering (Aggarwal, 2016).

Model-based collaborative filtering techniques emerged as an alternative to memory-based techniques; more specifically, they emerged in an attempt to solve the problems of memory-based techniques, namely the limitations of scalability and real-time performance, as well as the high sensitivity to the problems of scarcity of data and cold start, which, due to the nature of the collaborative filtering mechanism, affect to a greater or lesser extent any technique based on it.

The concept of model-based collaborative filtering was formalized by Breese et al. in 1998 (Breese et al., 1998) to refer to the class of collaborative filtering techniques that exploit any kind of predictive model in order to represent the behavior, interests or preferences of the users from the data of a community and, using this representation, make intelligent predictions in the collaborative filtering-based recommendation process. Throughout history, different models that exist in the intersection of the research areas on machine learning (and artificial intelligence in general) and, on the one hand, the research area on data mining and, on the other hand, the discipline of statistics, have been widely used in the construction of model-based collaborative filtering techniques. In the first case, it is possible to mention: a) association rules, b) sequential patterns, c) clustering models and d) artificial neural networks; while in the second case it is possible to mention: a) dimensionality reduction models (e.g., Principal Component Analysis, PCA), b) matrix decomposition models (e.g., Singular Value Decomposition, SVD), c) linear models (e.g., linear regression models) and d) statistical models such as Bayesian network models and statistical latent class models (Cacheda et al., 2011); (Amatriain et al., 2011).

According to Hofmann (Hofmann, 2004), collaborative filtering techniques based on statistical latent class models can be considered as a separate family within model-based collaborative filtering techniques. In general, such techniques are based on statistical modeling techniques that introduce latent class variables into a mixed model with the aim of discovering user communities and prototypical user profiles.

The PLSA model can be considered as one of the most popular models within the category of statistical latent class model-based collaborative filtering techniques. However, in the context of the discovery of latent topic structures from collections of documents or historical usage data, the LDA model has historically been considered a reasonable alternative to the model mentioned above for two reasons. (1) By not carrying out the

generative process at the level of documents, it is difficult to use the PLSA model to assign probabilities to new documents, i.e., documents that are not present in a training corpus. 2) Related to the above reason, the number of parameters to be estimated in a PLSA model grows linearly with the number of training documents.

In particular, the work of Burke, Hammond and Young (R. D. Burke et al., 1997) describing the approach to assisted exploration of multidimensional information spaces called "FindMe" can be considered a precursor to the Knowledge-based Recommender Systems (KBRS) literature. This approach combines search and information exploration techniques with knowledge-based information retrieval techniques to address the problem of information overload on the Web. In that paper, the concept of knowledge-based similarity metric is introduced to refer to the measurement of the proximity of two products (depending on a specific characteristic) in the fulfillment of the same objective; the concept of recovery strategy is also introduced, and it refers to algorithms designed to capture a particular notion of similarity through the orderly execution of different metrics.

It was not until the 2000s, particularly the second half, that KBRS research took a new direction thanks to the first results in the development of the Semantic Web, specifically after the recognition of the specification of the first version of OWL as a W3C (World Wide Web Consortium) Recommendation in 2004. In this context, the first proposals focused on the use of OWL ontologies as a means to infer implicit knowledge not only from the formal conceptualization of the underlying domain to a recommender system -domain ontology, but also from the formal conceptualization of the preferences of its users (Middleton et al., 2004); (Blanco-Fernandez et al., 2006); (I. Cantador et al., 2006).

An important aspect of the pioneering research in KBRS and Semantic Web is that the proposed recommendation systems are hybrid systems that use knowledge-based recommendation techniques, specifically techniques based on semantic knowledge (Semantic Web technologies) along with techniques considered more traditional, such as collaborative filtering and content-based filtering. This is because, given their nature, knowledge-based recommendation techniques were originally proposed as a complementary approach to existing recommendation techniques, rather than as a competing approach.

The development of recommender systems that leverage data published under the Linked Data approach is currently attracting the attention of KBRS and Semantic Web researchers. (Di Noia et al., 2012); (Peska & Vojtas, 2013); (Figueroa et al., 2015, p.); (Allahyari & Kochut, 2016).

Delving into the topic of ontology-based semantic similarity metrics, which is the kind of knowledge-based recommendation techniques that concern this research, it should be noted that the classification proposed by Sánchez et al. in (Sánchez et al., 2012) has been taken as a reference in this research. According to this classification, it is possible to distinguish between: (1) path-based approaches (graphs), 2) approaches based on ontological features, and 3) approaches based on the concept of "information content" from information theory.

On the one hand, path-based metrics assume that an ontology can be considered a directed graph in which the concepts are interrelated mainly by taxonomic relations (is-a) and, to a lesser extent, by non-taxonomic relations. This type of metrics allows calculating the similarity between two terms by mapping them to the concepts in an ontology and calculating the length of the path (sequence of arcs represented by taxonomic relations) between these concepts. In this sense, it is important to bear in mind that in the literature there are different interpretations of the concept of minimum path, with the corresponding implications that this can have.

On the other hand, metrics based on sets of ontological characteristics arose from the need to overcome another important limitation of the kind of metrics mentioned above: the assumption that taxonomic relations in ontologies represent uniform distances. They allow calculating the semantic similarity between two concepts as the degree of overlap between sets resulting from interpreting concepts as functions of their properties. This approach is

based on Tversky's model of similarity that comes from set theory (Tversky, 1977), and considers both the common features and the uncommon features of the concepts.

As will be explained in detail in the next chapter, the semantic ontology-based similarity metric proposed as part of this research is similar in nature to the metric proposed in (Blanco-Fernández et al., 2008b), since it is aimed at capturing the two different types of knowledge commonly available in ontologies: taxonomic knowledge and explicit and implicit non-taxonomic knowledge. This metric comprises two components corresponding to the two types of knowledge mentioned above, which, in turn, correspond to the approaches of path-based similarity (graphs) and similarity based on sets of ontological characteristics, respectively.

The work by Balabanović & Shoham, which proposed a multiagent software architecture and software system for Web page recommendation (Balabanović & Shoham, 1997), can be currently considered a pioneer technique for hybridization of recommendation techniques, specifically collaborative filtering and content-based filtering techniques.

According to the research by Burke (R. Burke, 2002) in which the concept of hybrid recommender system, as well as the different hybridization approaches of recommendation techniques used by existing hybrid systems to date were formalized, the extension to the Internet news filtering architecture and software system "GroupLens" proposed by Sarwar and collaborators (B. M. Sarwar et al., 1998) represents in fact the first knowledge-based hybrid recommender system.

The extended collaborative filtering approach proposed by Mobasher, Jin, & Zhou in 2004 (Mobasher et al., 2004) deserves special mention, as it can be currently considered the first hybrid knowledge-based recommendation approach, specifically the first hybrid semantic knowledge-based recommendation approach (Semantic Web technologies). In that recommendation approach, structured semantic knowledge about the elements to be recommended is automatically extracted from the Web using domain ontologies as a reference; this knowledge is then used in combination with the ratings granted in the past by users (memory-based collaborative filtering) to predict ratings for elements unknown to them.

Based on the analysis of a series of recommendation techniques proposed to date (2002), in terms of the input data and the phases of the recommendation processes, i.e., the algorithms, Burke (R. Burke, 2002) identified and formalized seven recurring hybridization methods in systems combining two or more recommendation techniques of different nature, namely: (1) weighted, (2) switching, (3) mixed, (4) feature-based combination, 5) cascade, (6) feature augmentation and (7) meta-level.

Furthermore, the weighted method consists of predicting the ratings that users would assign to items from the weighted results of all the recommendation techniques available in the recommender system, with the advantage that it is relatively easy to make adjustments to the hybridization and the disadvantage represented by the implicit assumption that the relative value obtained by each one of the combined techniques is uniform throughout the set of all items.

The concept of a new type of recommender system (not related to the type of technique used to calculate similarities or to predict ratings) with the ability to suggest elements of possible interest to the user in specific circumstances, that is, starting from information about the context in which recommendations occur was formalized In the early 2000s: context-aware recommender system (Adomavicius & Tuzhilin, 2008). This type of recommender system emerged with the aim of preserving potentially useful information for rating prediction, which could ultimately result in more relevant recommendations from the user's point of view.

In addition to formalizing the concept of a "context-aware recommender system", (Adomavicius & Tuzhilin, 2008), different interpretations of the concept of "context" across different areas of research in the disciplines of Computer Science and Information Science related to the area of recommender systems, e.g., ubiquitous and

mobile computing, e-commerce and data mining. In that paper, it was concluded that the concept of "context" is a multifaceted concept used throughout different disciplines, each of which leaves a certain mark in its definition.

Adomavicius et al. (Adomavicius et al., 2011) were a step beyond the perspectives in defining the concept of context proposed in previous works and defined six different dimensions of contextual information in the particular field of recommender systems from the possible combinations between the possible values of two identifiable characteristics of the contextual factors in that context. Specifically, the authors defined the following characteristics and values: (1) the knowledge that the recommender system has about the contextual factors, that is, if these factors are: (a) fully observable factors, (b) partially observable factors or (c) unobservable factors, and (2) whether the contextual factors change over time, i.e., whether they are: (a) static factors or (b) dynamic factors.

In the particular case of the type of contextual information that comprises partially observable attributes or unobservable attributes, latent variable models such as PLSA and LDA have been extensively exploited for the purposes of representing and modeling under the assumption that user interactions involve a set of relatively small contextual "states" that can explain the behavior of users at different points along the interactions (Mobasher, 2014). In this sense, these models can be considered types of probabilistic graphical models in which graph notation is used to express conditional dependence structures among random variables (Koller & Friedman, 2009).

According to Adomavicius and Tuzhilin (Adomavicius & Tuzhilin, 2008), the most recent trend in the incorporation of contextual information in recommender systems corresponds to the modeling and learning of context-aware user preferences, which they formally call "contextual preference mining and estimation", and it involves the application of intelligent techniques of data mining and machine learning. However, context-aware recommendation processes based on such an approach can also take one of three distinct forms depending on the phase of the recommendation process in which contextual information is exploited, in other words, depending on the component of the process in which the information is exploited.

Furthermore, the "contextual pre-filtering" paradigm involves using contextual information at the early stage of the 2D (user x item) recommendation process to select or construct (pre-filter) the most relevant user and item data space for the generation of recommendations. In other words, this paradigm can be interpreted as a paradigm of reduction of multidimensional context-aware recommendation processes to 2D recommendation processes, allowing the use of any 2D recommendation technique proposed in the literature.

6.2.2. Discussion

From the results of the analysis of the state of the art carried out in this research it can be deduced that the domains of application of recommender systems represented by third-party economic sectors related to activities partially supported as a result of the tourist influx, for example, the restaurant industry, currently do not have the support of context-aware recommendation techniques and tools to the same extent that the domain represented by the leisure industry, even though this industry is not necessarily related to activities supported by tourists.

Furthermore, the few contributions related to existing context-aware recommender systems for the restaurant domain do not take advantage of the possible conjunctures between Semantic Web technologies and statistical latent class models for the definition of both more powerful techniques for representing and modeling contextual information and more effective recommendation techniques, and employ either Semantic Web technologies, specifically the vocabulary definition and description language, OWL, and the rule language, SWRL, or statistical latent class models, specifically the generative topic model, LDA.

In addition, these works do not explore the importance of complex contextual information related to the social situations of users, which is presumed to be as important as complex temporal information and other common types of simple contextual information, such as location information, in the specific domain of restaurants.

With respect to the ontology-based semantic similarity metrics proposed in the literature on knowledge-based recommender systems, it is worth mentioning that most of these employs either an approach based on graph paths, or an approach based on sets of ontological characteristics. In fact, very few existing ontology-based semantic similarity metrics use a hybrid approach based on graph paths and sets of ontological characteristics with the aim to exploit, on the one hand, taxonomic relations and, on the other hand, non-taxonomic relations including explicit relationships and implicit relations.

Based on the results of the study of the state of the art on knowledge-based recommendation and Semantic Web, it is possible to observe a trend in current proposals towards the use of data published under the Linked Data approach. However, more research on the use of this approach in conjunction with context-aware recommendation approaches beyond traditional recommendation approaches is still required.

6.2.3. Semantic Web

In an attempt to make possible a vision of the World Wide Web in which the meaning of the content of Web pages is structured, giving rise to an environment in which there are software agents that are able, not only to display such content, but also to automatically process and "understand" it to quickly perform tasks that are useful for users, Tim Berners-Lee, James Hendler and Ora Lassila (Berners-Lee et al., 2001) coined the term "Semantic Web" in the early 2000s.

Visualized, not as separate from the document and hypertext links-based Web -traditional Web, but rather as an evolution or extension of it, in which information is given well-defined meaning in order to enable collaborative work between users (human beings) and software agents (machines), Berners-Lee et al. defined the main "building blocks" necessary for the materialization of the Semantic Web: knowledge representation languages, ontologies and agents.

From these primitive building blocks, and on the basis represented by the eXtensible Markup Language (XML) markup language, the XML Schema schema language, the Uniform Resource Identifier (URI) technology, and the UNICODE character encoding standard, all of them representing essential components of the traditional Web, it was formalized a first approximation to the architecture of the Semantic Web, which is currently in development under the leadership of the W3C.

In this context, it is necessary to open a parenthesis to mention that W3C defines Semantic Web as: "A common framework for the exchange and reuse of data between applications, companies and communities. A collaborative effort led by the W3C with the participation of a large number of partners from academia and industry.".

The architecture layer represented by URI and UNICODE as a whole allows the consistent management of information regardless of the writing system in which it is represented, as well as its unambiguous identification (naming and location) within the Web.

The layer represented by XML and XML Schema represents, on the one hand, the mechanism for the transmission and "reading" of the structured information between the agents in the Semantic Web and, on the other hand, the foundation of the mechanism that allows retrieving and querying the meaning of such information, and not only the retrieval of the documents themselves. In fact, XML allows documents to be given certain structure, but does not allow indicating anything about the meaning of this structure.

The layer on which the RDF (Cyganiak et al., 2014) and the RDFS technologies (Brickley et al., 2014) are the core of the architecture, as it allows expressing the meaning of information or, properly speaking, of resources, where a resource is any abstract or real-world entity, such as a number or a physical entity. A structure of triplets, in which each triplet is composed of a subject, a predicate and an object, as with any fundamental sentence is defined for this purpose; this representation is intended to offer a natural way to describe the vast majority of data processed by the machines. A set of triplets thus represents a "network" of information about related entities, and because RDF uses IRIs (Internationalized Resource Identifiers) to encode such information into documents, it ensures that concepts are not just words in the document but unique definitions that can be accessed by any person or agent on the Web. Furthermore, unlike XML (Bray et al., 2008) and XMLSchema (Fallside & Walmsley, 2004), RDF and RDF Schema are oriented towards the effective and efficient integration of data.

The layer called "ontological vocabulary" allows defining collections of information in the form of taxonomies and sets of rules called "ontologies"; with which it is sought to improve the functionality and expressiveness of the Web with respect to the capabilities provided by the previous layer. The term "ontology" comes from Philosophy, specifically from the branch of Metaphysics, and refers to theories about the nature of existence and the types of things that exist, as well as to the discipline that is in charge of the study of these theories; it was introduced to Computer Science and Information Science by researchers in the fields of Artificial Intelligence, and is currently used to refer to formal definitions of relationships between concepts. Although there is no universally accepted definition of the term "ontology" within the aforementioned fields, one of the most widespread definitions today is the one proposed by Studer, Benjamins and Fensel (Studer et al., 1998): “A formal and explicit specification of a shared conceptualization.".

