Recomendação de conteúdo baseada em
informações semânticas extraídas de bases de
conhecimento

Salmo Marques da Silva Júnior

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Salmo Marques da Silva Júnior

Recomendação de conteúdo baseada em informações semânticas extraídas de bases de conhecimento

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Marcelo Garcia Manzato

USP – São Carlos
Julho de 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

J586r Júnior, Salmo Marques da Silva
Recomendação de conteúdo baseada em informações
semânticas extraídas de bases de conhecimento /
Salmo Marques da Silva Júnior; orientador Marcelo
Garcia Manzato. - São Carlos - SP, 2017.
102 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática Computacional)
- Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Filtragem colaborativa. 2. Modelos de
vizinhança. 3. Cenário de partida fria. 4. Distância
semântica. 5. Similaridade semântica. I. Manzato,
Marcelo Garcia, orient. II. Título.

Salmo Marques da Silva Júnior

**Content recommendation based on semantic information
extracted from knowledge bases**

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Marcelo Garcia Manzato

USP – São Carlos
July 2017

*Este trabalho é dedicado a todas as pessoas que direta
ou indiretamente contribuíram na minha formação pessoal, acadêmica e profissional.
Em especial, a minha família e aos professores do Instituto de Ciências Matemáticas e de
Computação (ICMC).*

AGRADECIMENTOS

A conclusão de meu trabalho de mestrado e confecção desta dissertação foram muito desejadas e são de grande significância na minha vida. Portanto, agradeço a todos que de algum modo contribuíram, direta ou indiretamente, na concretização deste trabalho.

Agradeço à Universidade de São Paulo e o Instituto de Ciências Matemáticas e de Computação pela oportunidade de desenvolvimento deste trabalho e pelo conhecimento transmitido ao longo do curso.

Ao Prof. Dr. Marcelo Garcia Manzato, agradeço pela solicitude apresentada a todo momento, e auxílio prestado ao orientar meu trabalho.

À minha esposa, Alessandra Ribeiro da Silva, e ao meu filho, Victor Hugo Ribeiro da Silva, agradeço o companheirismo e compreensão necessários ao longo deste trabalho.

Dentre os amigos, agradeço especialmente os colegas de trabalho no Senac São Carlos, que muito contribuíram para produção deste trabalho, oferecendo as condições necessárias para minha dedicação.

Por fim, agradeço a oportunidade recebida e o apoio despendido por todos no decorrer deste gratificante trabalho.

*“Quer você acredite que consiga fazer uma coisa ou não,
você está certo.”
(Henry Ford)*

RESUMO

SILVA JÚNIOR, S.M. **Recomendação de conteúdo baseada em informações semânticas extraídas de bases de conhecimento**. 2017. 102 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

A fim de auxiliar usuários durante o consumo de produtos, sistemas Web passaram a incorporar módulos de recomendação de itens. As abordagens mais populares são a baseada em conteúdo, que recomenda itens a partir de características que são do seu interesse, e a filtragem colaborativa, que recomenda itens bem avaliados por usuários com perfis semelhantes ao do usuário alvo, ou que são semelhantes aos que foram bem avaliados pelo usuário alvo. Enquanto que a primeira abordagem apresenta limitações como a sobre-especialização e a análise limitada de conteúdo, a segunda enfrenta problemas como o novo usuário e/ou novo item, também conhecido como partida fria. Apesar da variedade de técnicas disponíveis, um problema comum existente na maioria das abordagens é a falta de informações semânticas para representar os itens do acervo. Trabalhos recentes na área de Sistemas de Recomendação têm estudado a possibilidade de usar bases de conhecimento da Web como fonte de informações semânticas. Contudo, ainda é necessário investigar como usufruir de tais informações e integrá-las de modo eficiente em sistemas de recomendação. Dessa maneira, este trabalho tem o objetivo de investigar como informações semânticas provenientes de bases de conhecimento podem beneficiar sistemas de recomendação por meio da descrição semântica de itens, e como o cálculo da similaridade semântica pode amenizar o desafio enfrentado no cenário de partida fria. Como resultado, obtém-se uma técnica que pode gerar recomendações adequadas ao perfil dos usuários, incluindo itens novos do acervo que sejam relevantes. Pode-se observar uma melhora de até 10% no RMSE, no cenário de partida fria, quando se compara o sistema proposto com o sistema cuja predição de notas é baseada na correlação de notas.

Palavras-chave: Filtragem colaborativa, Modelos de vizinhança, Cenário de partida fria, Distância semântica, Similaridade semântica.

ABSTRACT

SILVA JÚNIOR, S.M. **Content recommendation based on semantic information extracted from knowledge bases**. 2017. 102 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

In order to support users during the consumption of products, Web systems have incorporated recommendation techniques. The most popular approaches are content-based, which recommends items based on interesting features to the user, and collaborative filtering, which recommends items that were well evaluated by users with similar preferences to the target user, or that have similar features to items which were positively evaluated. While the first approach has limitations such as overspecialization and limited content analysis, the second technique has problems such as the new user and the new item, limitation also known as cold start. In spite of the variety of techniques available, a common problem is the lack of semantic information to represent item's features. Recent works in the field of recommender systems have been studying the possibility to use knowledge databases from the Web as a source of semantic information. However, it is still necessary to investigate how to use and integrate such semantic information in recommender systems. In this way, this work has the proposal to investigate how semantic information gathered from knowledge databases can help recommender systems by semantically describing items, and how semantic similarity can overcome the challenge confronted in the cold-start scenario. As a result, we obtained a technique that can produce recommendations suited to users' profiles, including relevant new items available in the database. It can be observed an improvement of up to 10% in the RMSE in the cold start scenario when comparing the proposed system with the system whose rating prediction is based on the correlation of rates.

Keywords: Content-based filtering, Neighborhood models, Cold start scenario, Semantic distance, Semantic similarity.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura em Alto Nível de um Sistema de Recomendação Baseado em Conteúdo, traduzido de Lops, Gemmis e Semeraro (2011)	31
Figura 2 – Ilustração de um grafo representando as declarações RDF, no qual os sujeitos e objetos são vértices e os predicados são as arestas.	43
Figura 3 – Ilustração de um grafo representando as declarações RDF, identificando sujeitos, predicados e objetos por IRIs.	43
Figura 4 – Relacionamentos entre as subclasses OWL e RDF/RDFS. Extraída de Antoniou e vanHarmelen (2004).	47
Figura 5 – Consulta SPARQL para obter lugares com nomes em homenagem a George Washington. Extraído de Hebelier <i>et al.</i> (2009)	49
Figura 6 – Fragmento da estrutura de relação é-um da WordNet. Adaptada de Meng, Huang e Gu (2013)	51
Figura 7 – Ilustração de um grafo representando os recursos vinculados (<i>Linked Data</i>), onde $a:r_1$, $a:r_2$, $a:r_3$, $a:r_4$ e $a:r_5$ são recursos (vértices do grafo) e $a:l_1$ e $a:l_2$ são <i>links</i> (arestas do grafo)	54
Figura 8 – Visão geral da arquitetura do sistema de recomendação proposto.	62
Figura 9 – Demonstração gráfica da estrutura hierárquica das categorias e da relação de categorias associadas a dois itens.	64
Figura 10 – O ancestral comum “ <i>Space_in_fiction</i> ” das categorias “ <i>Space_adventure_films</i> ” e “ <i>Extraterrestrial_life_in_popular_culture</i> ” revela um <i>link</i> entre o filme <i>Alien</i> e o episódio <i>Lost in Space</i>	66
Figura 11 – Visão geral da arquitetura implementada nos experimentos do SR proposto.	71
Figura 12 – Representação de parte dos dados da base MovieLens e da base de conhecimento local projetada para o sistema proposto.	74
Figura 13 – Representação de parte da hierarquia de categorias semânticas, após a adição da categoria <i>Category:Thing</i>	76
Figura 14 – LDSDiExt vs. baselines	79
Figura 15 – Sim. LDSDiExt vs. baselines	81
Figura 16 – Sim. LDSDiExt vs. baselines	83
Figura 17 – ItemKNN_c vs. ItemKNN_g e ItemKNN_n	85