The logic layer represents the effort to add to the Web the ability to use rules to make automatic inferences about data, to take courses of action and to answer questions, that is, to generate new knowledge from the available explicit knowledge.

The first-level layers ("proof layer" and "trust layer") refer to a fundamental facet of the operation of agents in the Semantic Web: the exchange of "checking tests"; in particular, this operation is related to the operation of the previous layer in the sense that the rules are executed under a security (confidence) mechanism that allows evaluating their results and determining if it is appropriate or not to trust the evidence provided. The development of these layers, and the "digital signature layer", whose aim is the identification of alterations in documents within the Web, is part of the ongoing research of the W3C working groups.

Linked Data is the remaining piece of the puzzle that represents the Web of Data. This piece is what allows applications in the environment enabled by Semantic Web technologies to access not only the data in its pure state but also the relationships between them, in order to materialize the Web of Data as a collection of interconnected datasets which is popularly known as Linked Data

According to Tim Bernes-Lee's Linked Data design note (2006) [7], there are four key principles behind Linked Data as a paradigm: (1) the use of URIs to identify "things", (2) the use of HTTP (HyperText Transfer Protocol) URIs in such a way that individuals and computers (software agents) can easily find such identifiers, (3) the provision of relevant information about "things" as a response to the search for identifiers, and (4) the inclusion of links to other URIs in order to allow the discovery of new "things".

[7] https://www.w3.org/DesignIssues/LinkedData.html

6.2.4. Objectives

6.2.4.1. General and Specific Objectives

The objective of this thesis is to design and implement a hybrid knowledge-based and collaborative filtering context-aware recommendation method using a statistical latent class model-based approach for the restaurant domain.

This objective can be broken down into a series of particular objectives which are mentioned below.

- Design and build an ontological model (and rule base) of the restaurant domain that allows integrating and linking heterogeneous location-based social network and user opinion Web site APIs.
- Design and implement a contextual information modeling technique based on the LDA probabilistic generative topic model that takes advantage of the ontological domain model to infer high level contextual information.
- Design and implement a semantic similarity metric based on the ontological domain model that allows capturing explicit and inferred taxonomical and non-taxonomical knowledge about food and beverage establishments.
- Design and implement a collaborative filtering recommendation technique based on the semantic similarity metric and ontology-based user profiling techniques.
- Design a software architecture that allows integrating the different techniques designed and implemented in a set of interoperable software components.
- Validate the proposed method and architecture by implementing a prototypical recommender system as proof of concept.

6.1.4.2. Hypothesis

The hypothesis with which it is tried to demonstrate the truthfulness of the thesis of this research can be expressed in the following statement.

The combination of Semantic Web technologies and statistical latent class models will allow the design and implementation of both more powerful contextual information representation and modeling techniques and more powerful context-aware recommendation techniques.

In turn, this hypothesis can be broken down into the following sub-hypotheses.

Sub-hypothesis 1. Semantic Web technologies, specifically vocabulary definition and description languages and rules languages can be used in combination with statistical latent class models, specifically probabilistic generative topic models, for the design and implementation of both techniques for representing and modeling contextual information and context-aware recommendation techniques.

- What are the advantages and disadvantages of existing approaches to the combination of Semantic Web technologies and statistical latent class models for the purposes mentioned above?
- What are the vocabulary definition and description languages and the Semantic Web rule languages most appropriate for the aforementioned purposes?
- What are the statistical latent class models most appropriate for the aforementioned purposes?

Sub-hypothesis 2. The resulting contextual information modeling and representation techniques will allow the representation and modeling of complex contextual information potentially relevant to context-aware recommendation processes.

- What types of complex contextual information are commonly considered by the resulting contextual information modeling and representation techniques?

- Which types of contextual information are most **r**elevant to the restaurant domain?

Sub-hypothesis 3. Resulting context-aware recommendation techniques will make it possible to generate more accurate recommendations in the traditional sense of the word, as well as more realistic recommendations from the user's point of view.

- To what extent do the resulting context-sensitive recommendation techniques improve the accuracy of recommendations?
- To what extent are the resulting recommendations more realistic from the point of view of the user?

6.1.4.3. Methodology

The methodology to follow in this thesis to reach the proposed objectives and to allow demonstrating the thesis of the research mainly comprises four tasks.

- Analysis of the state of the art. This first task represents the study of the state of the art on two main aspects: (1) recommender systems and (2) Semantic Web technologies, with a clear emphasis on the points of convergence between both research areas.
- Formalization of the proposal. This task comprises a series of subtasks described in detail in the next chapter, which result in the definition of a hybrid knowledge-based and collaborative filtering context-aware recommendation method using a statistical latent class model-based approach for the restaurant domain, in the context of location-based social network and user opinion Web site APIs.
- Implementation of the proposal. This task represents the implementation of a prototype of a context-aware recommender system for food and beverage establishments based on a software architecture designed as a result of the proposal formalization tasks.
- Validation of the proposal. This latter task involves the design and execution through a user study and an offline experiment of a two-part evaluation method in the form of comparative analysis based on an Information Science approach, specifically, an information retrieval approach focused on accuracy measurement under two different interpretations: prediction of ratings and prediction of use. The user study and the offline experiment allow evaluating the proposed recommendation method under two different scenarios, respectively: scarcity of ratings and sufficiency of ratings.

6.3. Proposed Method

6.3.1. Proposed Software Architecture

Figure 3.1 shows the software architecture from which it is possible to implement the context-aware recommendation method for the restaurant domain proposed in this thesis. As seen in previous chapter, in order to validate the proposal of this thesis, a prototype of the resulting recommender system has been built; however, this chapter refers to this system as a concept.

At a high level of abstraction, the proposed architecture consists of eight main components, namely the semantic knowledge integrator, the domain ontology, the knowledge repository, the LDA model-based topic discoverer, the context-aware recommendation subsystem, the ontology-based user profiler, the knowledge-based probabilistic collaborative filter, and the user interface. Users interact with this architecture through a single entry-point: the user interface. Along with the geolocation functionality, the user interface has been implemented as a Web-based native mobile application, which is hereinafter referred to as “client application”.

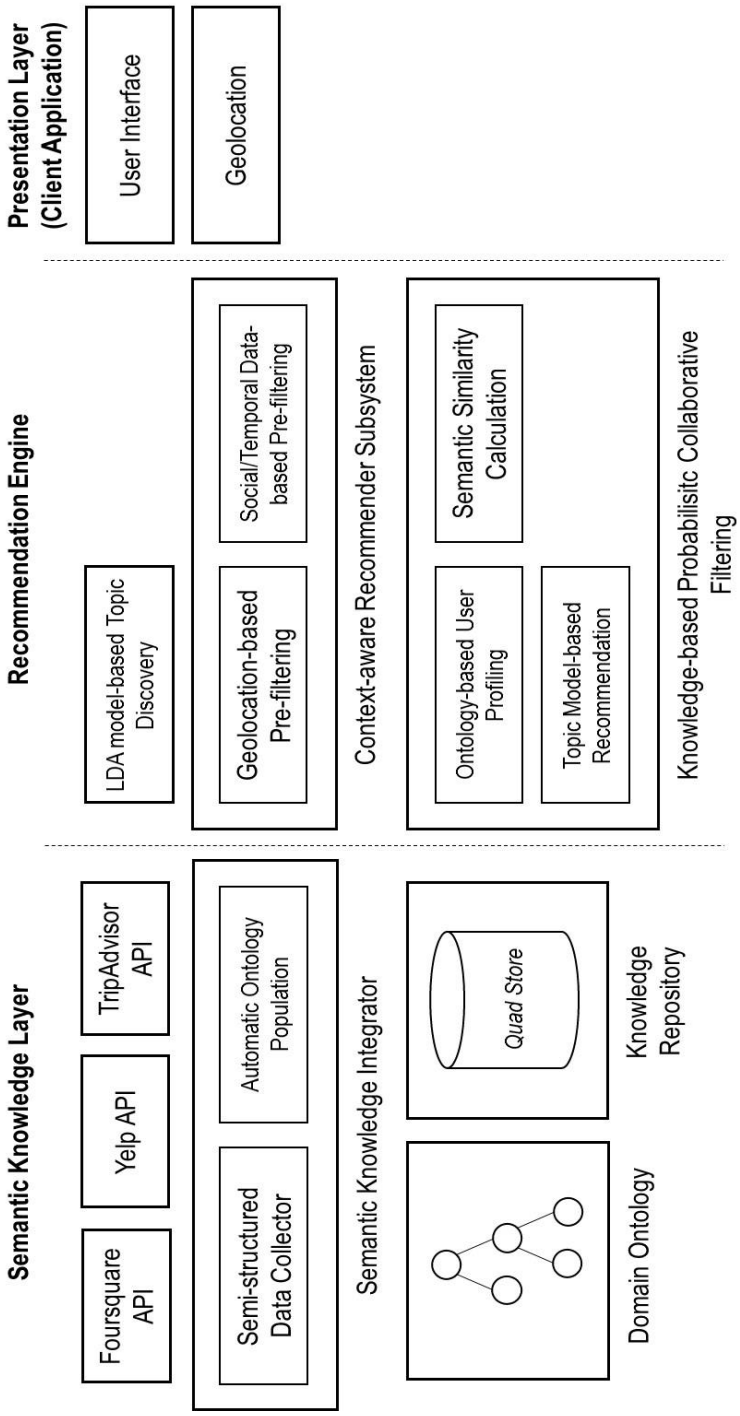


Figure 6.1. Proposed Software Architecture.

6.3.2. Domain Ontology Definition

The domain ontology in this thesis was manually defined using the OWL language (OWL 2). The aim of defining a new ontology for the restaurant domain, although a related proposal may exist, was to link and finally integrate vocabularies of location-based social network APIs and other location-based service APIs such as user opinion Web site APIs that provide heterogeneous content in this domain using a Linked Data approach. This thesis attempts to integrate the vocabularies of the Foursquare, Yelp and TripAdvisor APIs.

The core of this ontology is a class hierarchy representing categories of food and beverage establishments, whose root class is the class "FoodEstablishment", as well as a hierarchy of classes representing categories of cuisine styles, whose root class is the class "Cuisine".

Although location-based social network APIs and other location-based service APIs provide general taxonomies of places, they generally contain subcategories of food and beverage establishments that refer to cuisine styles. However, in some cases, these APIs provide secondary lists of cuisine styles (they are not formally taxonomies) for the particular case of the places in this domain, which can be extended in order to classify each place under one or more categories of food and beverage establishments and, at the same time, one or more categories of cuisine styles.

It should be finally mentioned that the domain ontology comprises classes and data and object properties related to the profiling of users and contexts, in addition to classes and properties related to the modeling of the domain.

Figure 3.2 depicts, respectively, an excerpt (four levels) of the class hierarchy headed by the "Cuisine" class focusing on the subclass that represents the American cuisine (the "AmericanCuisine" class), as well as an excerpt (three levels) of the class hierarchy headed by the "Dish" class focusing on the subclass that represents cereal based dishes (the "CerealDish" class).

6.3.3. Semantic Knowledge Integration

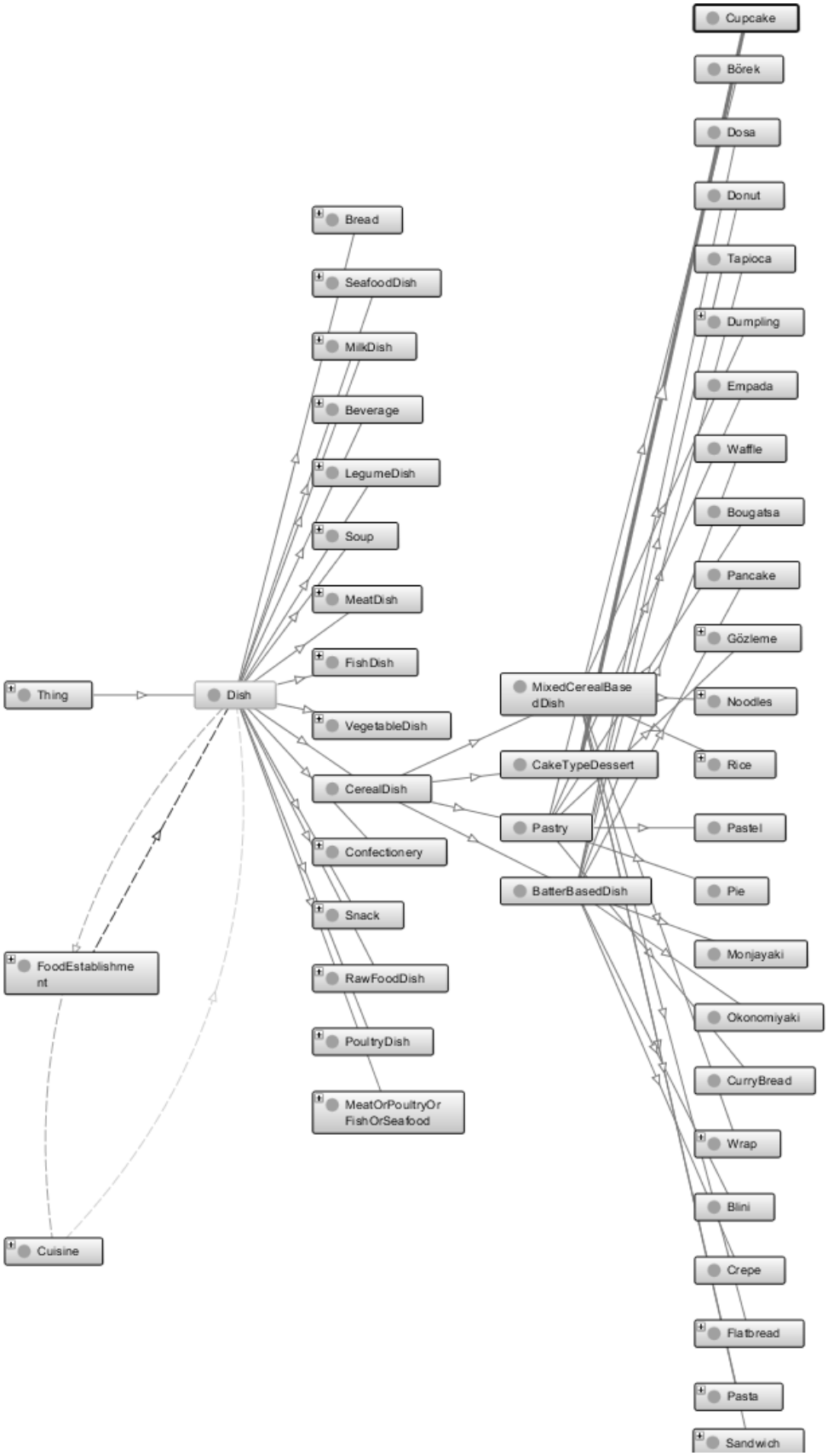
A Java wrapper class hierarchy for the underlying data sources (APIs) was defined for the purposes of semi-structured data gathering so that it is relatively easy to extend the implementation of the proposed architecture to be used with any of the location-based service APIs whose vocabularies are linked using the domain ontology, namely, Foursquare, Yelp and TripAdvisor

This mechanism is based on the previous work of our research group on integration of heterogeneous cloud services, and consists in creating an adapter or wrapper class to adapt the functionalities (services or Web resources) of a particular API to the interface of a class in a class hierarchy, according to its architectural style. There is thus a base class that abstracts the minimum required functionalities of the location-based social service APIs, as well as a pair of child classes that redefine those functionalities according to a particular architectural style, namely SOA (Service Oriented Architecture) and RESTful (REpresentational State Transfer).

In this context, location-based service APIs occasionally expose a Web service or resource for searching for existing food and beverage establishments within an area relative to a given geographical location. In response, such Web services or resources provide a document with as many results in namely XML tags or JSON (JavaScript Object Notation) objects, as found places. Each result provides certain information about an establishment, such as its name and identifier of the establishment.

These APIs also provide Web services or resources that allow obtaining details about a particular establishment, including its category (type of food and beverage establishment), its cuisine styles, its geographical location (latitude-longitude coordinates), and the different components of its address, for example, street, number and city, by querying by its identifier. In response, such Web services or resources provide a document with a single result (XML tag or JSON object) that represents the details of the establishment being queried.

a)


Cupcake
Börek
Dosa
Donut
Tapioca
Dumpling
Empada
Waffle
Bougatsa
Pancake
Gözleme
Noodles
Rice
Pastel
Pie
Monjayaki
Okonomiyaki
CurryBread
Wrap
Blini
Crepe
Flatbread
Pasta
Sandwich
Bread
SeafoodDish
MilkDish
Beverage
LegumeDish
Soup
MeatDish
FishDish
VegetableDish
CerealDish
Confectionery
Snack
RawFoodDish
PoultryDish
MeatOrPoultryOrFishOrSeafood
Thing
Dish
FoodEstablishment
Cuisine
MixedCerealBasedDish
CakeTypeDessert
Pastry
BatterBasedDish

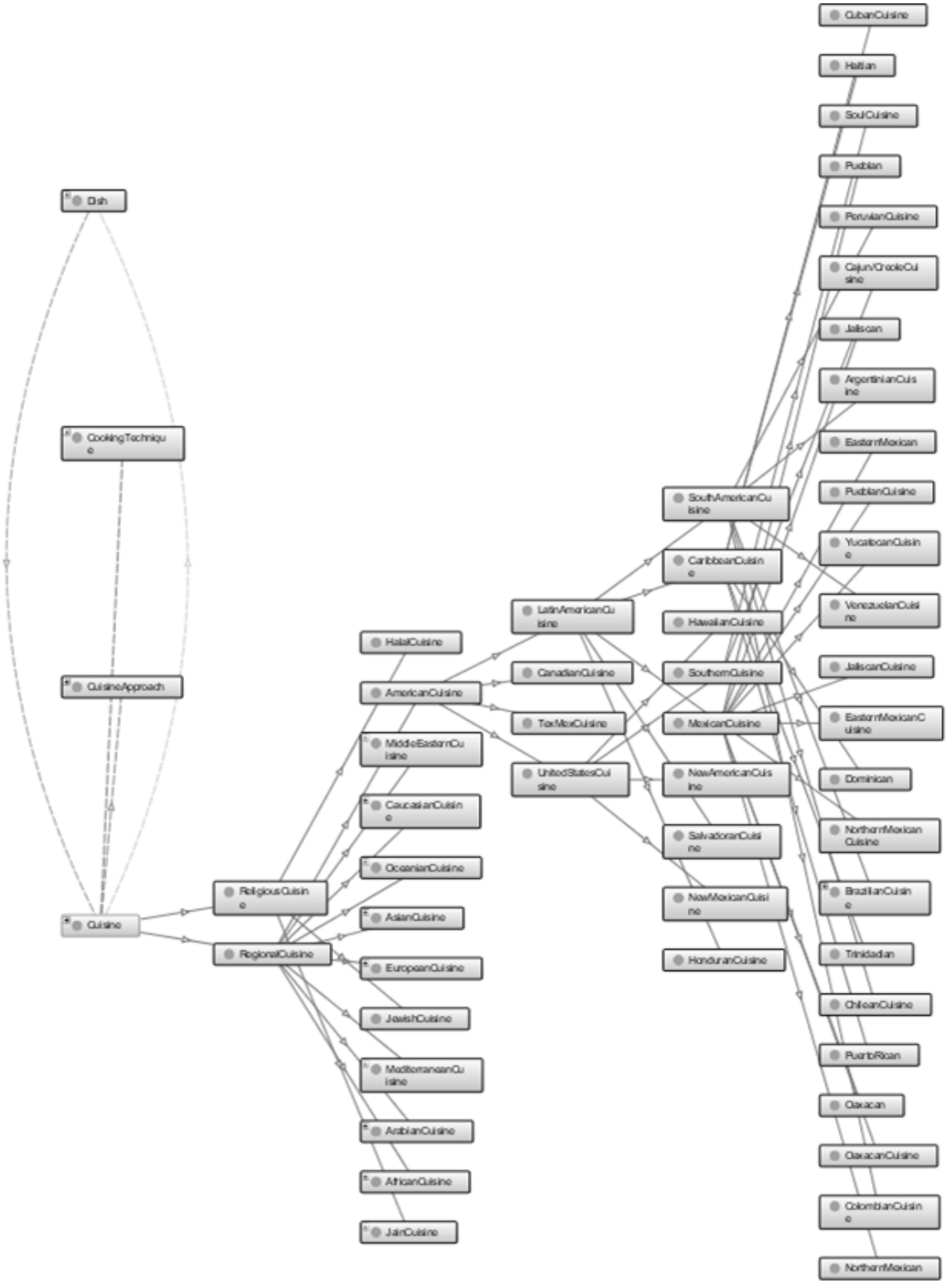


Figure 6.2. An excerpt from the class hierarchies of the domain ontology: a) class hierarchy headed by the class “Dish”, b) class hierarchy headed by the class “Cuisine”.

The next step of the proposed recommendation method consists in the direct mapping of the semi-structured data collected as a result of the previous stage as individuals and property values starting from the classes and datatype and object property values defined in the domain ontology. The Apache Jena framework has been used for this purpose.

At this point it is important to mention that each class and datatype and object property of the domain ontology has been mapped to a corresponding object or attribute (formally to its name) in one or more of the location-based social network APIs and location-based service APIs. Annotation axioms seen as RDF triplets in which the objects are literals of the datatype string representing the names of the objects and attributes have been used for this purpose.

A Jena ontology model is created from a domain ontology graph (serialized in RDF/XML format) stored in a named RDF graph store, specifically as RDF triplets in a second named graph serialized in RDF/XML format. Support for the OWL language, specifically for the OWL-DL dialect, is added to the model, not so for OWL reasoning capabilities. It is worth mentioning that the named RDF graph store used in this research is OpenLink Virtuoso.

It has been proposed to take advantage of the reasoning capabilities of the OpenLink Virtuoso SPARQL implementation with the aim of using the RDFS (subset of RDFS implications) and OWL 1 (subset of OWL-Lite constructs) languages to dynamically or "on the fly" infer additional facts, i.e., RDF triplets representing non-explicit semantic knowledge, from RDF triplets physically stored in a named RDF graph and a series of associated axioms. In fact, Virtuoso OpenLink supports the RDFS properties "subClassOf" and "subPropertyOf" (subclass and sub-property relations), as well as the OWL properties "equivalentClass" (class axioms), "equivalentProperty" and "sameAs" (instance identity axioms).

6.3.4. LDA model-based Topic Discovery

The next step in the proposed recommendation method is the discovery of topics in a corpus derived from check-ins (instances of the "CheckIn" class in the knowledge repository) made by users, using a technique based on the LDA probabilistic generative model.

To do this, it is first necessary to infer high-level contextual information (data type property values whose domain is the "CheckIn" class) from certain low-level contextual information associated with check-ins (instances of the "CheckIn" class). The types of high-level contextual information to be inferred are, specifically, (1) social information, namely the social situations (the "visitedInSocialSituation" datatype property) of the users during the visits to the establishments, which are inferred from the types of social relations between the users and the people who accompanied them during the visits, (2) temporal information, namely the periods of the day and the days of the week (the "visitedAtPeriodOfDay" and "VisitedAtDayOfWeek" datatype properties) in which users, along with their companions, visited the establishments; such information is inferred from the timestamps of the check-ins themselves.

Unlike the RDFS and OWL-based inference process that was previously described, this inference process is outside the scope of ontology-based inference, and rather corresponds to a rule-based inference approach. It has therefore been proposed to use the SPARQL Inferencing Notation (SPIN) notation, which is formally an RDF representation of the SPARQL query language, in order to define these rule sets using the same medium used by the RDFS and OWL-based inference process, i.e., the SPARQL query language.

A SPIN-based inference rule has been created based on the CONSTRUCT expression of the SPARQL query language. This inference rule allows inferring the value of the "visitednSocialSituation" property and, at the same time, the value of the property "visitedAtPeriodOfDay" for each instance of the "CheckIn" class in the knowledge repository.

Once the high-level contextual information is inferred, the LDA model can be built. In this model, attribute values obtained from this information are interpreted as the words, while the documents correspond to the collection of the check-ins occurring at each establishment. This contextual information representation and modeling technique assumes that groups of variables (high-level contextual attribute values) observed in the check-in events can be explained by groups of variables (topics) not observed in these events, and that, finally, there is a significant cooccurrence relationship between these values interpreted as words. This co-dependency relationship makes it possible to recommend food and beverage establishments that best meet users' preferences within specific socio-temporal contexts, that is, topics.

Table 6.1 depicts the possible values for the high-level contextual attributes used as words in the construction of the LDA model.

Domain Ontology		**LDA Model**	
Low-level C.I. (Asserted)	**High-Level C.I. (Inferred)**	**Attribute**	**Value (Word)**
-hasUserCompanion (object property) -isFriendOf, isRelativeOf, isSignificantOtherOf, isSpouseOf, y isCoWorkerOf (object properties, domain "Person" class)	visitedInSocialSituation (datatype property)	SocialSituation	{"friends", "family", "couple", "married_couple", "co-workers"}
timestamp (datatype property)	visitedAtPeriodOfDay (datatype property)	PeriodOfDay	{"morning", "lunchtime", "afternoon", "evening", "night"}
	visitedAtDayOfWeek (datatype property)	DayOfWeek	{"sunday", "monday", "tuesday", "thursday", "friday", "saturday"}

Table 6.1. High-level contextual attributes used in constructing the LDA model (C.I. = Contextual Information).

The LDA-based document clustering algorithm implemented in the LingPipe Java library (the static class "LatentDirichletAllocation") has been used for the actual construction of the LDA model. This implementation mainly performs two functions: (1) it estimates the parameters of an LDA model from an unlabelled corpus of documents given the two Dirichlet priors and a fixed number of topics using a Gibbs sampling, and (2) it builds an LDA model based on the samples provided by the Gibbs sampling from the posterior distribution over topic distributions given the data.

The choice of the number of topics *k* to be used by the LDA model has been regarded as crucial task in this research so that it has been proposed to dynamically calculate this parameter using the stability analysis method proposed by Greene, O'Callaghan, & Cunningham (2014). This method allows calculating the similarity between an LDA model generated from a complete corpus of documents (reference model) and a series of test models generated from different subsets of documents of the same corpus (test models). The similarity between each pair of models is calculated as the level of correspondence, given by Jaccard's coefficient of similarity, among the top ranked terms in their topics, where a high correspondence level means a greater degree of similarity. The repetition of this procedure for a range of tentative values of k eventually allows determining the best option within that range of possible values. This method requires the prior construction of the LDA models to be analyzed; the parallelized algorithm shown in Figure 6.2 has been used for this purpose.

```
static String[] WORDS = new String[] {
        “friends”, “family”, “couple”, “married_couple”, “co-workers”, “morning”, “lunchtime”, “afternoon”,
        “evening”, “night”, “sunday”, “monday”, “tuesday”, “thursday”, “friday”, “saturday”
};

static SymbolTable SYMBOL_TABLE = new MapSymbolTable();
        static {
                for (String word : WORDS)
                        SYMBOL_TABLE.getOrAddSymbol(word);
}

static double DOC_TOPIC_PRIOR = 0.1;
static double TOPIC_WORD_PRIOR = 0.01;

static int BURNIN_EPOCHS = 15;
static int SAMPLE_LAG = 1;
static int NUM_SAMPLES = 16;

short right;
short left;
short nTModels;
File corpus;
short pDocuments;

LatentDirichletAllocation[] rModels;
LdaRunnable[][] tModels;
Thread[][] tModelsThreads;

LDA(short mRight, short mLeft, short mNTM, File mCorpus, short mPDocuments) {
        right = mRight;
        left = mLeft;
        nTModels = mNTM;

        corpus = mCorpus;
        pDocuments = mPDocuments;

        rModels = new LatentDirichletAllocation[right-1];
        tModels = new LdaRunnable[right-left+1][nTModels];
        tModelsThreads = new Thread[right-left+1][nTModels];
}

public LatentDirichletAllocation getLda(LatentDirichletAllocation.GibbsSample sample) {
        return sample.lda();
}

public void sample () {
        short i;
        short j;
        CharSequence [] documents;
```

```
        for(i=0; i<right-left+1; i++) {
                documents = Corpus.readCorpus((short) 100);
                rModels[i] =
                getLda(LatentDirichletAllocation.gibbsSampler(Corpus.tokenizeDocuments(documents,
                SYMBOL_TABLE), (short) (left+i), DOC_TOPIC_PRIOR, TOPIC_WORD_PRIOR,
                BURNIN_EPOCHS, SAMPLE_LAG, NUM_SAMPLES, new Random(), new
                LdaReportingHandler(SYMBOL_TABLE))
                );

        for(j=0; j<nTModels; j++) {
                documents = Corpus.readCorpus(pDocuments);
                tModels[i][j] = new LdaRunnable(Corpus.tokenizeDocuments(documents,
                SYMBOL_TABLE), (short) (j+1), new Random(), new
                LdaReportingHandler(SYMBOL_TABLE));
        }
}

for(i=0; i<right-left+1; i++)
        for(j=0; j<nTModels; j++)
                tModelsThreads[i][j] = new Thread(tModels[i][j]);

for(i=0; i<right-left+1; i++)
        for(j=0; j<nTModels; j++)
                tModelsThreads[i][j].start();

for(i=0; i<right-left+1; i++)
        for(j=0; j<nTModels; j++)
        try {
                tModelsThreads[i][j].join();
        } catch (InterruptedException ex) {
                Logger.getLogger(LDA.class.getName()).log(Level.SEVERE, null, ex);
        }
}

static class LdaRunnable implements Runnable {
        LatentDirichletAllocation mLda;
        final int[][] mDocWords;

        final ObjectHandler<LatentDirichletAllocation.GibbsSample> mHandler;
        final Random mRandom;

        LdaRunnable(int[][] docWords, i, Random random,
        ObjectHandler<LatentDirichletAllocation.GibbsSample> handler) {
                mDocWords = docWords;
                mI = i;
                mRandom = random;
                mHandler = handler;
        }

        public void run() {
                mLda = sample(mDocWords, mI, mRandom, mHandler);
```

```
        }

        public LatentDirichletAllocation sample(int[][] mDocWords, short mI, Random mRandom,
        ObjectHandler<LatentDirichletAllocation.GibbsSample> mHandler) {
                LatentDirichletAllocation.GibbsSample sample;
                sample = LatentDirichletAllocation.gibbsSampler(mDocWords, mI, DOC_TOPIC_PRIOR,
                TOPIC_WORD_PRIOR, BURNIN_EPOCHS, SAMPLE_LAG, NUM_SAMPLES, new
                Random(), new LdaReportingHandler(SYMBOL_TABLE)
                );
                return sample.lda();
        }
}
```

Figure 6.3. Algorithm for generating LDA models for stability analysis.