LISTA DE TABELAS

Tabela 1	–	LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário normal. . . .	80
Tabela 2	–	LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário de partida fria.	80
Tabela 3	–	Sim. LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário normal.	82
Tabela 4	–	Sim. LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário de partida fria.	82
Tabela 5	–	Sim. LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário normal, comparando o desempenho do SR Sim. LDSDiExt usando o conjunto de vizinhos $S_i^k(u)$ com os SRs baselines usando o conjunto de vizinhos $R_i^k(u)$. .	84
Tabela 6	–	Sim. LDSDiExt vs. baselines - Valores de RMSE obtidos no cenário de partida fria, comparando o desempenho do SR Sim. LDSDiExt usando o conjunto de vizinhos $S_i^k(u)$ com os SRs baselines usando o conjunto de vizinhos $R_i^k(u)$	84
Tabela 7	–	ItemKNN_c vs. ItemKNN_g e ItemKNN_n - Valores de RMSE obtidos na comparação do SR ItemKNN_c com os SRs ItemKNN_g e ItemKNN_n no cenário normal.	86
Tabela 8	–	ItemKNN_c. vs. ItemKNN_g e ItemKNN_n - Valores de RMSE obtidos na comparação do SR ItemKNN_c com os SRs ItemKNN_g e ItemKNN_n no cenário de partida fria.	86

LISTA DE ABREVIATURAS E SIGLAS

k NN	k Vizinhos mais Próximos (do Inglês, <i>k-Nearest-Neighbors</i>)
API	Interface de Programação de Aplicativos (do Inglês, <i>Application Programming Interface</i>)
DEA	Distância de Edição de Árvores
IRI	Identificador de Recursos Internacionalizado (do Inglês, <i>Internationalized Resource Identifiers</i>)
ItemKNN_c	Algoritmo ItemKNN com similaridade baseada na correlação de categorias semânticas diretas
ItemKNN_g	Algoritmo ItemKNN com similaridade baseada na correlação de gêneros
ItemKNN_n	Algoritmo ItemKNN com similaridade baseada na correlação de notas
L&C	Leacock & Chodorow - Cálculo de Similaridade Baseado na WordNet
LDSD	Distância Semântica em <i>Linked Data</i> (do Inglês, <i>Linked Data Semantic Distance</i>)
LDSDd	Distância Semântica Direta em <i>Linked Data</i> (do Inglês, <i>Linked Data Direct Semantic Distance</i>)
LDSDi	Distância Semântica Indireta em <i>Linked Data</i> (do Inglês, <i>Linked Data Indirect Semantic Distance</i>)
LDSDiExt	Distância Semântica Indireta em <i>Linked Data</i> (Extensão) (do Inglês, <i>Linked Data Indirect Semantic Distance (Extension)</i>)
LOD	Dados Abertos Conectados (do Inglês, <i>Linked Open Data</i>)
MAE	Erro Médio Absoluto (do Inglês, <i>Mean Absolute Error</i>)
NPL	Processamento de Linguagem Natural (do Inglês, <i>Natural Language Processing</i>)
OWL	Linguagem de Ontologia na Web (do Inglês, <i>Web Ontology Language</i>)
OWL 2 EL	Linguagem Existencial (do Inglês, <i>Existential Language</i>)
OWL 2 QL	Linguagem de Consulta (do Inglês, <i>Query Language</i>)
OWL 2 RL	Regras de Linguagem (do Inglês, <i>Rule Language</i>)
RDF	<i>Framework</i> de Descrição de Recursos (do Inglês, <i>Resource Description Framework</i>)
RDQL	Linguagem para Consulta de Dados em RDF (do Inglês, <i>RDF Data Query Language</i>)
RMSE	Raiz do Erro Quadrático Médio (do Inglês, <i>Root Mean Square Error</i>)
SeRQL	Linguagem Sesame para Consulta RDF (do Inglês, <i>Sesame RDF Query Language</i>)
SGBD	Sistema de Gerenciamento de Banco de Dados

Sim LDSDiExt	Similaridade a partir da Distância Semântica Indireta em <i>Linked Data</i> (Extensão)
SKOS	Sistema de Organização do Conhecimento Simples (do Inglês, <i>Simple Knowledge Organization System</i>)
SPARQL	Protocolo SPARQL e Linguagem de Consulta RDF (do Inglês, <i>SPARQL Protocol and RDF Query Language</i>)
SR	Sistemas de Recomendação
SVD++	Decomposição em Valores Singulares (do Inglês, <i>Singular Value Decomposition++</i>)
TF-IDF	Frequência do Termo–Inverso da Frequência nos Documentos (do Inglês, <i>Term Frequency–Inverse Document Frequency</i>)
URL	Localizador Padrão de Recursos (do Inglês, <i>Uniform Resource Locator</i>)
W&P	Wu & Palmer - Cálculo de Similaridade Baseado na WordNet
WGS84	Sistema Geodésico Mundial 1984 (do Inglês, <i>World Geodetic System 1984</i>)
XML	Linguagem Extensível de Marcação (do Inglês, <i>eXtensible Markup Language</i>)

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	25
1.2	Objetivo	26
1.3	Estrutura do Trabalho	26
2	SISTEMAS DE RECOMENDAÇÃO	29
2.1	Abordagens de Recomendação	30
2.1.1	<i>Filtragem Baseada em Conteúdo</i>	30
2.1.2	<i>Filtragem Colaborativa</i>	33
2.1.2.1	<i>Métodos Baseados em Vizinhança</i>	34
2.1.2.2	<i>Fatoração de Matrizes</i>	35
2.1.3	<i>Sistemas Híbridos</i>	36
2.1.4	<i>Outras Abordagens Relevantes</i>	36
2.2	Avaliação de Sistemas de Recomendação	37
2.2.1	<i>Avaliação da Predição</i>	38
2.2.2	<i>Validação Cruzada em k partições</i>	39
2.2.3	<i>Significância Estatística</i>	39
2.3	Considerações Finais	40
3	DADOS ABERTOS CONECTADOS	41
3.1	Representação de Dados na Web Semântica	42
3.1.1	<i>RDF</i>	42
3.1.2	<i>RDF Schema</i>	44
3.2	Ontologia	45
3.2.1	<i>OWL 2</i>	46
3.2.2	<i>Bases de Conhecimento</i>	48
3.3	Similaridade Semântica	49
3.3.1	<i>Métrica Baseada no Menor Caminho</i>	50
3.3.2	<i>Métrica Leakcock & Chodorow</i>	51
3.3.3	<i>Métrica Wu & Palmer</i>	52
3.3.4	<i>Distância Semântica em Dados Conectados</i>	52
3.4	Trabalhos relacionados	55
3.5	Considerações finais	58