6.3.5. Ontology-based User Profiling

The next step in the proposed recommendation method represents the entry-point of the recommendation process itself. This is because this stage involves building and constantly updating a characteristic and preference profile for each user. Preferences are updated based on the explicit and implicit ratings given by users to the establishments during the visits to them. The domain ontology or, rather, the knowledge repository, is populated with instances of the "Rating" and "CheckIn" classes as a result of the creation of explicit and implicit ratings.

The total number of check-ins made by a user at an establishment has been considered as an indicator of that user's preference for that establishment, specifically, as that user's implicit rating for that establishment. Since location-based service users often take the time to both check into the places they visit at the moment of the arrival and give explicit ratings for those places after they leave, a user's preference for a place has been calculated as the combination of an explicit rating and an implicit rating (when both are available).

This combination has been actually calculated as the weighted arithmetic mean of the explicit rating given by the user to the establishment and an implicit rating calculated using the Term Frequency-Inverse Document Frequency (TF-IDF) statistical technique.

It has also been proposed to consider user preferences for topics in the LDA model during user profiling. The probability distributions of the topics in the documents representing establishments need to be first calculated for this purpose. The preference of a user u for a topic t is, therefore, calculated as the TF measure of the topic *t* in the user u's check-in history; specifically, it is calculated as the ratio of: (1) the number of check-ins associated to the user u and, at the same time, to the food and beverage establishment p (formally, the corresponding document in the LDA model) for which topic t is the best ranked topic and (2) the total number of check-ins associated to that user.

It is not possible, however, to obtain such feedback from users if no recommendations are made, and it is not possible to produce recommendations if the LDA model, as well as user profiles, are not previously constructed.

The following alternative approach has been proposed to elicit user preferences. During the construction of each user profile, preferences for establishments should be calculated only from explicit ratings obtained by applying a technique for propagating preferences throughout class hierarchies in ontology-based user profiles., for user-establishment pairs for which there is an explicit rating, but not an implicit rating, the explicit rating itself will, therefore, be considered the final rating.

On the one hand, this preference propagation technique consists in presenting to the user a dynamic questionnaire of random questions intended to reveal his/her preferences for at least five different cuisine styles or dishes; On the other hand, these preferences should be assigned to all instances of subclasses of the "FoodEstablishment" class (existing in the knowledge repository) that are associated to the instances of the subclasses of the "Cuisine" and "Dish" classes that represent the corresponding cuisine styles and dishes.

During the construction of each user profile, preferences for topics should similarly be obtained directly from the user through the preference elicitation questionnaire.

As will be explained in more detail in next chapter, during the "training" phase of the implementation of the proposed recommendation method, that is, the phase that corresponds to the collection of users' check-in histories and aims to build the LDA model of high-level contextual information, it will not be possible to compute users' preference for topics, and not even to compute the topic distributions of documents representing establishments. The functionalities of the component of the proposed architecture that is in charge of pre-filtering establishments based on social / temporary data are disabled as a result of this; similarly, the functionality of the component that is in charge of pre-filtering the set of available establishments based on geolocation is partially disabled as a result of this.

Besides requesting the user to establish their preferences through the previously described questionnaire, it is also required that he/she establish other information, specifically, his/her social relationships, in order to enable the construction of the corresponding user profile. Social relationships can be alternatively obtained from users' social networks, specifically Facebook, for which it is necessary that the user provides the credentials of a user account in such social network.

6.3.6. Geolocation-based Pre-filtering

According to Adomavicius & Tuzhilin (2008), the processing of contextual data in a context-aware recommender system can be distributed throughout the different stages of the typical two-dimensional recommendation process: user and element (item). In this research, once the user's preferences have been learned, the recommendations are calculated according to the context-aware recommendation approach described below, which formally corresponds to the pre-filtering paradigm. It has been proposed to distribute the processing of the contextual information, namely location information, social information and temporal information, through two different operations during a pre-recommendation stage (pre-filtering operations): geolocation-based pre-filtering and social/temporal information-based pre-filtering.

During the geolocation-based pre-filtering operation, the geographic location of the user (the active user) must first be estimated in terms of latitude and longitude geographic coordinates. According to the proposed architecture, this activity corresponds to the real-time geolocation functionality that is part of the client application, that is, the hybrid mobile application that materializes the presentation layer of the architecture. It has been decided to, therefore, use the geolocation API provided by the multiplatform mobile application framework used for the implementation of that application, namely the Apache Cordova framework.

Once the geolocation functionality described above has been leveraged in order to identify in real time the geographic position of the user or, rather, the geographic position of the mobile device in which the client application of the proposed architecture is executed, it is possible to identify all food and beverage establishments within the area surrounding that geographical position. It is necessary to resort to the API of the underlying location-based service, specifically to the Web service or resource that allows searching for establishments within an area relative to a specified geographical position.

In order to consider users' preferences for topics in geolocation-based pre-filtering establishments, an algorithm has been proposed for the iterative adjustment of the length of the search radius established by an active user

as part of a request for recommendations. By means of this algorithm it is ensured that the results of the geolocation-based pre-filtering operation sufficiently cover the greatest number of topics of interest to the user, in exchange for the possible minimum search radius increase. This algorithm requires the identification of the topic with the highest probability of being assigned to each document that represents an establishment in the geographic search area (an instance of the "FoodEstablishment" class of the domain ontology).

6.3.7. Social/Temporal data-based Pre-filtering

With regard to the processing of social/ temporal data, it is necessary to first infer the social situation of the active user based on the social relations he has with the people indicated in the request for recommendations (i.e., the people with whom he/she is/will be). It is also necessary to use the timestamp of the request to infer the day of the week and the period of the day during which the user and his companions intend to visit the food and beverage establishments.

In particular, the value of the "hasSocialSituation" datatype property of the instances of the "RecommendationRequest" class of the domain ontology is inferred using one SPIN rule similar to the rule used to infer the value of the "visitedInSocialSituation" datatype property, whose domain is the "CheckIn" class. Likewise, the values of the "hasIntendedDayOfWeek" and "hasIntendedPeriodOfDay" datatype properties, whose domain is the "RecommendationRequest" class, are inferred using an approach similar to that used for inferring the values of the "visitedAtPeriodOfDay" and "VisitedAtDayOfWeek" datatype properties, whose domain is the "CheckIn" class.

Once the high-level contextual information has been inferred from the low-level contextual information associated to the individuals of the "RecommendedRequest" class, it is possible to perform the social/temporal data-based pre-filtering. On the one hand, this corresponds to the estimation of the likelihood of each food and beverage establishment appearing in the outcome of the geolocation-based pre-filtering operation, with respect to the inferred social situation and the intended day of the week and period of the day (words that compose the documents in the LDA model), given the probability distribution of the document that represents it in the LDA model,

To do this, it is necessary to: (1) select the topic with the highest probability of being assigned to the document that represents the establishment (an individual of the class "FoodEstablishment" of the domain ontology), (2) select both the value of the "SocialSituation" contextual attribute obtained from the inferred social situation, as well as the values of the "DayOfWeek" and "PeriodOfDay" contextual attributes respectively obtained from the intended day of the week and the intended period of the day, and calculate the probability of generating each of those values interpreted as words in the selected topic, and (3) calculate the product of the probability of assigning the selected topic to the document representing the establishment and the sum of the previously calculated probabilities.

6.3.8. Ontology-based Semantic Similarity Calculation

The actual recommendation of food and beverage establishments of potential interest to the active user requires a previous phase related to the calculation of the semantic similarities between all the establishments in the recommendation space (individuals of the reference class of the domain ontology in the knowledge repository), in order to identify those establishments similar to those liked by the user, i.e., establishments frequently visited by him/her, as well as establishments for which he/she has indicated their preference in the past.

A new ontology-based similarity metric using a hybrid taxonomic/non-taxonomic-based approach has been proposed for the calculation of semantic similarities in this research. By relying on semantic knowledge about the elements to be recommended and not on the ratings given by the users to the elements, this metric is intended to avoid the "cold start" problem, specifically, the "new element" modality (Su & Khoshgoftaar, 2009).

The values resulting from this calculation are represented in memory in a square matrix of order n - the overall similarities matrix.

In general terms, the proposed metric takes into account not only the number of third individuals/values associated at the same time to the individuals of the reference class being compared but also the number of pairs of third individuals that are not simultaneously associated with the pair of individuals being compared, but which are similar in a taxonomic sense. This has been achieved by integrating the concept of taxonomic semantic similarity proposed in (Sánchez et al., 2012) into the concept of semantic similarity proposed in (Carrer-Neto et al., 2012), which in turn is based on the concept of inferential semantic similarity proposed in (Blanco-Fernández et al., 2008b) and allows capturing explicit non-taxonomic knowledge. It is important to note that, unlike the calculation of the overall similarities between the establishments, the calculation of taxonomic similarities applies only to object properties.

The values resulting from this calculation are represented in memory in a series of matrices of order n (one for each reference object property), where n is the number of instances of the domain ontology class in the range of the corresponding object property -taxonomic similarities matrices.

Once the taxonomic similarities matrices have been built, it is possible to establish the value of a threshold that allows taking into account only the pairs of third individuals having significant taxonomic similarity degrees, that is, only the pairs of third individuals whose degrees of taxonomic similarity are greater than a predetermined value. In this research, it has been proposed to establish this value in the value that separates 50% of the calculated similarity values (ordered from lowest to highest), that is, in the value of the 50th percentile.

Once the threshold value for the taxonomic similarities has been established, the overall semantic similarities can be calculated. An algorithm that includes the calculation of taxonomical similarities (and obviously the construction of the taxonomic similarities matrices), as well as the calculation of the aforementioned threshold has been proposed for this purpose (see Figure 6.3).

```
public double[][] fillTSMatrix(ObjectProperty p){

    OntModel model=OntologyModel.getOntModel();
    Query query = new Query();
    ResultSet results = query.queryAllInstancesOfSubclassesOfRangeClass(model, p);
    List<QuerySolution> list=new ArrayList<>();

    while(results.hasNext()){
       list.add(results.next());
    }

    double[][] tSMatrix = new double[list.size()][list.size()];

    int i, j;
    Resource resource;
    Individual a, b;

    for(i=0; i<tSMatrix.length; i++){

       resource = list.get(i).getResource("subject");
       a = model.getResource(resource.getURI()).as(Individual.class);
```

```
        for(j=i+1; j<tSMatrix[i].length; j++){

            resource = list.get(j).getResource("subject");
            b = model.getResource(resource.getURI()).as(Individual.class);

            tSMatrix[i][j]=tSimilarity(a, b, p);
        }
    }

    return tSMatrix;
}

public double[][] fillOSMatrix(double tST, OntProperty[] ps, double[] ws){

    OntModel model=OntologyModel.getOntModel();
    Query query = new Query();
    ResultSet results = query.queryAllFoodEstablishmennts(model);
    List<QuerySolution> list=new ArrayList<>();

    while(results.hasNext()){
        list.add(results.next());
    }

    double[][] oSMatrix = new double[list.size()][list.size()];

    int i, j;
    Resource resource;
    Individual a, b;

    for(i=0; i<oSMatrix.length; i++){

        resource = list.get(i).getResource("subject");
        a = model.getResource(resource.getURI()).as(Individual.class);

        for(j=i+1; j<oSMatrix[i].length; j++){

            resource = list.get(j).getResource("subject");
            b = model.getResource(resource.getURI()).as(Individual.class);

            oSMatrix[i][j]=oSimilarity(a, b, ps, ws, tST);
        }
    }
    return oSMatrix;
}

public double[][] calculateSimilarities(OntProperty[] ps, double[] ws){

    Object[] tSMatrixes =new Object[ps.length];

    for(int i=0; i<ps.length; i++)
        if(ps[i].isObjectProperty())
```

```
            tSMatrixes[i] = fillTSMatrix(ps[i].asObjectProperty());

        double tSThreshold = calculateTSThreshold(tSMatrixes);
        double[][] oSMatrix = fillOSMatrix(tSThreshold, ps, ws);
        double[][] nOSMatrix = normalizeOSMatrix(oSMatrix);
        return nOSMatrix;
    }
```

Figure 6.4. Algorithm for calculating Ontology-based semantic similarities.

6.3.9. Topic model-based Collaborative Filtering

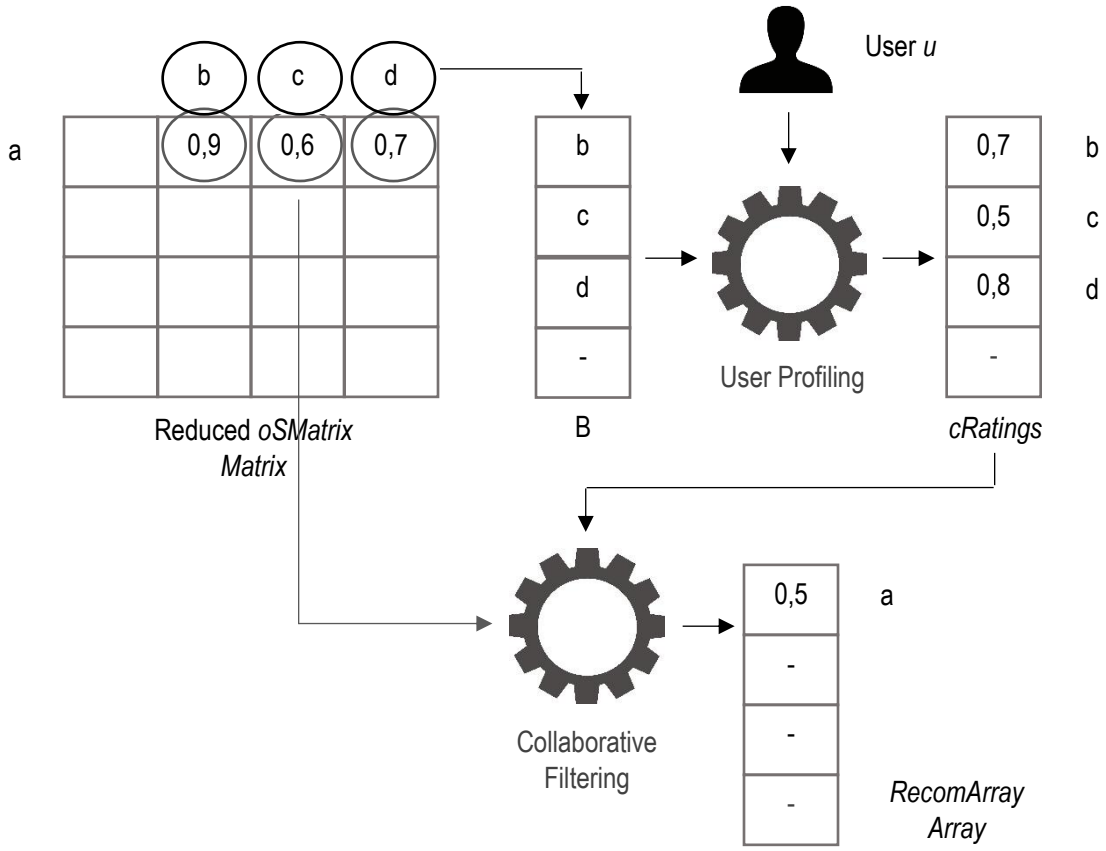
It is possible to calculate the actual recommendations, once the overall similarity matrix has been completed. To do this, it is necessary to first search in this matrix for the elements representing degrees of similarity between individuals of the reference class representing the preferences of the active user and individuals of the reference class appearing in the outcome of the social/temporal data-based pre-filtering operation (henceforth called candidate set of establishments), and to construct a new matrix to represent those elements in memory - the reduced overall similarities matrix.