4	PROPOSTA DE SOLUÇÃO	61
4.1	Arquitetura	61
4.1.1	<i>Base de Dados</i>	62
4.1.2	<i>Base de Conhecimento</i>	63
4.1.3	<i>Cálculo da Distância Semântica</i>	63
4.1.4	<i>Cálculo da Similaridade Semântica</i>	67
4.1.5	<i>Sistemas de Recomendação Baseados em Filtragem Colaborativa</i> .	67
4.2	Considerações Finais	69
5	AVALIAÇÃO DO SISTEMA	71
5.1	Metodologia	71
5.1.1	<i>Base de Dados</i>	72
5.1.2	<i>Base de Conhecimento</i>	72
5.1.3	<i>Cálculo da Similaridade Semântica</i>	75
5.1.4	<i>Sistemas Implementados</i>	76
5.2	Método de Avaliação	77
5.3	Resultados	78
5.3.1	<i>SR LDSDiExt vs. SRs baselines</i>	78
5.3.2	<i>SR Sim. LDSDiExt vs. SRs baselines</i>	80
5.3.3	<i>SR Sim. LDSDiExt usando o conjunto $S_i^k(u)$ vs. SRs baselines</i> <i>usando o conjunto $R_i^k(u)$</i>	82
5.3.4	<i>SR ItemKNN_g vs. SR ItemKNN_c</i>	84
5.4	Condiderações Finais	86
6	CONCLUSÕES E TRABALHOS FUTUROS	89
6.1	Resumo do Trabalho	89
6.2	Contribuições da Pesquisa	90
6.2.1	<i>Conclusões e Aplicações</i>	90
6.2.2	<i>Limitações do Sistema Proposto</i>	91
6.2.3	<i>Trabalhos Submetidos e Publicados</i>	92
6.3	Trabalhos Futuros	93
	REFERÊNCIAS	95
	APÊNDICE A CONSULTAS SPARQL	101

INTRODUÇÃO

Devido à popularização da Internet e à melhora de sua infraestrutura ao longo dos últimos anos, observou-se um crescimento na quantidade de produtos e serviços que as pessoas podem consumir na Web. Existem inúmeros sistemas de lojas de livros, roupas e artigos esportivos, além de sistemas de serviços de locação de filmes ou de acesso a músicas e notícias. Tal diversidade de opções faz com que usuários tenham uma dificuldade para encontrar o que desejam, ocasionando um problema conhecido como sobrecarga de informação.

Os Sistemas de Recomendação fazem parte de um importante mecanismo que ajuda os usuários a lidar com a crescente sobrecarga de informação. Eles fornecem sugestões de itens e serviços, que são selecionados automaticamente de acordo com as preferências e interesses dos usuários. Essas ferramentas seguem basicamente dois princípios para retornar itens para um usuário: a partir de um texto de busca ou recomendações a partir de um perfil traçado do usuário (KAGIE; WEZEL; GROENEN, 2011).

As duas abordagens mais populares de sistemas de recomendação são a baseada em conteúdo e a baseada em filtragem colaborativa (ADOMAVICIUS; TUZHILIN, 2005). Quando baseado em conteúdo, o sistema traça o perfil de cada usuário baseando-se em itens acessados anteriormente por eles e retorna, para cada um deles, itens novos que estejam alinhados com seus respectivos perfis. Sistemas colaborativos realizam recomendações baseadas em avaliações e interações feitas por outros usuários no sistema. Neste contexto, duas técnicas se destacam na filtragem colaborativa: i) modelos de vizinhança, que buscam e comparam avaliações de grupos de usuários ou itens semelhantes para prever notas; e ii) modelos de fatores latentes, que têm se revelado um caminho alternativo para inserir itens e usuários em um mesmo espaço latente de características, possibilitando que eles possam ser comparados diretamente (KOREN; BELL, 2011).

Nos modelos de vizinhança, grupos de usuários ou itens similares são construídos para auxiliar o sistema na geração de recomendações ao usuário alvo. Dessa maneira, se um usuário

gostou de um determinado item, outros itens do mesmo grupo podem ser recomendados para ele. Esses grupos são usualmente construídos usando-se métricas de correlação (Pearson, Cosseno, entre outras) aplicadas a vetores de notas, onde cada vetor representa o histórico de avaliações de um determinado usuário sobre os itens avaliados por ele.

Os principais desafios a serem resolvidos em sistemas colaborativos baseados em vizinhança estão associados à inserção de usuários e itens novos, problema também conhecido como partida fria (DESROSIERS; KARYPIS, 2011). Visto que o modelo de vizinhança busca usuários com perfis similares ao do usuário alvo, um indivíduo novo, com poucas interações no sistema não terá um perfil completo o suficiente para ser comparado com os demais usuários. Consequentemente, as recomendações geradas para esse usuário podem ser inconsistentes com seus interesses. A inserção de novos itens também representa um desafio a ser resolvido porque eles só serão recomendados após receberem boas avaliações por um número considerável de indivíduos. Além disso, quando os itens avaliados por cada usuário são representados em uma matriz, é possível observar uma grande esparsidade nos dados, que demanda um tratamento especial na sua utilização. Internamente, no cenário de partida fria, o sistema lida com um vetor de notas que não está preenchido com notas suficientes para que seja possível calcular sua correlação com outros vetores. Por exemplo, se um item é novo no sistema, ele não terá avaliações suficientes de usuários, e seu vetor de notas estará praticamente vazio. Consequentemente, ele não será associado corretamente a um grupo de itens similares para ser recomendado futuramente (BURKE; RAMEZANI, 2011).

Para resolver esse problema, uma alternativa é usar os metadados disponíveis de cada item. No caso de filmes, por exemplo, metadados podem ser o título, ano de lançamento, lista de atores, gêneros, entre outros, que descrevem um filme. Usando essas descrições, é possível realizar recomendações usando duas abordagens diferentes: i) baseada em conteúdo, conforme mencionado anteriormente, de modo que os perfis de usuários sejam traçados com palavras-chave que serão usadas para selecionar itens com metadados similares; e ii) criação de vetores de descrições, os quais são usados para calcular as correlações de itens de maneira similar ao cálculo com vetores de notas.

Em relação ao uso de metadados em sistemas de recomendação, um desafio é a obtenção de dados estruturados e com carga semântica para descrever os itens do acervo, possibilitando que o sistema possa inferir conclusões de modo semelhante ao que acontece com os humanos quando adquirem compreensão sobre as coisas. Tais limitações têm sido exploradas em estudos recentes sobre o uso de bases de conhecimento da Web como fonte de informações semânticas (OSTUNI *et al.*, 2013) (ROWE, 2014) (ZHANG *et al.*, 2016).

As bases de conhecimento são ferramentas desenvolvidas na área de Web Semântica que provêem dados estruturados de uma variedade de fontes da Web. O objetivo principal da Web Semântica é o desenvolvimento de estruturas de dados para representar informações na Web, que possam ser interpretadas e compartilhadas por máquinas (ANTONIOU; VANHAR-

MELEN, 2004). A ideia é oferecer um formato de dados que favoreça o desenvolvimento de softwares capazes de procurar e filtrar informações relevantes para os usuários (ANTONIOU; VANHARMELEN, 2004).

Os avanços na área de pesquisa da Web Semântica têm alavancado o desenvolvimento de sistemas de recomendação baseados em bases de conhecimento, possibilitando que as recomendações possam ser mais eficientes e significativas em relação ao que os usuários gostaram no passado. De fato, muitos estudos buscam uma maneira de combinar os frutos dessas duas áreas de pesquisa com sucesso (OSTUNI *et al.*, 2013)(HARISPE *et al.*, 2013)(ZHANG *et al.*, 2016). No caso de modelos de vizinhança, por exemplo, itens agrupados com base em descrições semânticas é uma abordagem promissora, visto que isso favorece a recomendação de itens semanticamente relacionados, em particular no cenário de partida fria.