The reduced overall similarities matrix should be successively reduced to those elements that represent degrees of similarity significant for the calculation of the recommendations, that is, to those elements that represent degrees of similarity greater than a predefined threshold or, in other words, to the elements that correspond to the k individuals representing the preferences of the active user most similar to the individuals representing the candidate set of establishments.

The values that represent the ratings that the active user would give to the establishments in the candidate set of establishments in the reduced matrix of similarities must be then calculated. In this research, the use of the popular method based on the weighted arithmetic mean has been proposed for this purpose. The values resulting from this calculation are represented in memory in an array of order n -the recommendation array, where n is the number of establishments in the candidate set of establishments. In order to facilitate the understanding of the proposed recommendation technique, an example of calculation of recommendations is presented in Figure 6.4.

It has been proposed to finally adjust each value in the recommendations array based on the preferences of the active user for the topics in the LDA model. The topic most likely to be assigned to the document representing the establishment for which the rating will be adjusted need to be selected in advance so that each rating in the recommendations array can be adjusted by adding the value of the user's preference for the topic selected.

As can be inferred from the algorithm for calculating semantic similarities and the recommendation algorithm previously discussed, the context-aware recommendation method for food and beverage establishments proposed in this thesis corresponds to a hybrid knowledge-based probabilistic CF (Collaborative Filtering) recommendation method, and more specifically, to a memory/model-based CF recommendation method (Badrul Sarwar et al., 2001) that relies on an ontology-based semantic similarity metric and a topic model-based recommendation technique.



$$recommendation(a, B, u) = \frac{1}{3}(0{,}7 * 0{,}9 + 0{,}5 * 0{,}6 + 0{,}8 * 0{,}7) = \frac{1}{3}(1{,}49) = 0{,}496$$

Figure 6.5. Example of computation of recommendations.

6.4. Evaluation

6.4.1. Proposed Evaluation Method

A two-part evaluation method based on an Information Science approach has been proposed in this research in order to validate the context-aware recommendation method for food and beverage establishments that represents the contribution of this thesis; specifically, it is an information retrieval approach based on: (1) metrics for measuring accuracy from a classification task perspective and (2) metrics for measuring accuracy from an ordering task perspective.

On the one hand, traditional evaluation metrics from the field of information retrieval, namely, recall, precision and f1-measure have been selected for this purpose; on the other hand, the Normalized Distance-Based Performance Measure (NDPM) metric, which also comes from this field, has been selected for this purpose.

Furthermore, in order to carry out this evaluation in the most reliable way possible, a hybrid memory-based and model-based collaborative filtering recommendation method (baseline recommendation approach) has been implemented; This method lies in both a baseline similarity metric, namely the adjusted cosine-based similarity metric (Badrul Sarwar et al., 2001), and the topic model-based recommendation technique, which represents an integral part of the contribution of this thesis.

6.4.1.1. Contextualization of Evaluation Metrics

With regard to traditional metrics from the field of information retrieval, in this evaluation method, recall (as a measure) has been interpreted as the fraction of food and beverage establishments relevant to a user that are recommended by the proposed recommendation method and the selected baseline recommendation approach. As will be clearly explained in the next section, a food and beverage establishment is considered relevant to a user, if he/she judges it as such with respect to his/her preferences in a given context, namely, in a certain geographical position, social situation, day of the week and period of the day.

On the other hand, precision (as a measure) has been interpreted as the fraction of the recommended food and beverage establishments (recommended by the proposed recommendation method and the selected baseline recommendation approach) that are relevant for a user.

Since the precision and recall measures become more relevant when interpreted together, there are metrics that allows summarizing these measures in a single measurement. The f-measure or f-score metric measures the harmonic mean of the precision and recall measurements obtained for the same query; formally, the harmonic mean of these measures represents only one of the possible cases of the metric fβ-measure (i.e., f_1-measure), where β represents a non-negative real number.

Regarding the NDPM ranking quality metric, in this research it has been proposed its use regarding lists of top-5 establishments (reference rankings). The details about the generation of these lists of establishments are presented in the context of the details about the offline experiment that has been designed and executed in order to carry out the proposed evaluation.

6.4.1.2. Design and execution of the user study

The first phase of the implementation of the proposed evaluation method was a user study in which 48 students of the degree in Information Science at the University of Murcia were invited to participate.

The preferences of the participants for the establishments, that is, the user profiles, were automatically learned from the propagation of the explicit ratings given by the participants, through the preferences questionnaire, to the cuisine styles and dishes. At this stage of the evaluation, the preferences were obtained only from explicit ratings propagated to the establishments from the cuisine styles and dishes (formally from the classes which represent them) through the class hierarchies of the domain ontology, due to the lack of check-in histories associated to the participants. A total of 121 food and beverage establishments in the city of Murcia, Spain, as well as a total of 912 explicit ratings, were collected by the component of the proposed architecture that is in charge of user profiling as a result of the rating propagation process.

In addition, due to the lack of check-in histories from which the LDA model of high-level contextual information could be built, the features of the proposed recommendation method related to context-awareness, specifically the social/temporal data-based pre-filtering operation, could not be involved at this stage of the evaluation but at the subsequent stage.

Nonetheless, two geographic locations of the city of Murcia of possible relevance to the test were selected with the aim of allowing participants to interact with the test prototype while enabling the geolocation-based pre-filtering operation: (1) the "Real Casino" building at the "Trapería" street of the city center and (2) the "Zig Zag" shopping mall at the "Juan Carlos I" street, which is approximately 2 kms. from the city center. Two sets of establishments unknown by the participants were also selected, each composed of 30 establishments available in a geographic area relative to one of the aforementioned locations, where the geographical areas were given by a spherical cap with a base radius of 750 m.

Two groups of 24 participants each were formed, accordingly; one of the groups was randomly assigned to one of the previously selected geographic locations, so that the second group was assigned to the remaining

geographic location. At the end of the test preparations, each participant was asked to judge each of the establishments in the set of establishments associated to the corresponding geographic location as relevant or not relevant to his/her personal preferences.

At this point in the test, a list of top-n recommendations was generated for each participant by the proposed recommendation method, based on the prediction of ratings for the items judged by him/her as relevant or not relevant to their personal preferences during the preparation of the test. Similarly, a second list of recommendations was generated for each participant by using the baseline recommendation approach described in the previous section. The results of the proposed recommendation method were finally compared in terms of recall, precision and f1-measure measures with the results of the baseline recommendation approach; the results of this comparative analysis, as well as the measures themselves, are discussed in detail in the next section.

6.4.1.3. Design and Execution of the Offline Experiment

An offline experiment that allowed the construction of a "synthetic" LDA model of high-level contextual information was carried out in order to evaluate the social/temporal data-based pre-filtering operation. This experiment represented the second phase of the execution of the proposed evaluation method in the sense that it was based on the dataset obtained thanks to the interaction of the participants with the test prototype during the user study.

A total of 726 check-ins were created by randomly involving the establishments, as well as the users (and their companions), in the dataset, resulting in an average of six check-ins per establishment. The extended dataset was used as a training dataset by the component of the proposed architecture that is in charge of the user profiling so that the preferences of the users for the establishments were updated, i.e., they were calculated mixed ratings for the establishments with respect to the users.

On the other hand, the set of 60 establishments in the two areas (of radius equal to 750 m.) relative to the two geographic positions selected for the purposes of the user study was used as a test dataset in this offline experiment. Eight social-temporal contexts were simulated from all possible combinations between the "Friends" and "Family" values for the "SocialSituation" contextual variable, the "Tuesday" and "Thursday" values for the "DayOfWeek" contextual variable and the "Afternoon" and "Night" values for the" PeriodOfDay" contextual variable.

Once the synthetic LDA model was constructed, and the user profiles were updated, the proposed recommendation method was able to produce a list of top-n recommendations for each user. Each list was constructed from the prediction of ratings for the establishments in the geographic area to which the user was assigned, according to the corresponding simulated socio-temporal context. The baseline recommendation approach was also used to generate a second list of recommendations that would lead to the proposed comparative analysis. For each user in the experiment, one request for recommendations was simulated; It explicitly included the high-level contextual information and the corresponding geographical location, besides the predefined search radius (750 m.).

Once the lists of recommendations were generated, a group of 3 students of the degree in Information Science of the University of Murcia (students not related to the user study, which are hereinafter referred to as "experts") was asked to judge every pair of lists corresponding to the same user in order to manually classify each recommended establishment as relevant or not relevant to the user preferences given the corresponding simulated socio-temporal context. In order to identify establishments that were not recommended by the prototype of the system, although relevant to the corresponding request for recommendations, the set of establishments in the geographic area to which each user was assigned was analyzed. From the analysis of the predefined sets of establishments, the five establishments most relevant to each request for recommendation

were identified and ordered in descending order of priority, regardless of whether they were present in the lists of recommendations produced, the lists of top-5 establishments (see previous section). The lists of recommendations were finally compared in terms of measures of recall, precision, f1-measure and NDPM; the measures themselves and the results of the comparative analysis are discussed in detail below.

6.4.2. Results and Discussion

In order to calculate the recall, precision, f1-measure and NDPM measures for both the recommendation method and the baseline recommendation approach, the values for the variables of the corresponding formulas were obtained; certain data related to the recommendations produced by both recommendation methods during the user study and the offline experiment were collected for this purpose.

Figures 6.6 and 6.7 respectively represent the results of the calculation of the recall and precision measures corresponding to the user study for the case of the proposed recommendation method and for the baseline recommendation method. For practical purposes, the values of the variables of formulas representing f1-mesure, contextualized recall and contextualized precision obtained by each participant and simulated user for the case of both recommendation methods are not shown in this document.

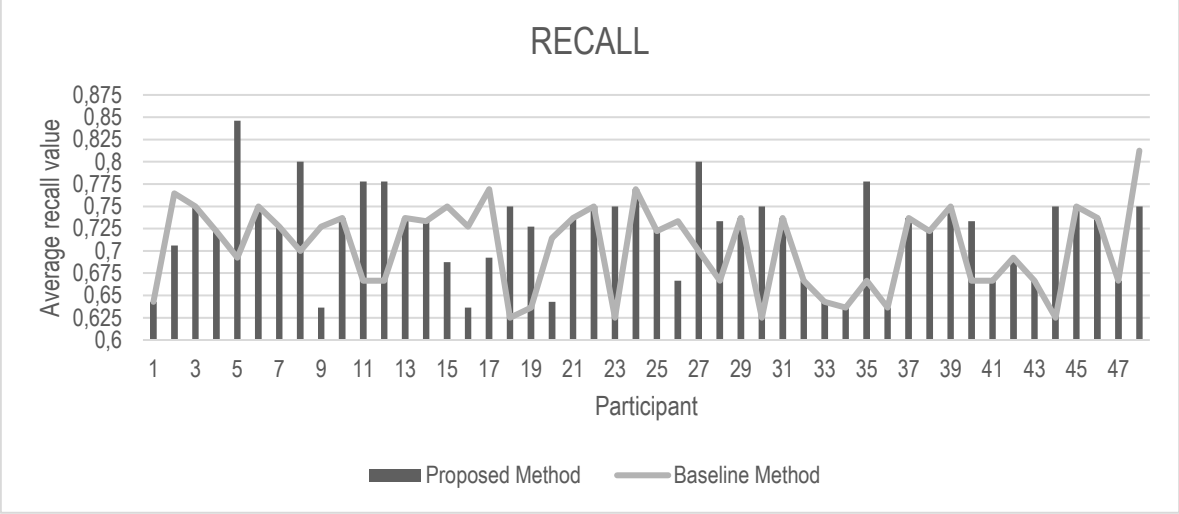


Figure 6.6. Comparison of recall measures calculated for the proposed recommendation method and for the baseline recommendation method (user study).

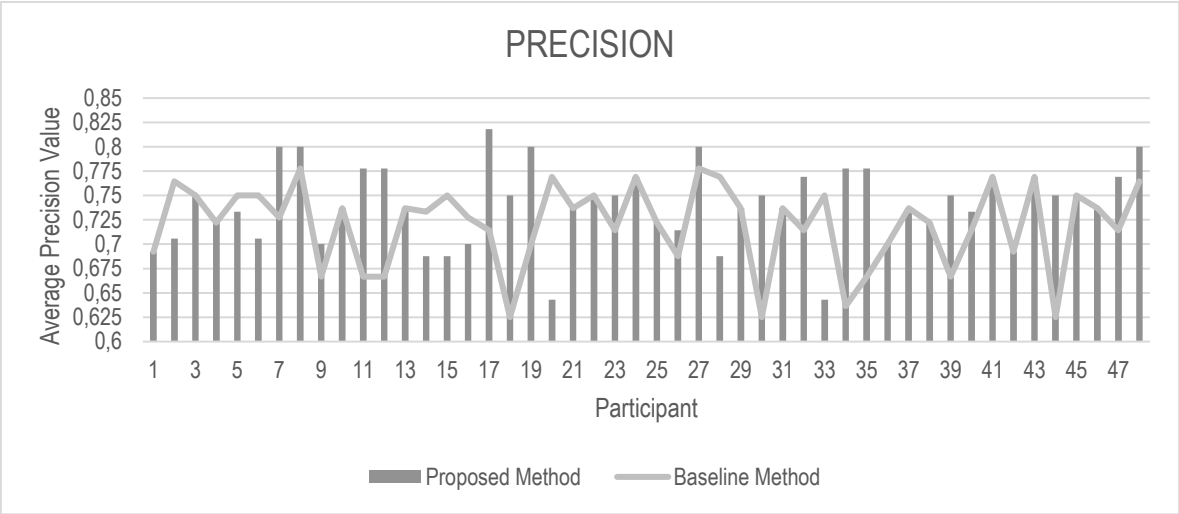


Figure 6.7. Comparison of precision measures calculated for the proposed recommendation method and for the baseline recommendation method (user study).

From Figures 6.6 and 6.7 it can be concluded that in a scenario of scarcity of ratings the proposed recommendation method is able to overcome the selected state-of-the-art recommendation method by 22% both in terms of recall and in terms of precision in the best-case scenario (participant 5 and participant 34, respectively). In these circumstances, the proposed recommendation method is, however, overcome by the selected state-of-the-art method by 14% in terms of recall and 20% in terms of precision in the worst-case scenario (participant 9/participant 16 and participant 20, respectively). In fact, in a scenario of scarcity of ratings, the proposed method is able to overcome the state-of-the-art method by only 2.46% in terms of average recall values and 2.73% in terms of average values of precision. This modest improvement is attributed to the use of the ontology-based semantic similarity metric, which represents an essential part of the proposed method. In this sense, similar behavior is expected against other popular "syntactic" similarity metrics among memory-based collaborative filtering techniques, such as the Pearson's correlation coefficient.

Figures 6.8 and 6.9 respectively represent the results of the recall and precision measurements for the offline experiment, both for the proposed method and for the baseline method. For practical purposes, these figures represent the mean values by simulated socio-temporal context.

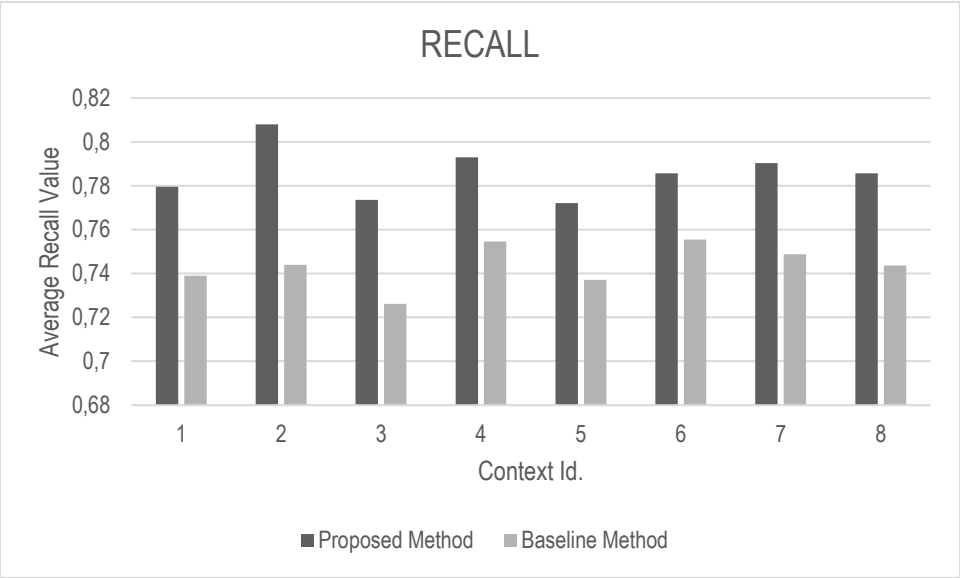


Figure 6.8. Comparison of recall measures calculated for the proposed recommendation method and for the baseline recommendation method (offline experiment)

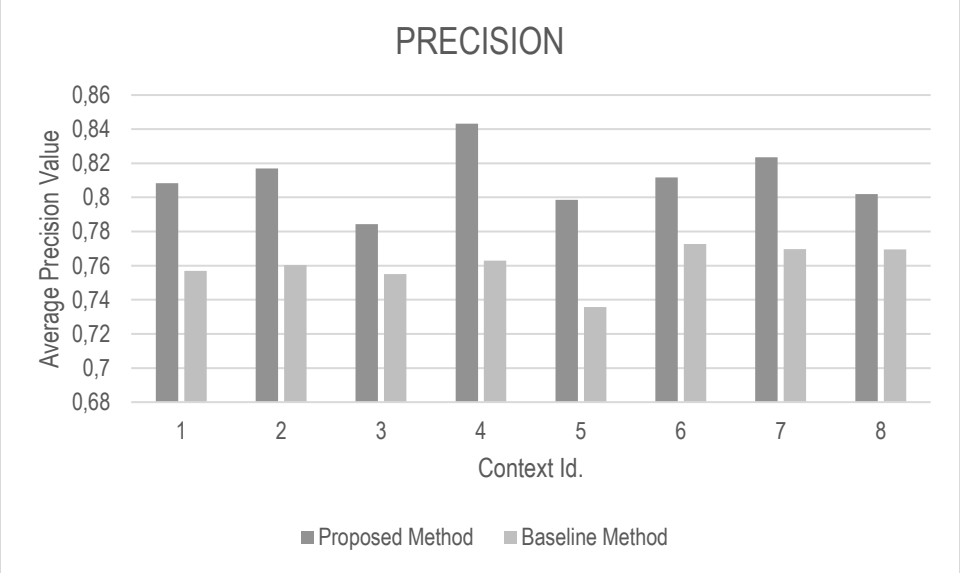


Figure 6.9. Comparison of recall measures calculated for the proposed recommendation method and for the baseline recommendation method (offline experiment)

From Figures 6.8 and 6.9 it can be concluded that, under conditions of normal availability of ratings, the recall and precision values, and especially the precision values, vary slightly between socio-temporal contexts, both in the case of the proposed method and in the case of the baseline method. This could be considered a clear indication that the types of contextual information considered by the technique for representation and modeling of contextual information presented in this document as an essential part of the proposed recommendation method are really relevant for the recommendation as realistic as possible of food and beverage establishments.

Furthermore, the second highest average precision value is reached in the interval corresponding to context 7 ("Thursday", "Night", "Friends") in both cases (the case of the proposed method and the case of the baseline method). Moreover, in the case of the proposed method the highest average precision value is reached in the

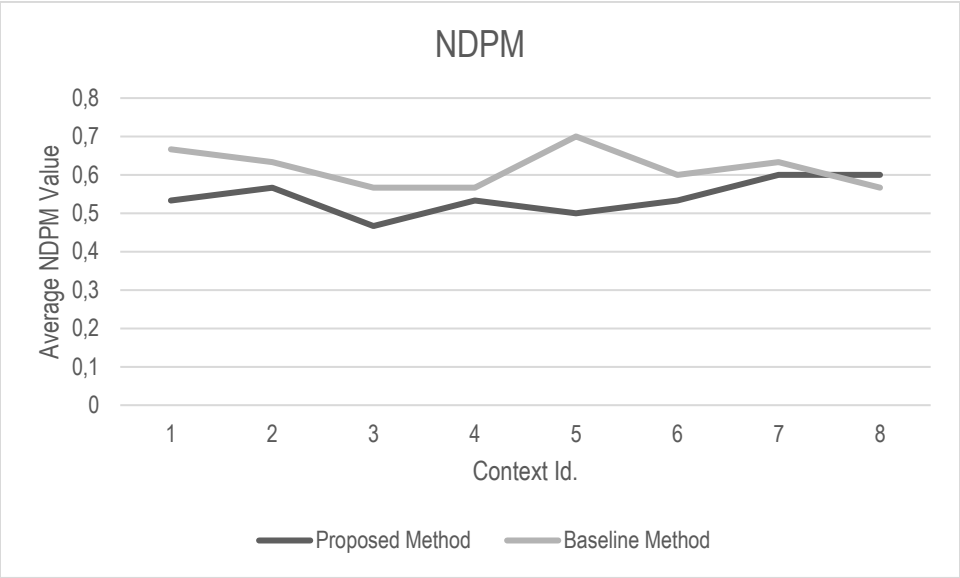
interval corresponding to context 4 ("Tuesday", "Night", "Family"); this increase in precision measurements corresponds to the lowest average number of recommended establishments. The fact that both the temporal contextual information (at least the day of the week) and the social contextual information vary between the two intervals with the two highest precision values, may mean that both types of contextual information are equally important for the prediction of the establishments that best fit the preferences of the users in specific socio-temporal contexts.

Similarly, the lowest average precision value is reached in the interval corresponding to context 3 ("Tuesday", "Night", "Friends") in practically both cases (except for the interval corresponding to context 5 in the case of the baseline method). The lowest average precision value in the case of the baseline method (the interval corresponding to context 5) actually corresponds to the third highest average number of recommended establishments.

With regard to the calculation of the recall measures, the following findings should be mentioned. The mean value reached in the interval corresponding to context 7 represents a minimum decrease in the recall measures in both cases (the case of the proposed method and the case of the baseline method); in fact, it is the fifth lowest average value in both cases. Furthermore, the average value reached in the interval corresponding to context 3 is, in fact, the lowest average value in practically both cases (except for the interval corresponding to context 5 in the case of the proposed method). One possible explanation for this behavior is the size of the ratings matrix, specifically the number of items because, in this research, the recommendation space was reduced to only 30 establishments in two predefined geographic areas for both the user study and the offline experiment.

In summary, under conditions of normal availability of ratings, the proposed method is able to overcome the baseline method by 8.60% in terms of average recall values and by 10.53% in terms of average precision values in the best-case scenario (intervals corresponding to contexts 2 and 4, respectively). Similarly, the proposed method is able to overcome the baseline method by 3.98% in terms of average recall values and by 3.87% in terms of average precision values in the worst-case scenario intervals (corresponding to contexts 6 and 3, respectively). The considerable improvement in performance with respect to the user study can be attributed to the joint use of the ontology-based semantic similarity metric and the LDA model of high-level contextual information, which represents another essential part of the proposed method (formally, the technique for representing and modeling contextual information). As seen throughout the previous section, this has a direct impact not only on the actual modeling of contextual information but also on the profiling of users.

Figure 6.10 depicts the results of the calculation of NDPM measures, for both the proposed recommendation method and the baseline method. This figure like the rest of the figures shown throughout this discussion section represents the average values by simulated socio-temporal context. In the particular case of the NDPM metric, however, the more the average value of the measures approaches the unit, the worse the performance of the information retrieval system (the recommender system, in this case) in reducing the distance between the reference rankings and the rankings produced by the system; In fact, in this case the perfect score is zero.



6.10. Comparison of the calculated NDPM measures for the proposed recommendation method and for the baseline recommendation method.

As can be observed from Figure 6.10, the NDPM measures vary slightly between socio-temporal contexts, both in the case of the proposed method and in the case of the baseline method, which holds true especially for the precision measures. Furthermore, note that the NDPM measures are inversely proportional to the precision measures. One possible explanation for this is that, the more establishments judged to be relevant by a user are recommended by the proposed recommendation method (or by the baseline method), the less likely the contradictory preference relations (C-) between the ranking produced by this and the reference ranking are, and vice versa. In this context, it should be noted that, in this experiment, neither the rankings produced by the recommendation methods nor the reference rankings contained repeated ranks so that the number of compatibility relations (Cu) between the rankings was set to zero for all simulated users; accordingly, since the size of the rankings was reduced to five, the number of priority relations (Ci) among the rankings was set to five for all simulated users.

In summary, in the best-case scenario (the interval corresponding to context 5), the proposed method is able to overcome the baseline method by 28.57% in terms of average values of NDPM (0.5 vs. 0.7). Nevertheless, in the worst-case scenario (the interval corresponding to context 8), the proposed method is overcomed by the baseline method by 5.56% in the same terms (0.57 vs. 0.6). The cumulative average value of NDPM is 0.54 (54%) for the proposed method and 0.62 (62%) for the baseline method. This may be a clear indication that the topic model-based recommendation technique, which is an essential part of the proposed method, allows generating recommendations of food and beverage establishments in a more realistic way when considering the preferences of the users for the topics in the underlying LDA model, and not only the preferences for the establishments.

6.5. Conclusions, Contributions and Future Work

6.5.1. Conclusions and Contributions

Based on the analysis of the state of the art made in this research, it was possible to determine that the tourism and leisure domains represent two very hot areas of application of context-aware recommender systems. This research aimed to contribute technological support to an area that partially depends on the influx of tourists, namely the restaurant industry, through the application of a context-aware recommender system. Furthermore, the particular contributions of this research are intended to serve as a starting point for other academics and

developers of the area of recommender systems in an attempt to build an open ecosystem of data and semantic applications homogenized and integrated under a Linked Data approach.

The specific contributions of this thesis, their main limitations, as well as some future research lines intended to address these limitations in achieving the long-term objective of the research are briefly discussed below.

Semantic Web technologies, mainly the vocabulary definition and description languages, OWL and RDFS, the general-purpose framework for data representation in the Web, RDF, the SPARQL query language for RDF and the SPARQL-based notation for the definition of inference rules and restrictions, SPIN, have been harnessed in this research with two main objectives: knowledge representation in the restaurant domain and reasoning in the same domain.

With regard to knowledge representation, an OWL-based ontological model of the restaurant domain has been proposed. This aims to link and further integrate the vocabularies of the APIs of location-based social networks and user opinion websites, which provide heterogeneous content in the aforementioned domain, using a Linked Data approach.

This model is also used for the modeling of user preferences and context. In this sense, a hybrid technique for contextual information representation and modeling based on generative probabilistic topical models, specifically LDA models, has been proposed. This technique takes advantage of the ontological domain model, as well as of a base of inference rules represented under the SPIN notation, for the inference of high-level temporal and social contextual information from low-level related information. The context model proposed in this research is, in fact, intended to integrate such high-level contextual information with low-level location information, that is, geolocation data obtained from mobile devices, with the primary objective of exploring the importance of the social situations of users by recommending to them food and beverage establishments that best meet their needs in certain contexts.

This technique represents a contribution of this thesis in the field of reasoning. Moreover, an ontology-based semantic similarity metric has been proposed in the same field. This metric uses a hybrid taxonomic/non-taxonomic ontological characteristics approach, and allows capturing, besides taxonomic knowledge, explicit and inferred non-taxonomic knowledge. It is worth nothing that, although this metric does not formally employ a hybrid approach based on graph paths and ontological characteristics sets, it allows capturing the types of semantic knowledge to which these approaches are commonly bounded: taxonomic knowledge and non-taxonomic knowledge, respectively. In conjunction with a technique for propagating preferences through class hierarchies in ontology-based preference profiles, the proposed semantic similarity metric aims to alleviate the problem by which pure collaborative filtering-based recommender systems are likely to be affected: the cold boot in its modalities of new user and new item.

In the scope of the recommendation itself, it should be mentioned that the LDA model underlying the contextual modeling technique is also exploited for the purposes of user profiling (modeling preferences) so that food and beverage establishments are recommended to users based on the semantic similarities between establishments and two different types of preferences: preferences for establishments and preferences for topics in the LDA model. In this sense, a hybrid memory-based and model-based collaborative filtering technique has been proposed under the top-n recommendations approach, where the model-based component is represented by the LDA model of high-level contextual information (generative probabilistic topic model), and the item-based component is represented by the ontology-based semantic similarity metric.

6.5.1.1. Demostration of the Research Thesis

On the one hand, thanks to the analysis of the state of the art it has been possible to answer the questions that aimed to demonstrate the thesis of this research through the **Sub-hypothesis 1**.

According to the results of the state of the art study discussed in section 6.2.2 of this document, there are practically no proposals in the literature on context-aware recommender systems that consists in applying Semantic Web technologies as well as latent class statistical models to the restaurant industry with two main purposes: (1) the definition of hybrid techniques for mining, modeling and representing contextual information and (2) the definition of hybrid recommendation techniques.

This should not be interpreted as an indication that it is not possible to combine these types of computational technologies and techniques for these purposes. In fact, according to the results of the execution of the evaluation method proposed in this research, which are discussed in depth in section 6.4.2 of this document, from a classification task perspective, accuracy of recommendations generated by the context-aware recommendation method for food and beverage establishments that represents the contribution of this thesis is considerably improved by integrating the LDA model of high-level contextual information (formally, the contextual information modeling technique) to the ontology-based semantic similarity metric. This can be specifically observed by comparing the results of the offline experiment with the results of the user study.

Furthermore, as mentioned in section 6.4.2, existing proposals in the application of recommendation techniques and tools to the restaurant industry employ either Semantic Web technologies, commonly the vocabulary definition and description language, OWL, and the SWRL rule language, or latent class statistical models, commonly the generative probabilistic topic model, LDA, for the aforementioned purposes.

On the other hand, with respect to the demonstration of the thesis of the research from the questions associated to **Sub-hypothesis 2**, the study of the state of the art has allowed answering, at least partially, these questions as in the previous case.

In this case, according to the results of this study, the proposals in the research field in context-aware recommender systems that consists in applying, either Semantic Web technologies or latent class statistical models, with the aim of defining more powerful techniques for mining, modeling and representing contextual information, commonly do not contemplate high-level contextual information as part of their context models. Furthermore, they consider different types of low-level information, mainly location information and time information; these contributions do not contemplate, however, social contextual information at any level of abstraction.

In this sense, according to the findings derived from the execution of the evaluation method proposed in this research, high-level temporal information, along with high-level social information, seems to be relevant in providing users with recommendations of food and beverage establishments that best fit the preferences of the user in a socio-temporal context and are more accurate in the traditional sense of information retrieval as well as more realistic from the point of view of the user (exact in the context of an ordering task).

By means of the evaluation method used to validate the proposal of this research, it has been possible to answer the questions intended to demonstrate the veracity of the research thesis from the **Sub-hypothesis 3**.