1.1 Motivação

Uma alternativa para superar os desafios relacionados à descrição semântica de itens para sistemas de recomendação baseados em vizinhança é extrair as informações necessárias de bases de conhecimento. Existem diversas bases sobre vários domínios de conhecimento que descrevem conceitos, entidades e seus relacionamentos de modo a fornecer informações completas e semânticas sobre os recursos representados. Os recursos fazem parte de uma estrutura modelada de acordo com uma ontologia que representa um domínio específico de conhecimento. A linguagem utilizada para a representação de uma ontologia pode permitir a modelagem de diversos tipos de relacionamentos entre recursos, restrições de intervalo e domínio das propriedades dos recursos, entre outras (HEBELER *et al.*, 2009).

Contudo, pode não ser tão simples encontrar uma base de conhecimento que seja abrangente o suficiente para o domínio trabalhado e que ao mesmo tempo utilize a versão completa da linguagem OWL (do Inglês, *Web Ontology Language*)¹, algo raro, para modelar sua ontologia. É desejável também que a base seja popular e utilizada em larga escala, indicando uma grande aceitação da comunidade em relação à ontologia modelada, e que seja de domínio público, para facilitar a integração com outros sistemas (BAUER; KALTENBÖCK, 2012).

A escolha de uma base de dados integrante da LOD (Dados Abertos Conectados, do Inglês, *Linked Open Data*), bases de conhecimento públicas inter-relacionadas, resolve boa parte dos problemas, pois o sistema estaria conectado indiretamente a várias bases públicas, o que aumenta a cobertura na representação de conceitos e entidades. Apesar de já existirem estudos que exploram LOD para descreverem itens do acervo, poucos exploram as ontologias profundamente para encontrar informações implícitas, suprimindo a ausência de modelagem de algumas regras e relacionamentos referentes ao domínio de conhecimento, ou para se calcular a distância semântica entre conceitos, algo muito útil para determinar a similaridade entre itens e

¹ <http://www.w3.org/TR/owl-features/>

perfis de usuários. Desse modo, é fundamental o estudo de meios para se obter a maior carga semântica possível nas descrições de itens a partir da exploração de ontologias, e maneiras de se beneficiar de tais descrições no cálculo da similaridade entre itens.

1.2 Objetivo

O objetivo principal deste trabalho de pesquisa é investigar como descrições semânticas construídas a partir de informações extraídas de bases de conhecimento na Web podem amenizar os problemas provenientes da ausência de semântica na descrição de itens em sistemas de recomendação. Para isso, propõe-se um sistema de recomendação híbrido, que é um sistema de filtragem colaborativa com algumas adaptações extraídas do sistema de recomendação baseado em conteúdo. Esse sistema possui um módulo responsável pelo acesso a uma base de conhecimento para explorar sua ontologia em busca de informações sobre os itens do acervo do sistema e formatar os dados retornados. A varredura pela ontologia é realizada por um conjunto de consultas específicas elaboradas na linguagem SPARQL (do Inglês, *SPARQL Protocol and RDF Query Language*)², a fim de completar ou suprir a ausência de algumas regras na modelagem da ontologia. Outro módulo do sistema realiza a análise dos dados obtidos na varredura da base de conhecimento e calcula a similaridade semântica entre os itens para que os itens possam ser agrupados por similaridade. Tais informações obtidas e calculadas são, por fim, aplicadas no sistema de recomendação para melhorar sua acurácia, tanto no cenário normal de recomendação de itens previamente avaliados, quanto no cenário de partida fria.

1.3 Estrutura do Trabalho

Este documento está estruturado em capítulos que apresentam os principais métodos e técnicas relacionados com o trabalho realizado. O Capítulo 2 apresenta fundamentos básicos sobre sistemas de recomendação e fornece uma visão geral sobre essa área de pesquisa. Nele, são apresentados as principais abordagens de recomendação e métricas para avaliar a eficiência de sistemas de recomendação. No Capítulo 3 são apresentados alguns conceitos relacionados com a Web Semântica, os quais serão explorados neste trabalho para se obter as informações semânticas dos itens. São apresentadas: linguagens e formatos para representar e serializar informações semanticamente; maneiras de disponibilizar e consultar tais informações; modos de explorar ontologias e obter informações implícitas na estrutura; e soluções relatadas em trabalhos relacionados ao deste trabalho. O Capítulo 4 expõe com maiores detalhes os objetivos deste trabalho e apresenta a arquitetura em alto nível e a metodologia proposta para alcançar os objetivos traçados. Os detalhes de implementação da arquitetura e metodologia propostas estão descritos no Capítulo 5. Nesse capítulo também são apresentados e discutidos os resultados dos experimentos realizados ao longo deste trabalho. Por fim, o Capítulo 6 conclui este documento

² <http://www.w3.org/TR/rdf-sparql-query/>

listando as principais contribuições geradas por este trabalho e apresenta possíveis futuros trabalhos.

SISTEMAS DE RECOMENDAÇÃO

Sistemas de Recomendação (SR) são softwares com a habilidade de sugerir itens a usuários no intuito de auxiliar no processo de tomada de decisão em relação a diversas ações, como comprar itens, ouvir música, assistir a filmes, ler notícias, entre outras (RICCI; ROKACH; SHAPIRA, 2011). O termo item é usado para denotar o que o sistema recomenda, como CDs, livros ou lugares, por exemplo.

Os SRs são de grande utilidade em duas principais situações: quando o usuário não possui conhecimento suficiente para tomar uma decisão sobre um domínio específico ou quando ele não é capaz de avaliar todas as opções disponíveis, geralmente por causa do grande volume de opções. Um modo simples de gerar recomendações aos usuários é selecionar e exibir os itens mais populares no sistema ou os mais vendidos, por exemplo. Nesses casos, temos as mesmas recomendações para todos os usuários. Essas recomendações podem ser úteis, dependendo do contexto, mas a área de pesquisa de SR tem especial interesse em recomendações personalizadas. Tais recomendações tendem a atingir um grau de satisfação maior com o indivíduo, visto que as sugestões são alinhadas com as preferências de cada um. As recomendações personalizadas dos SRs são baseadas em perfis que são construídos a partir de informações explícitas e implícitas fornecidas pelos usuários. Um indivíduo pode fornecer informações explicitamente sobre seus interesses durante seu cadastro no sistema ou avaliando itens do acervo, antes ou depois de acessá-los no sistema. Mais informações sobre os interesses de um usuário podem ser coletadas pelo SR implicitamente observando o comportamento do mesmo e suas interações com o sistema, tais como itens pesquisados, páginas observadas, histórico de navegação, entre outros (RICCI; ROKACH; SHAPIRA, 2011).

Além dos benefícios mencionados, os provedores de conteúdo e serviços também usufruem de vantagens ao incorporar SRs em seus sistemas. Dentre essas vantagens destacam-se: o aumento de itens vendidos ou acessados; o aumento na diversidade de itens vendidos ou acessados; aumento na satisfação dos usuários; aumento na fidelidade do usuário; e melhor

compreensão do que o usuário quer (RICCI; ROKACH; SHAPIRA, 2011).

Dependendo do domínio da aplicação e dos objetivos a serem alcançados, os SRs podem usar abordagens diferentes para retornar recomendações a seus usuários. As principais abordagens relatadas na literatura serão apresentadas na próxima seção.

2.1 Abordagens de Recomendação

Dentre as abordagens mais utilizadas e mencionadas na literatura estão a filtragem baseada em conteúdo e a filtragem colaborativa. Porém, outras abordagens merecem destaque, como filtragem demográfica, baseada em conhecimento, ciência de contexto, além de técnicas de personalização e sumarização (RICCI; ROKACH; SHAPIRA, 2011). Uma alternativa interessante é a combinação de técnicas diferentes em um mesmo SR, os chamados sistemas híbridos. Todas as abordagens possuem problemas não resolvidos ou limitações, mas em um sistema híbrido isso é atenuado, pois as técnicas trabalham harmoniosamente e se auxiliam mutuamente.

As próximas subseções apresentam detalhes sobre a filtragem baseada em conteúdo, filtragem colaborativa e alguns exemplos de sistemas híbridos.

2.1.1 Filtragem Baseada em Conteúdo

Sistemas que adotam a abordagem de recomendação baseada em conteúdo traçam os perfis dos seus usuários a partir das características encontradas nos itens avaliados por eles. Dessa maneira, o perfil de um usuário é uma representação estruturada de seus interesses no domínio do sistema que será usado para recomendar-lhe novos itens relevantes. O processo de recomendação consiste basicamente em encontrar itens no acervo do sistema que possuem características que combinam com as presentes no perfil do usuário alvo. Em outras palavras, pode-se estimar a utilidade de cada item para um usuário e filtrar os itens que serão retornados a partir de um limiar pré-estabelecido.

A utilidade $util(u, i)$ do item i para o usuário u é estimada baseando-se nas utilidades $util(u, r_n)$ atribuídas pelo usuário u aos itens $r_n \in R_i$ que são similares ao item i (ADOMAVICIUS; TUZHILIN, 2005). Ou seja, a avaliação do usuário (explícita ou implícita) sobre um item reflete a utilidade do mesmo para ele. Então, itens similares àqueles avaliados pelo usuário provavelmente também possuirão utilidade semelhante.

Porém, os itens não são comparados diretamente entre si. Os metadados que descrevem os itens bem avaliados pelo usuário serão adicionados ao perfil dele, assim o sistema pode confrontar as descrições dos itens ainda não acessados com o seu perfil e estimar a utilidade dos mesmos para o usuário em questão. Em alguns casos (LOPS; GEMMIS; SEMERARO, 2011) o sistema pode inserir metadados de itens mal avaliados ao perfil do usuário para representarem características que o usuário não gosta. Para cada informação presente no seu perfil de interesses,

é calculado um peso por meio de um algoritmo que realiza uma avaliação do usuário sobre o conteúdo visitado observando fatores como: notas, questionários, tempo de visitação, comentários e anotações (MANZATO, 2010).

O processo de recomendação pode ser realizado em três etapas, cada qual sendo manipulada por componentes individuais (LOPS; GEMMIS; SEMERARO, 2011), conforme esquematizado na Figura 1 e descrito a seguir:

Figura 1 – Arquitetura em Alto Nível de um Sistema de Recomendação Baseado em Conteúdo, traduzido de Lops, Gemmis e Semeraro (2011)

- **Analisador de conteúdo:** esse componente é responsável por extrair e representar o conteúdo dos itens disponíveis, tais como notícias, músicas e vídeos, de forma estruturada e acessível para os demais componentes. Técnicas de extração de características (SNOEK; WORRING, 2005) são utilizadas para converter a representação das informações presentes em um item, em uma estrutura adequada para o sistema, como um vetor de palavras-chave, por exemplo;
- **Construtor de perfil:** responsável por construir os perfis dos usuários com dados que representem seus interesses. Usualmente, esses dados são obtidos a partir de itens já avaliados previamente pelo usuário;
- **Filtragem:** explora o perfil do usuário e o confronta com as características dos itens do acervo para recomendar itens relevantes. Isso gera uma lista de itens ranqueados por

similaridade entre o perfil e os itens. A similaridade pode ser medida por similaridade de cosseno (BOBADILLA *et al.*, 2013) quando o perfil e as características dos itens forem representados por vetores, por exemplo.

A primeira etapa do processo de recomendação é realizada pelo Analisador de Conteúdo, que na maioria dos sistemas utiliza técnicas originadas na área de Recuperação de Informação (MANNING; RAGHAVAN; SCHUTZE, 2009). O Analisador de Conteúdo extrai características dos itens, como palavras-chave, conceitos ou histogramas, processando os dados obtidos dos próprios itens. Extraídas as características, essas são armazenadas de forma estruturada em um repositório.

As reações dos usuários em relação aos itens consumidos são coletadas, processadas e armazenadas. Essas reações, chamadas de anotações ou *feedback*, são exploradas juntamente com as descrições dos itens relacionados no processo de construção do perfil do indivíduo para que se possa prever a relevância de itens novos a serem apresentados aos usuários. Os usuários podem ainda informar explicitamente seus interesses ao sistema para se obter um perfil inicial, quando ele ainda não realizou nenhum *feedback*, por exemplo. Os *feedbacks* podem revelar basicamente dois tipos de sentimentos do usuário: aceitação (*feedback* positivo), que indica um interesse pelas características presentes no item avaliado; ou rejeição (*feedback* negativo), que indica uma falta de interesse pelas características presentes no item avaliado. Dependendo da maneira como o *feedback* é obtido, ele é classificado como *feedback* explícito, quando o usuário avalia diretamente um item, ou como *feedback* implícito, quando ele é obtido a partir do monitoramento de suas atividades.

A filtragem baseada em conteúdo possui algumas vantagens em relação à filtragem colaborativa (LOPS; GEMMIS; SEMERARO, 2011):

- **Independência de outros usuários:** para recomendar itens a um determinado usuário, um sistema que adota a filtragem baseada em conteúdo necessita apenas dos *feedbacks* coletados desse usuário para traçar seu perfil e buscar itens similares. Por outro lado, sistemas que adotam filtragem colaborativa dependem do *feedback* de outros usuários com perfis similares ao do usuário ativo para lhe recomendar itens bem avaliados por eles.
- **Transparência:** o sistema pode fornecer explicações sobre os itens recomendados usando suas características, ou ainda esclarecer porque cada item foi recomendado. Já em um sistema colaborativo, a única informação disponível é que um conjunto de indivíduos com preferências similares a do usuário alvo avaliou positivamente o item recomendado.
- **Item novo:** para que um item novo seja recomendado por sistemas baseados em conteúdo, basta que as características dele sejam similares às do perfil do usuário ativo. Isso difere do que ocorre em sistemas colaborativos, nos quais um item só poderá ser recomendado a um usuário quando ele acumular avaliações positivas de usuários.

Porém, conforme relatado por [Adomavicius e Tuzhilin \(2005\)](#), alguns problemas são inerentes a essa abordagem:

- **Análise limitada de conteúdo:** sistemas baseados em conteúdo precisam extrair características dos itens para poder confrontá-los com os perfis dos usuários e essa não é uma tarefa fácil, independentemente se for realizada manual ou automaticamente. Em alguns sistemas, apenas alguns aspectos são explorados em um item, porém os demais aspectos poderiam influenciar na construção ou atualização dos perfis dos usuários. Como exemplo, um item bem avaliado sobre *rally*, do qual foi extraída a característica “corrida de carros”, pode levar o sistema a inferir que o usuário gosta de corridas de carros, quando na verdade ele gosta de *rally*, mas odeia fórmula *Indy*;
- **Sobre-especialização:** um sistema puramente baseado em conteúdo é incapaz de recomendar itens que não tenham alta similaridade com o perfil do usuário, ou seja, somente itens parecidos com os já acessados serão selecionados para o usuário. Isso limita o sistema a recomendar itens com apenas certo nível de novidade. Se um usuário avaliar apenas filmes de aventura, o sistema não lhe recomendará filmes de comédia, por exemplo;
- **Problema do novo usuário:** o sistema só produzirá recomendações confiáveis para um usuário quando conseguir construir o perfil dele de modo suficientemente completo para entender seus interesses. Isso só acontecerá após o usuário avaliar itens suficientes para que seus interesses sejam identificados corretamente.