In detail, according to the results of the evaluation method proposed in this research, in a context of an item classification task, the accuracy of the recommendations generated by the context-aware recommendation method for the restaurant domain that represents the contribution of this research is improved by 6,67% in terms of average recall values and 5.70% in terms of average precision values when compared to the recommendations generated by a baseline recommendation method (syntactic similarity metric). This can be deduced from the results of the offline experiment carried out in this research, which, in contrast to the user study, represents a scenario of sufficiency of ratings.

In the context of an ordering task, the accuracy of the recommendations generated by the proposed recommendation method is improved by 12.16% in terms of cumulative NDPM values when compared to the

recommendations generated by a baseline recommendation method (syntactic similarity metric). This is an indicator that the recommendations generated using recommendation techniques based on Semantic Web technologies and latent class statistical models are more realistic from the point of view of the user than the recommendations generated using state-of-the-art recommendation techniques.

6.5.2. Future Work

As explained throughout the two previous sections, during the execution of the proposed method, or rather the deployment of the prototype of the context-aware recommender system for food and beverage establishments implemented from the proposed software architecture, a phase consisting in collecting user check-ins histories is required. This so-called "training" phase is intended to build the LDA model of high-level contextual information; thus, as long as sufficient historical information to construct a robust LDA model is not collected, all the characteristics of the method that depend on the existence of the LDA model will not be available. This mainly affects the part of the method corresponding to the context-aware recommendation approach.

This is evidently due to the fact that the APIs of two of the location-based services that are attempted to integrate and link to the domain ontology and the corresponding knowledge repository (RDF data) using a Linked Data approach do not make available through their APIs data on the check-ins associated to the establishments in their databases. In this sense, it is essential to define some mechanism based on an alternative external data source (e.g., Facebook) that allows the system not to depend on its usage data to make it possible to build the LDA model of high-level contextual information. For more details about this approach, please refer to the academic products of this research, specifically to the work (Colombo-Mendoza et al., 2017).

As part of the evident future work of this thesis, it is planned to fully implement the prototype of the recommender system implemented as proof of concept from the proposed software architecture, which corresponds mainly to the implementation of the client application as a fully functional cross-platform mobile application. In the medium term, it would be very interesting to make this application available to real users on different digital distribution platforms of commercial mobile applications such as the Google Play Store and the App Store.

With regard to the evaluation of the system, it would be equally relevant to carry out a second evaluation in the field of Computer Science (the evaluation proposed in this research corresponds to this category), in this case under a computational learning approach. This type of evaluation is usually focused on the evaluation of aspects of the technical quality of the systems, which are related to their computational performance. Taking into consideration that the optimal computational performance was not considered as a priority in this thesis, this further evaluation should be aimed at assessing the computational demand of the algorithm that implements the ontology-based semantic similarity metric and the algorithm for generating LDA models for stability analysis, which at first glance did not prove to be a problem for the computational performance of the recommender system, although it is assumed that in certain circumstances they could be.

Referencias

Adomavicius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (2011). Context-Aware Recommender Systems. *AI Magazine*, *32*(3), 67–80.

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Trans. Inf. Syst.*, *23*(1), 103–145. https://doi.org/10.1145/1055709.1055714

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowl. and Data Eng.*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Adomavicius, G., & Tuzhilin, A. (2008). Context-aware Recommender Systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems* (pp. 335–336). New York, NY, USA: ACM. https://doi.org/10.1145/1454008.1454068

Agarwal, V., & Bharadwaj, K. K. (2013). A collaborative filtering framework for friends recommendation in social networks based on interaction intensity and adaptive user similarity. *Social Network Analysis and Mining*, *3*(3), 359–379. https://doi.org/10.1007/s13278-012-0083-7

Aggarwal, C. C. (2016). Neighborhood-Based Collaborative Filtering. In *Recommender Systems* (pp. 29–70). Springer International Publishing. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-29659-3_2

Al-Hassan, M., Lu, H., & Lu, J. (2015). A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system. *Decision Support Systems*, *72*, 97–109. https://doi.org/10.1016/j.dss.2015.02.001

Allahyari, M., & Kochut, K. (2016). Semantic Context-Aware Recommendation via Topic Models Leveraging Linked Open Data. In *Web Information Systems Engineering – WISE 2016* (pp. 263–277). Springer, Cham. https://doi.org/10.1007/978-3-319-48740-3_19

Alor-Hernández, G., Rosales-Morales, V. Y., & Colombo-Mendoza, L. O. (2015). *Frameworks, Methodologies, and Tools for Developing Rich Internet Applications:* IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-6437-1

Amatriain, X., Jaimes*, A., Oliver, N., & Pujol, J. M. (2011). Data Mining Methods for Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 39–71). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_2

Ayala, V. A. A., Przyjaciel-Zablocki, M., Hornung, T., Schätzle, A., & Lausen, G. (2014). Extending SPARQL for Recommendations. In *Proceedings of Semantic Web Information Management on Semantic Web Information Management* (p. 1:1–1:8). New York, NY, USA: ACM. https://doi.org/10.1145/2630602.2630604

Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, Collaborative Recommendation. *Commun. ACM*, *40*(3), 66–72. https://doi.org/10.1145/245108.245124

Bao, J., Zheng, Y., & Mokbel, M. F. (2012). Location-based and Preference-aware Recommendation Using Sparse Geo-social Networking Data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 199–208). New York, NY, USA: ACM. https://doi.org/10.1145/2424321.2424348

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation As Classification: Using Social and Content-based Information in Recommendation. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence* (pp. 714–720). Menlo Park, CA, USA: American Association for Artificial Intelligence. Retrieved from http://dl.acm.org/citation.cfm?id=295240.295795

Bazire, M., & Brézillon, P. (2005). Understanding Context Before Using It. In *Modeling and Using Context* (pp. 29–40). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11508373_3

Bechhofer, S., van Harmelenk, F., Hendler, J., Horrocks, I., McGuinnes, D. L., Patel-Schneider, P. F., & Stein, L. A. (2004). *OWL Web Ontology Language Reference* (W3C Recommendation). World Wide Web Consortium.

Beckett, D., Berners-Lee, T., Prud'hommeaux, E., & Carothers, G. (2014). *RDF 1.1 Turlte*. World Wide Web Consortium.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*, 34–43.

Billsus, D., & Pazzani, M. J. (1998). Learning Collaborative Information Filters. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 46–54). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=645527.657311

Blanco-Fernández, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrer, M., López-Nores, M., García-Duque, J., … Bermejo-Muñoz, J. (2008a). A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*, *21*(4), 305–320. https://doi.org/10.1016/j.knosys.2007.07.004

Blanco-Fernández, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrer, M., López-Nores, M., García-Duque, J., … Bermejo-Muñoz, J. (2008b). An MHP framework to provide intelligent personalized recommendations about digital TV contents. *Software: Practice and Experience*, *38*(9), 925–960. https://doi.org/10.1002/spe.855

Blanco-Fernandez, Y., Pazos-Arias, J. J., Lopez-Nores, M., Gil-Solla, A., & Ramos-Cabrer, M. (2006). AVATAR: an improved solution for personalized TV based on semantic interface. In *2006 Digest of Technical Papers International Conference on Consumer Electronics* (pp. 145–146). https://doi.org/10.1109/ICCE.2006.1598352

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Boley, H., Paschke, A., & Shafiq, O. (2010). RuleML 1.0: The Overarching Specification of Web Rules. In *Semantic Web Rules* (pp. 162–178). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-16289-3_15

Boley, H., Tabet, S., & Wagner, G. (2001). Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In *Proceedings of the First International Conference on Semantic Web Working* (pp. 381–401). Aachen, Germany, Germany: CEUR-WS.org. Retrieved from http://dl.acm.org/citation.cfm?id=2956602.2956628

Borràs, J., Moreno, A., & Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, *41*(16), 7370–7389. https://doi.org/10.1016/j.eswa.2014.06.007

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition)* (W3C Recommendation). World Wide Web Consortium.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 43–52). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2074094.2074100

Brickley, D., Guha, R. V., & McBride, B. (2014). *RDF Schema 1.1* (W3C Recommendation). World Wide Web Consortium.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, *12*(4), 331–370. https://doi.org/10.1023/A:1021240730564

Burke, R. D. (2000). Knowledge-based recommender systems. *Encyclopedia of Library and Information Science*, *69*(Supplement 32).

Burke, R. D., Hammond, K. J., & Young, B. C. (1997). The FindMe Approach to Assisted Browsing. *IEEE Expert: Intelligent Systems and Their Applications*, *12*(4), 32–40. https://doi.org/10.1109/64.608186

Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-performance Recommender Systems. *ACM Trans. Web*, *5*(1), 2:1–2:33. https://doi.org/10.1145/1921591.1921593

Candillier, L., Meyer, F., & Boullé, M. (2007). Comparing State-of-the-Art Collaborative Filtering Systems. In *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition* (pp. 548–562). Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-540-73499-4_41

Cantador, I., Bellogín, A., & Castells, P. (2008). A Multilayer Ontology-based Hybrid Recommendation Model. *AI Commun.*, *21*(2–3), 203–210.

Cantador, I., Castells, P., & Vallet, D. (2006). Enriching Group Profiles with Ontologies for Knowledge-Driven Colalborative Content Retrieval. In *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)* (pp. 358–363). https://doi.org/10.1109/WETICE.2006.36

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications*, *39*(12), 10990–11000. https://doi.org/10.1016/j.eswa.2012.03.025

Celma, Ò. (2010). Evaluation Metrics. In *Music Recommendation and Discovery* (pp. 109–128). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-13287-2_5

Chen, H., Finin, T., & Joshi, A. (2003). An Ontology for Context-aware Pervasive Computing Environments. *Knowl. Eng. Rev.*, *18*(3), 197–207. https://doi.org/DOI:10.1017/S0269888904000025

Chen, L.-C., Kuo, P.-J., & Liao, I.-E. (2015). Ontology-based library recommender system using MapReduce. *Cluster Computing*, *18*(1), 113–121. https://doi.org/10.1007/s10586-013-0342-z

Chen, R.-C., Huang, Y.-H., Bau, C.-T., & Chen, S.-M. (2012). A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection. *Expert Systems with Applications*, *39*(4), 3995–4006. https://doi.org/10.1016/j.eswa.2011.09.061

Chen, Y., & George, E. I. (1999). A Bayesian Model for Collaborative Filtering. In *Online Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*. San Francisco, CA.

Christensen, I., Schiaffino, S., & Armentano, M. (2016). Social group recommendation in the tourism domain. *Journal of Intelligent Information Systems*, *47*(2), 209–231. https://doi.org/10.1007/s10844-016-0400-0

Colombo-Mendoza, L. O., Alor-Hernández, G., Rodríguez-gonzález, A., & Valencia-garcía, R. (2014). MobiCloUP!: a PaaS for cloud services-based mobile applications. *Automated Software Engineering*, *21*(3), 391–437. https://doi.org/10.1007/s10515-014-0143-5

Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Colomo-Palacios, R., & Alor-Hernández, G. (2017). Towards a knowledge-based probabilistic and context-aware social recommender system. *Journal of Information Science*, 165551517698787. https://doi.org/10.1177/0165551517698787

Cramer, H., Evers, V., Ramlal, S., Someren, M. van, Rutledge, L., Stash, N., … Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, *18*(5), 455. https://doi.org/10.1007/s11257-008-9051-3

Cyganiak, R., Wood, D., Lanthaler, M., Klyne, G., Carrol, J. J., & McBride, B. (2014). *RDF 1.1 Concepts and Abstract Syntax* (W3C Recommendation). World Wide Web Consortium.

Davis, M., & Whistler, K. (2016). *UNICODE NORMALIZATION FORMS* (No. Unicode Standard Annex #15). Unicode Consortium.

de Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks.

International Journal of Approximate Reasoning, *51*(7), 785–799. https://doi.org/10.1016/j.ijar.2010.04.001

Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 107–144). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_4

Dey, A. K. (2001). Understanding and Using Context. *Personal Ubiquitous Comput.*, *5*(1), 4–7. https://doi.org/10.1007/s007790170019

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012). Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems* (pp. 1–8). New York, NY, USA: ACM. https://doi.org/10.1145/2362499.2362501

Dourish, P. (2004). What We Talk About when We Talk About Context. *Personal Ubiquitous Comput.*, *8*(1), 19–30. https://doi.org/10.1007/s00779-003-0253-8

Duerst, M., & Suignard, M. (2005). *Internationalized Resource Identifiers (IRIs)* (No. RFC 3987). Internnet Engineering Task Force.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative Filtering Recommender Systems. *Found. Trends Hum.-Comput. Interact.*, *4*(2), 81–173. https://doi.org/10.1561/1100000009

Fallside, D. C., & Walmsley, P. (2004). *XML Schema Part 0: Primer Second Edition* (W3C Recommendation). World Wide Web Consortium.

Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., & Reiterer, S. (2013). Toward the Next Generation of Recommender Systems: Applications and Research Challenges, 81–98. https://doi.org/10.1007/978-3-319-00372-6_5

Ferguson, G. A. (1959). *Statistical analysis in psychology and education* (Vol. vii). New York, NY, US: McGraw-Hill.

Figueroa, C., Vagliano, I., Rocha, O. R., & Morisio, M. (2015). A systematic literature review of Linked Data-based recommender systems. *Concurrency and Computation: Practice and Experience*, *27*(17), 4659–4684. https://doi.org/10.1002/cpe.3449

Fling, B. (2009). *Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps - Animal Guide* (1st ed.). O'Reilly Media, Inc.

Gandon, F., & Schreiber, G. (2014). *RDF 1.1 XML Syntax*. World Wide Web Consortium.

Gao, H., Chen, D.-B., Wang, G.-N., Mensah, D. N. A., & Fu, Y. (2014). A continuous rating model for news recommendation. *Journal of Information Science*, *40*(5), 568–577. https://doi.org/10.1177/0165551514542065

Gearon, P., Passant, A., & Polleres, A. (2013). *SPARQL 1.1 Update* (W3C Recommendation). World Wide Web Consortium.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM*, *35*(12), 61–70. https://doi.org/10.1145/138859.138867

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Inf. Retr.*, *4*(2), 133–151. https://doi.org/10.1023/A:1011419012209

Greene, D., O'Callaghan, D., & Cunningham, P. (2014). How Many Topics? Stability Analysis for Topic Models. In T. Calders, F. Esposito, E. Hüllermeier, & R. Meo (Eds.), *Machine Learning and Knowledge Discovery in Databases* (Vol. 8724, pp. 498–513). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-662-44848-9_32

Gu, W., Dong, S., & Chen, M. (2016). Personalized news recommendation based on articles chain building. *Neural Computing and Applications*, *27*(5), 1263–1272. https://doi.org/10.1007/s00521-015-1932-x

Hadj Taieb, M. A., Ben Aouicha, M., & Ben Hamadou, A. (2014). Ontology-based approach for measuring semantic similarity. *Engineering Applications of Artificial Intelligence*, *36*, 238–261. https://doi.org/10.1016/j.engappai.2014.07.015

Hariri, N., Mobasher, B., & Burke, R. (2012). Context-aware Music Recommendation Based on Latenttopic Sequential Patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems* (pp. 131–138). New York, NY, USA: ACM. https://doi.org/10.1145/2365952.2365979

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2013). Semantic Measures Based on RDF Projections: Application to Content-Based Recommendation Systems (pp. 606–615). Presented at the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems," Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41030-7_44

Harris, S., Seaborne, A., & Prud'hommeaux, E. (2013). *SPARQL 1.1 Query Language* (W3C Recommendation). World Wide Web Consortium.