2.1.2 Filtragem Colaborativa

Diferentemente da recomendação baseada em conteúdo, a recomendação colaborativa procura prever a utilidade $util(u, i)$ de um item i para um usuário u baseando-se nas utilidades $util(p_n, i)$ associadas ao item i por outros usuários $p_n \in P_u$ que são similares ao usuário u ([ADOMAVICIUS; TUZHILIN, 2005](#)). A recomendação também pode ser feita explorando-se a similaridade entre os itens, ou seja, recomendar itens semelhantes aos do conteúdo já avaliado positivamente pelo usuário. Esta técnica reduz o problema da sobre-especialização, citada no tópico anterior, porque o sistema leva em consideração apenas as notas de avaliação associadas previamente por outros usuários e nenhum metadado referente ao conteúdo. Contudo, manter os relacionamentos entre usuários semelhantes (e também entre itens semelhantes) exige um custo adicional ([DESROSIERS; KARYPIS, 2011](#)). Também é possível observar limitações em relação a usuários novos, itens novos e usuários com preferências muito distintas dos demais usuários.

A filtragem colaborativa pode ser realizada de duas formas, basicamente: baseada em vizinhança ou por fatoração de matrizes. Esses dois métodos são apresentados nas próximas subseções.

2.1.2.1 Métodos Baseados em Vizinhança

Na filtragem colaborativa baseada em vizinhança a predição de notas para itens novos é calculada diretamente a partir das notas que os usuários atribuíram aos itens do acervo. Os sistemas que adotam essa abordagem podem ainda ser divididos em dois grupos: baseado em usuários e baseados em itens.

Os sistemas baseados em usuários avaliam o interesse de um usuário u em um item i usando as notas atribuídas a i por outros usuários, chamados de vizinhos, que possuem avaliações semelhantes as do usuário u em relação a itens acessados em comum (BOBADILLA *et al.*, 2013). Nos cálculos para predição apresentados a seguir, serão denotados por P_u o conjunto de usuários que apresentam preferências mais similares as do usuário u e por $P_u(i)$ os usuários pertencentes a P_u que avaliaram o item i .

O sistema de avaliação adotado também influencia na maneira como a predição de notas será calculada, pois as notas podem ser valores numéricos (-10 a 10, por exemplo) ou descritivos (ruim, regular, bom, muito bom ou ótimo, por exemplo). Para valores numéricos, usualmente, calcula-se a predição por meio de regressão. Para valores descritivos ou que não podem ser ordenados, usualmente, calcula-se a predição por meio de classificação (AGGARWAL, 2016).

Para predizer a nota $\hat{r}_{ui}$ do usuário u para o item i usando regressão, pode-se simplesmente calcular a média das notas r_{pi} de cada usuário p pertencente a $P_u(i)$, conforme a Equação 2.1.

$$\hat{r}_{ui} = \frac{1}{|P_u(i)|} \sum_{p \in P_u(i)} r_{pi} \quad (2.1)$$

Porém, a Equação 2.1 não leva em conta os diferentes níveis de similaridade de cada usuário p em relação a u . Sendo assim, é necessário atribuir um peso w_{up} às notas de cada usuário p conforme o grau de similaridade com u :

$$\hat{r}_{ui} = \frac{\sum_{p \in P_u(i)} w_{up} r_{pi}}{\sum_{p \in P_u(i)} |w_{up}|} \quad (2.2)$$

Utiliza-se o módulo de w_{up} no denominador da Equação 2.2, pois valores negativos podem gerar notas fora do intervalo estabelecido.

Contudo, ainda há outra variável que precisa ser considerada. Usuários diferentes têm critérios diferentes para avaliar um item, de modo que alguns podem ser mais propensos a atribuir notas mais altas que outros. Isso pode ser resolvido normalizando-se as notas $h(r_{pi})$ de cada vizinho p , conforme a Equação 2.3 (DESROSIERS; KARYPIS, 2011):

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{p \in P_u(i)} w_{up} h(r_{pi})}{\sum_{p \in P_u(i)} |w_{up}|} \right), \quad (2.3)$$

onde h^{-1} garante a conversão da predição para a escala original adotada.

Quando o cálculo da predição por classificação for necessário, a predição pode ser o voto p_{in} dado pelos usuários $P_u(i)$, conforme a nota dada ao item i e a similaridade de cada um com o

usuário u , conforme segue (DESROSIERS; KARYPIS, 2011):

$$p_{in} = \arg \max_{n \in N} \sum_{p \in P_u(i)} \delta(n_{pi} = n) w_{up}, \quad (2.4)$$

onde N é o conjunto de notas possíveis e $\delta(n_{pi} = n)$ é 1 se $n_{pi} = n$, e 0 caso contrário.

O outro grupo de sistemas que também adota filtragem colaborativa baseada em vizinhança, mas realiza a predição de notas baseada em itens, segue uma lógica parecida da apresentada acima, porém focada nos itens semelhantes. Ou seja, realiza-se a predição observando as notas que foram atribuídas pelo usuário u a outros itens semelhantes ao item i . Denotando por $R_i(u)$ como o conjunto de itens mais semelhantes ao item i que foram avaliados pelo usuário u , a predição da nota $\hat{r}_{ui}$ do usuário u para o item i será a média ponderada das notas r_{uj} dadas pelo usuário u a cada item j pertencente a $R_i(u)$. Adaptando a Equação 2.2 para esse contexto, onde w_{ij} representa a similaridade entre os itens i e j , tem-se (DESROSIERS; KARYPIS, 2011):

$$\hat{r}_{ui} = \frac{\sum_{j \in R_i(u)} w_{ij} r_{uj}}{\sum_{j \in R_i(u)} |w_{ij}|} \quad (2.5)$$

Também é possível levar em conta os diferentes critérios de avaliação de usuários diferentes na abordagem baseada em itens e normalizar as notas com $h(r_{uj})$ (DESROSIERS; KARYPIS, 2011):

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{j \in R_i(u)} w_{ij} h(r_{uj})}{\sum_{j \in R_i(u)} |w_{ij}|} \right) \quad (2.6)$$

Analogamente, também é possível realizar a predição por classificação, calculando o voto u_{jn} do usuário u para os itens $j \in R_i(u)$, conforme a nota dada a cada j e a similaridade dele com o item i , chegando à versão normalizada expressa na Equação 2.7 (DESROSIERS; KARYPIS, 2011):

$$u_{jn} = h^{-1} \left(\arg \max_{n \in N} \sum_{j \in R_i(u)} \delta(h(n_{uj}) = n) w_{ij} \right) \quad (2.7)$$

2.1.2.2 Fatoração de Matrizes

A fatoração de matrizes tem se popularizado entre sistemas de filtragem colaborativa devido à acurácia de predição de notas, escalabilidade e aprendizagem relativamente rápidos que podem ser observados em sistemas baseados nesses modelos. Eles contam com fatores latentes para revelar características latentes dos itens sob as notas atribuídas pelos usuários (RENDLE, 2011). Em outras palavras, pode-se decompor uma matriz M de notas que relaciona usuários e itens em outras duas matrizes C e D , dado um número f de fatores considerados. A matriz C relaciona usuários com fatores latentes e indica o grau de interesse dos usuários por tais fatores. De maneira semelhante, a matriz D relaciona itens com fatores latentes e indica o grau de relevância do item frente a tais fatores. Os valores encontrados em C e D podem ser positivos ou negativos, indicando afinidade ou aversão em C e presença ou ausência em D .