Hawalah, A., & Fasli, M. (2014). Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications*, *41*(10), 4777–4797. https://doi.org/10.1016/j.eswa.2014.01.039

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

Hoekstra, R. (2009). *Ontology Representation: Design Patterns and Ontologies that Make Sense - Volume 197 Frontiers in Artificial Intelligence and Applications ... in Artificial Intelligence and Applications)*. Amsterdam ; Fairfax, VA: IOS Press.

Hofmann, T. (2004). Latent Semantic Models for Collaborative Filtering. *ACM Trans. Inf. Syst.*, *22*(1), 89–115. https://doi.org/10.1145/963770.963774

Hofmann, T., & Puzicha, J. (1999). Latent Class Models for Collaborative Filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2* (pp. 688–693). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1624312.1624317

Hong, J., Suh, E.-H., Kim, J., & Kim, S. (2009). Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, *36*(4), 7448–7457. https://doi.org/10.1016/j.eswa.2008.09.002

Horrocks, I., Glimm, B., & Sattler, U. (2007). Hybrid Logics and Ontology Languages. *Electronic Notes in Theoretical Computer Science*, *174*(6), 3–14. https://doi.org/10.1016/j.entcs.2006.11.022

Horrocks, I., & Patel-Schneider, P. F. (2004). A Proposal for an Owl Rules Language. In *Proceedings of the 13th International Conference on World Wide Web* (pp. 723–731). New York, NY, USA: ACM. https://doi.org/10.1145/988672.988771

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* (W3C Member Submission). World Wide Web Consortium.

Huang, S. (2011). Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications*, *10*(4), 398–407. https://doi.org/10.1016/j.elerap.2010.11.003

Jannach, D., & Friedrich, G. (2011, July). *Tutorial: Recommender Systems*. Presented at the International Joint Conference on Artificial Intelligence, Barcelona, Spain.

Jannach, D., Zanker, M., Ge, M., & Gröning, M. (2012). Recommender Systems in Computer Science and Information Systems – A Landscape of Research. In C. Huemer & P. Lops (Eds.), *E-Commerce and Web Technologies* (pp. 76–87). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-32273-0_7

Jiang, S., Qian, X., Mei, T., & Fu, Y. (2016). Personalized Travel Sequence Recommendation on Multi-Source Big Social Media. *IEEE Transactions on Big Data*, *2*(1), 43–56. https://doi.org/10.1109/TBDATA.2016.2541160

Jing, L., Wang, P., & Yang, L. (2015). Sparse Probabilistic Matrix Factorization by Llaplace Distribution for Collaborative Filtering. In *Proceedings of the 24th International Conference on Artificial Intelligence* (pp. 1771–1777). Buenos Aires, Argentina: AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=2832415.2832495

Karpus, A., Vagliano, I., Goczyła, K., & Morisio, M. (2016). An Ontology-based contextual pre-filtering technique for Recommender Systems. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 411–420).

Karypis, G. (2001). Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (pp. 247–254). New York, NY, USA: ACM. https://doi.org/10.1145/502585.502627

Kitchenham, B., Linkman, S., & Law, D. (1997). DESMET: a methodology for evaluating software engineering methods and tools. *Computing & Control Engineering Journal*, *8*(3), 120–126. https://doi.org/10.1049/cce:19970304

Knublauch, H. (2011a). *SPIN - Modeling Vocabulary* (W3C Member Submission). World Wide Web Consortium.

Knublauch, H. (2011b). *SPIN - SPARQL Syntax* (W3C Member Submission). World Wide Web Consortium.

Knublauch, H., Hendler, J. A., & Idehen, K. (2011). *SPIN - Overview and Motivation* (W3C Member Submission). World Wide Web Consortium.

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Commun. ACM*, *40*(3), 77–87. https://doi.org/10.1145/245108.245126

Krzywicki, A., Wobcke, W., Kim, Y. S., Cai, X., Bain, M., Mahidadia, A., & Compton, P. (2015). Collaborative Filtering for People-to-people Recommendation in Online Dating. *Int. J. Hum.-Comput. Stud.*, *76*(C), 50–66. https://doi.org/10.1016/j.ijhcs.2014.12.003

Lai, C.-H., Liu, D.-R., & Liu, M.-L. (2015). Recommendations based on personalized tendency for different aspects of influences in social media. *Journal of Information Science*, *41*(6), 814–829. https://doi.org/10.1177/0165551515603324

Li, Y.-M., Wu, C.-T., & Lai, C.-Y. (2013). A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship. *Decision Support Systems*, *55*(3), 740–752. https://doi.org/10.1016/j.dss.2013.02.009

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*. Retrieved from http://psycnet.apa.org/psycinfo/1933-01885-001

Lin, C., Xie, R., Guan, X., Li, L., & Li, T. (2014). Personalized news recommendation via implicit social experts. *Information Sciences*, *254*, 1–18. https://doi.org/10.1016/j.ins.2013.08.034

Liu, B., & Xiong, H. (2013). Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (Vols. 1–0, pp. 396–404). Society for Industrial and Applied Mathematics. Retrieved from http://epubs.siam.org/doi/abs/10.1137/1.9781611972832.44

Mao, J., Lu, K., Li, G., & Yi, M. (2016). Profiling users with tag networks in diffusion-based personalized recommendation. *Journal of Information Science*, *42*(5), 711–722. https://doi.org/10.1177/0165551515603321

Marlin, B. (2004). Modeling User Rating Profiles For Collaborative Filtering. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.

Martín-Vicente, M. I., Gil-Solla, A., Ramos-Cabrer, M., Pazos-Arias, J. J., Blanco-Fernández, Y., & López-Nores, M. (2014). A semantic approach to improve neighborhood formation in collaborative recommender systems. *Expert Systems with Applications*, *41*(17), 7776–7788. https://doi.org/10.1016/j.eswa.2014.06.038

Massa, P., & Bhattacharjee, B. (2004). Using Trust in Recommender Systems: An Experimental Analysis. In *Trust Management* (pp. 221–235). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24747-0_17

Mettouris, C., & Papadopoulos, G. A. (2013). Contextual Modelling in Context-Aware Recommender Systems: A Generic Approach. In *Web Information Systems Engineering – WISE 2011 and 2012 Workshops* (pp. 41–52). Springer, Berlin, Heidelberg. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-38333-5_6

Meymandpour, R., & Davis, J. G. (2016). A semantic similarity measure for linked data: An information content-based approach. *Knowledge-Based Systems*, *109*, 276–293. https://doi.org/10.1016/j.knosys.2016.07.012

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological User Profiling in Recommender Systems. *ACM Trans. Inf. Syst.*, *22*(1), 54–88. https://doi.org/10.1145/963770.963773

Miyahara, K., & Pazzani, M. J. (2000). Collaborative Filtering with the Simple Bayesian Classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence* (pp. 679–689). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1764967.1765055

Mobasher, B. (2014, August). *Context Aware Recommendation*. Presented at the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, US.

Mobasher, B., Jin, X., & Zhou, Y. (2004). Semantically Enhanced Collaborative Filtering on the Web, 57–76. https://doi.org/10.1007/978-3-540-30123-3_4

Mokbel, M. F., & Levandoski, J. J. (2009). Toward Context and Preference-aware Location-based Services. In *Proceedings of the Eighth ACM International Workshop on Data Engineering for Wireless and Mobile Access* (pp. 25–32). New York, NY, USA: ACM. https://doi.org/10.1145/1594139.1594150

Montaner, M., López, B., & de la Rosa, J. L. (2002). Developing Trust in Recommender Agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1* (pp. 304–305). New York, NY, USA: ACM. https://doi.org/10.1145/544741.544811

Moreno, A., Valls, A., Isern, D., Marin, L., & Borràs, J. (2013). SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities. *Engineering Applications of Artificial Intelligence*, *26*(1), 633–651. https://doi.org/10.1016/j.engappai.2012.02.014

Movahedian, H., & Khayyambashi, M. R. (2014). Folksonomy-based user interest and disinterest profiling for improved recommendations: An ontological approach. *Journal of Information Science*, *40*(5), 594–610. https://doi.org/10.1177/0165551514539870

Nakatsuji, M., & Fujiwara, Y. (2014). Linked taxonomies to capture users' subjective assessments of items to facilitate accurate collaborative filtering. *Artificial Intelligence*, *207*, 52–68. https://doi.org/10.1016/j.artint.2013.11.003

Nilashi, M., Ibrahim, O. bin, & Ithnin, N. (2014). Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system. *Knowledge-Based Systems*, *60*, 82–101. https://doi.org/10.1016/j.knosys.2014.01.006

O'Donovan, J., & Smyth, B. (2005). Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (pp. 167–174). New York, NY, USA: ACM. https://doi.org/10.1145/1040830.1040870

Peska, L., & Vojtas, P. (2013). Enhancing Recommender System with Linked Open Data (pp. 483–494). Presented at the International Conference on Flexible Query Answering Systems, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40769-7_42

Phillips, A., & Davis, M. (2009). *Tags for Identifying Languages* (No. RFC 5646). Internnet Engineering Task Force.

Poitrenaud, S. (2001). *COMPLEXITÉ COGNITIVE DES INTERACTIONS HOMME-MACHINE*. Éditions L'Harmattan. Retrieved from https://www.leslibraires.fr/livre/4002940-complexite-cognitive-des-interactions-homme-mac--sebastien-poitrenaud-editions-l-harmattan

Pyo, S., Kim, E., & kim, M. (2015). LDA-Based Unified Topic Modeling for Similar TV User Grouping and TV Program Recommendation. *IEEE Transactions on Cybernetics*, *45*(8), 1476–1490. https://doi.org/10.1109/TCYB.2014.2353577

Ranganathan, A., & Campbell, R. H. (2003). A Middleware for Context-aware Agents in Ubiquitous Computing Environments. In *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware* (pp. 143–161). New York, NY, USA: Springer-Verlag New York, Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1515915.1515926

Ren, X., Song, M., E, H., & Song, J. (2017). Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation. *Neurocomputing*, *241*, 38–55. https://doi.org/10.1016/j.neucom.2017.02.005

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (pp. 175–186). New York, NY, USA: ACM. https://doi.org/10.1145/192844.192905

Resnick, P., & Varian, H. R. (1997). Recommender Systems. *Commun. ACM*, *40*(3), 56–58. https://doi.org/10.1145/245108.245121

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_1

Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., … Hyvönen, E. (2013). SMARTMUSEUM: A mobile recommender system for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, *20*, 50–67. https://doi.org/10.1016/j.websem.2013.03.001

Salamó, M., McCarthy, K., & Smyth, B. (2012). Generating recommendations for consensus negotiation in group personalization services. *Personal and Ubiquitous Computing*, *16*(5), 597–610. https://doi.org/10.1007/s00779-011-0413-1

Salton, G., & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.

Sánchez, D., Batet, M., Isern, D., & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, *39*(9), 7718–7728. https://doi.org/10.1016/j.eswa.2012.01.082

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of Recommendation Algorithms for e-Commerce. In *Proceedings of the 2Nd ACM Conference on Electronic Commerce* (pp. 158–167). New York, NY, USA: ACM. https://doi.org/10.1145/352871.352887

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). {Application of Dimensionality Reduction in Recommender System -- A Case Study}. Presented at the Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Web Mining for E--Commerce -- Challenges and Opportunities (WEBKDD'00).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 285–295). New York, NY, USA: ACM. https://doi.org/10.1145/371920.372071

Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In *Proceedings*

of the 1998 ACM Conference on Computer Supported Cooperative Work (pp. 345–354). New York, NY, USA: ACM. https://doi.org/10.1145/289444.289509

Scheel, C., Castellanos, A., Lee, T., & Luca, E. W. D. (2012). The Reason Why: A Survey of Explanations for Recommender Systems. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation* (pp. 67–84). Springer, Cham. https://doi.org/10.1007/978-3-319-12093-5_3

Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications* (pp. 85–90). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/WMCSA.1994.16

Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 257–297). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_8

Sharma, R., & Ray, S. (2016). Explanations in Recommender Systems: An Overview. *Int. J. Bus. Inf. Syst.*, *23*(2), 248–262. https://doi.org/10.1504/IJBIS.2016.078909

Shi, B., Ifrim, G., & Hurley, N. (2016). Learning-to-Rank for Real-Time High-Precision Hashtag Recommendation for Streaming News. In *Proceedings of the 25th International Conference on World Wide Web* (pp. 1191–1202). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2872427.2882982

Sporny, M., Longly, D., Kellogg, G., Lanthaler, M., & Lindstrom, N. (2014). *JSON-LD 1.0*. World Wide Web Consortium.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, *25*(1), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

Su, X., & Khoshgoftaar, T. (2006). Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms (pp. 497–504). IEEE. https://doi.org/10.1109/ICTAI.2006.41

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, *2009*, e421425. https://doi.org/10.1155/2009/421425

Tintarev, N., & Masthoff, J. (2011). Designing and Evaluating Explanations for Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 479–510). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_15

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352. https://doi.org/10.1037/0033-295X.84.4.327

Vallet, D., Castells, P., Fernandez, M., Mylonas, P., & Avrithis, Y. (2007). Personalized Content Retrieval in Context Using Ontological Knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, *17*(3), 336–346. https://doi.org/10.1109/TCSVT.2007.890633

Vesin, B., Ivanović, M., Klašnja-Milićević, A., & Budimac, Z. (2012). Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. *Expert Systems with Applications*, *39*(15), 12229–12246. https://doi.org/10.1016/j.eswa.2012.04.052

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking LDA: Why Priors Matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 22* (pp. 1973–1981). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.pdf

Wang, X. H., Zhang, D. Q., Gu, T., & Pung, H. K. (2004). Ontology Based Context Modeling and Reasoning Using OWL. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops* (p. 18–). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=977405.978618

Xu, Z., Chen, L., & Chen, G. (2015). Topic based context-aware travel recommendation method exploiting geotagged photos. *Neurocomputing*, *155*, 99–107. https://doi.org/10.1016/j.neucom.2014.12.043

Yao, Y. Y. (1995). Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*, *46*(2), 133–145. https://doi.org/10.1002/(SICI)1097-4571(199503)46:2<133::AID-ASI6>3.0.CO;2-Z

Yin, H., Cui, B., Chen, L., Hu, Z., & Zhang, C. (2015). Modeling Location-Based User Rating Profiles for Personalized Recommendation. *ACM Trans. Knowl. Discov. Data*, *9*(3), 19:1–19:41. https://doi.org/10.1145/2663356

Yılmaz, Ö., & Erdur, R. C. (2012). iConAwa – An intelligent context-aware system. *Expert Systems with Applications*, *39*(3), 2907–2918. https://doi.org/10.1016/j.eswa.2011.08.152

Yu, K., Zhang, B., Zhu, H., Cao, H., & Tian, J. (2012). Towards Personalized Context-Aware Recommendation by Mining Context Logs through Topic Models. In P.-N. Tan, S. Chawla, C. K. Ho, & J. Bailey (Eds.), *Advances in Knowledge Discovery and Data Mining* (Vol. 7301, pp. 431–443). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-30217-6_36

Yu, Z., Nakamura, Y., Jang, S., Kajita, S., & Mase, K. (2007). Ontology-based Semantic Recommendation for Context-aware e-Learning. In *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing* (pp. 898–907). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=2391319.2391414

Zhao, F., Zhu, Y., Jin, H., & Yang, L. T. (2016). A personalized hashtag recommendation approach using LDA-based topic model in microblog environment. *Future Generation Computer Systems*, *65*, 196–206. https://doi.org/10.1016/j.future.2015.10.012