A partir das matrizes C e D pode-se prever a nota $\hat{r}_{ui}$ do usuário u para o item i conforme a Equação 2.8:

$$\hat{r}_{ui} = \mu + b_i + b_u + d_i^T c_u \quad (2.8)$$

onde μ é a nota média global, b_u e b_i são, respectivamente, o desvio médio das notas do usuário u e o desvio médio das notas do item i , ambos em relação à média global. $d_i^T c_u$ é o produto entre o item d_i da transposta de D e o usuário c_u de C . Para minimizar o erro quadrático, um parâmetro é adicionado à equação e seu valor será definido durante o treinamento do sistema em um conjunto de treinamento, normalmente usando validação cruzada, aplicando-se os conceitos de gradiente descendente estocástico (Equação 2.9) (KOREN; BELL, 2011).

$$\min_{b_*, d_*, c_*} \sum_{(u,i) \in G} (r_{ui} - \mu - b_i - b_u - d_i^T c_u)^2 + \lambda (b_i^2 + b_u^2 + \|d_i\|^2 + \|c_u\|^2), \quad (2.9)$$

onde G é o conjunto de pares (u, i) que possuem nota, ou seja, do conjunto de treinamento, e λ é o fator de regularização definido por validação cruzada.

Entretanto, essa técnica apresenta problemas em relação a novos usuários ou itens no sistema e necessita de método de preenchimento da matriz usuário-item, pois esta geralmente é esparsa. O preenchimento da matriz pode ser realizado com a inserção de zeros onde há ausência de valor, ou usando um método mais complexo para se obter maior acurácia na predição de notas (KOREN; BELL, 2011).

2.1.3 Sistemas Híbridos

A recomendação híbrida é uma combinação das duas técnicas estudadas anteriormente a fim de reduzir os problemas encontrados em cada uma delas. São estudados quatro modos de combinar estas técnicas: i) implementação isolada de ambas as técnicas e posterior combinação dos resultados; ii) sistema de recomendação baseado em conteúdo com algumas adaptações extraídas do sistema de recomendação colaborativa; iii) sistema de recomendação colaborativo com algumas adaptações extraídas do sistema de recomendação baseada em conteúdo; e iv) construção de um modelo genérico que incorpora características de ambas as técnicas (ADOMAVICIUS; TUZHILIN, 2005).

2.1.4 Outras Abordagens Relevantes

Existem outras abordagens de sistemas de recomendação que adotam métodos de filtragem menos populares (RICCI; ROKACH; SHAPIRA, 2011). Dentre eles, serão apresentados a seguir outros três métodos de filtragem que têm ganhado destaque: contextual (ARAÚJO, 2013; ADOMAVICIUS; TUZHILIN, 2011), baseada em conhecimento (FELFERNIG *et al.*, 2011) e baseada em comunidade (VICTOR; COCK; CORNELIS, 2011).

Sistemas que adotam uma filtragem contextual exploram o contexto no qual o usuário está inserido para fazer recomendações mais adequadas ao momento, local e dispositivo utilizado,

por exemplo. Araújo (2013) apresenta um estudo de caso, em que um sistema captura e armazena materiais didáticos (vídeo, áudio e *slides*) gerados durante aulas ministradas em um ambiente previamente preparado e possibilita a recuperação dessas mídias de maneira personalizada. Um aluno que acessa o sistema de seu *smartphone* e o informa que dispõe de apenas 30 minutos para estudar, pode receber uma lista de materiais didáticos que prioriza *slides* sobre a disciplina selecionada, visto que esse tipo de material propõe uma revisão rápida e não exige uma conexão excelente para *download*.

Sistemas baseados em conhecimento realizam suas recomendações com base em um domínio de conhecimento específico buscando itens conforme a necessidade e preferência do usuário. Existem dois tipos de sistemas baseados em conhecimento: baseado em casos e baseado em restrições. Ambos analisam os requisitos fornecidos pelo usuário, sugerem reformulações automaticamente quando os requisitos fornecidos são inconsistentes e apresentam uma explicação sobre os resultados retornados. A principal diferença entre eles está no modo que são calculadas as recomendações. Um sistema baseado em casos tem como base métricas de similaridade, enquanto em um sistema baseado em restrições as recomendações são obtidas a partir de bases de conhecimento predefinidas que contêm regras explícitas de como encontrar itens com as características requeridas pelo usuário.

A filtragem baseada em comunidade recomenda itens de acordo com as preferências dos amigos do usuário alvo. Essa abordagem é muito parecida com a filtragem colaborativa, sendo apenas mais restrita na seleção dos vizinhos, cujas avaliações serão analisadas para prever a nota de um determinado usuário para um item, pois somente amigos são selecionados.

2.2 Avaliação de Sistemas de Recomendação

A avaliação de sistemas de recomendação tem a finalidade de mensurar a eficiência da abordagem dado um conjunto de propriedades a serem avaliadas. Ela pode ser útil no momento de selecionar a técnica que será adotada em um sistema ou no momento de avaliar melhorias ou extensões. Dentre as propriedades que podem ser avaliadas estão: o sistema de predição de notas, a capacidade de recomendar itens novos fora do perfil do usuário que o agradem, a diversidade de itens recomendados, confiabilidade, privacidade, robustez, escalabilidade, velocidade no tempo de resposta, entre outros (SHANI; GUNAWARDANA, 2011). Diferentes sistemas têm diferentes necessidades e algumas propriedades podem apresentar pior desempenho em detrimento da melhoria de outra propriedade. Então, o projetista precisa estabelecer prioridades maiores às propriedades que proporcionarão uma melhor experiência aos usuários do sistema.

Uma das propriedades que recebem maior atenção em sistemas de recomendação é a predição de notas, visto que a maioria deles baseia suas recomendações na predição da opinião de usuários sobre itens (notas para filmes, por exemplo) do acervo ou na predição de utilidade de itens para usuários (utilidade para compra, por exemplo).

A avaliação pode abordar basicamente três tipos de experimentos: i) *offline*, no qual são usados conjuntos de dados que contemplam dados fornecidos por usuários, como notas e anotações entre outros, que possibilitam simular a interação dos mesmos; ii) estudo de usuário, no qual será solicitado a um pequeno grupo de usuários que realizem algumas tarefas usando o sistema e, posteriormente, as opiniões dos usuários são coletadas, no formato de um questionário, por exemplo; e iii) *online*, no qual o experimento é realizado com um grande número de usuários, o que permite avaliar eventuais situações que não são observadas em um experimento controlado (SHANI; GUNAWARDANA, 2011).

Neste trabalho o foco serão os experimentos *offline* porque são facilmente reproduzidos e facilitam a comparação entre versões diferentes do sistema, ou até mesmo de sistemas diferentes. Experimentos *offline* são atrativos porque não exigem a interação de usuários reais e possibilitam a comparação de um grande número de algoritmos a um custo baixo. Nesse tipo de experimento, geralmente são usadas três medidas de qualidade para avaliar a acurácia do sistema: i) avaliação da predição de notas; ii) avaliação de conjuntos de recomendações; e iii) avaliação de recomendações em formato de *ranking* (BOBADILLA *et al.*, 2013).

As próximas subseções apresentam em detalhes as medidas de qualidade mencionadas acima e suas respectivas métricas.

2.2.1 Avaliação da Predição

Como citado anteriormente, muitos sistemas de recomendação adotam a técnica de predição de notas que um determinado usuário daria a itens ainda desconhecidos. Para se avaliar a eficiência na predição de notas é necessário quantificar o erro em cada predição, ou seja, analisar o quanto a predição diverge da nota real atribuída pelo usuário. Visto que o objetivo de todo sistema de recomendação que usa predição de notas é reduzir esse erro, pode-se usar as métricas de avaliação para treinar iterativamente o sistema.

Segundo Bobadilla *et al.* (2013), as métricas mais comuns documentadas na literatura são: MAE (Erro Médio Absoluto, do Inglês, *Mean Absolute Error*) (Equação 2.10) e o RMSE (Raiz do Erro Quadrático Médio, do Inglês, *Root Mean Square Error*) (Equação 2.11).

$$MAE = \sqrt{\frac{1}{|H|} \sum_{(u,i) \in H} |\hat{r}_{ui} - r_{ui}|} \quad (2.10)$$

$$RMSE = \sqrt{\frac{1}{|H|} \sum_{(u,i) \in H} (\hat{r}_{ui} - r_{ui})^2} \quad (2.11)$$

onde H é conjunto de pares usuário-item do conjunto de teste, r_{ui} é a nota real do usuário u para o item i , e $\hat{r}_{ui}$ é a nota predita do usuário u para o item i .

É possível observar que RMSE penaliza desproporcionalmente erros maiores, se comparado com o MAE.

2.2.2 Validação Cruzada em k partições

Quando se opta por trabalhar com experimentos *offline* surge a necessidade de dividir a base de dados original, neste caso os registros das avaliações dos usuários, em dois conjuntos: um de treinamento; e outro de teste. O conjunto de treinamento será usado na fase de aprendizado do parâmetro supracitado, enquanto o conjunto de teste será usado para avaliar se o modelo obtém um bom desempenho com um conjunto de dados desconhecido.

O processo de divisão desses conjuntos pode ser aleatório, ou manipulado de modo a garantir certas propriedades em ambos subconjuntos. O processo pode, ainda, ser não exclusivo ou exclusivo. Ou seja, os itens selecionados para um conjunto podem, ou não, estar presentes no outro conjunto. Definido o método de separação, é necessário definir a proporção em que será realizada a divisão. Uma configuração bastante comum é a divisão aleatória exclusiva na proporção 80 por 20 (AMATRIAIN *et al.*, 2011). Isso significa que o conjunto original será dividido aleatoriamente de modo que uma avaliação presente no conjunto de treinamento, não aparecerá no conjunto de teste, e que 80% das avaliações ficarão para treinamento e 20% para teste na fase de aprendizagem do modelo e teste do sistema, respectivamente.

Tal divisão em treinamento e teste pode ser realizada k vezes, a fim de se calcular uma média a partir dos resultados obtidos em cada avaliação. Esse processo é conhecido como validação cruzada.

A validação cruzada em k partições é uma das técnicas de validação cruzada na qual o conjunto original é dividido, de modo exclusiva, em k subconjuntos e o sistema é executado k vezes. Para cada execução, um subconjunto diferente é selecionado para ser usado como conjunto de teste e os demais devem compor o conjunto de treinamento. Ao final das k execuções do sistema, uma média é calculada a partir dos resultados obtidos em cada avaliação.

Contudo, um sistema pode alcançar um ótimo desempenho em experimentos com uma determinada base de dados e obter resultados significativamente inferiores com outras bases de dados. Um modo de avaliar experimentos *offline* e obter conclusões confiáveis é realizando testes de significância estatística.

2.2.3 Significância Estatística

A ferramenta de teste de significância estatística mais tradicional é o nível de significância ou valor p , que representa a probabilidade dos resultados terem sido obtidos por acaso. Tradicionalmente usa-se $p = 0,05$ como o limite aceitável para a probabilidade do acaso, o que representa 95% de confiança nos resultados. Ou seja, se os resultados do sistema a são superiores aos do sistema b , mas o valor p calculado sobre os resultados dos experimentos ultrapassa 0,05, os resultados não são considerados significativos. Dentre os diferentes tipos de teste, o teste t de *Student* leva em consideração a magnitude das diferenças nos resultados ao examinar a probabilidade de um sistema a ser melhor que um sistema b . Isso evita eventuais inconsistências

em que o teste aponta uma superioridade do sistema *a* sobre o sistema *b*, mas é possível observar que, ocasionalmente, *b* supera *a* (SHANI; GUNAWARDANA, 2011).

Os testes de significância estatística podem ser realizados com resultados: pareados, obtidos em experimentos em que ambos sistemas foram executados sobre todos os conjuntos de teste; ou não-pareados, obtidos em experimentos em que não é possível executar os sistemas sobre os mesmos conjuntos de teste. Resultados pareados são obtidos facilmente em experimentos *offline*, nos quais a base de dados pode ser dividida em diferentes conjuntos de treinamento e teste, conforme descrito na Seção 2.2.2, por exemplo. Por outro lado, é comum a obtenção de resultados não-pareados em experimentos *online*, visto que podem ser usados conjuntos de usuários diferentes na execução dos sistemas comparados (SHANI; GUNAWARDANA, 2011).

2.3 Considerações Finais

Este capítulo apresentou alguns conceitos presentes em sistemas de recomendação e algumas das abordagens mais relatadas na literatura, assim como as dificuldades e problemas enfrentados em cada uma. Conforme mencionado anteriormente, tais problemas podem ser amenizados em diferentes versões de sistemas híbridos, visto que abordagens diferentes, usadas em conjunto, podem se auxiliar mutuamente nos problemas inerentes a cada uma.

Apesar das novas tendências e evoluções relatadas nos trabalhos apresentados na seção anterior, os problemas clássicos da área ainda persistem. As abordagens baseadas em conteúdo são, talvez, as que encontrem maiores dificuldades para substituir tarefas manuais ao longo do processo de recomendação. Na literatura é possível encontrar vários exemplos de sistemas de recomendação baseados em conteúdo, nos quais a descrição dos itens é realizada manualmente na tentativa de se obter descrições mais semânticas e amenizar o problema relacionado à análise limitada de conteúdo. Entretanto, tal tarefa é exaustiva e propensa a erros, tendo em vista a grande quantidade de itens disponíveis atualmente. Algumas linhas de pesquisa estudam meios de se obter descrições semânticas a partir de ontologias de bases de conhecimento na Web, que descrevem entidades, conceitos e seus relacionamentos com riqueza de detalhes. Desse modo, o próximo capítulo apresenta alguns conceitos relacionados com a Web Semântica, fornecendo meios de se usar ontologias para descrever semanticamente itens de maneira automática.

DADOS ABERTOS CONECTADOS

Pode-se considerar a Web atual como uma enorme comunidade, onde qualquer um pode contribuir com seus conhecimentos para que outros tenham acesso. Essas contribuições podem surgir das mais diversas fontes de dados e sobre qualquer tópico que se possa imaginar, resultando em um surpreendente e diversificado acervo de informações. Uma informação “Web” cresce e ganha mais volume conforme aumenta o interesse da comunidade em tal informação (ALLEMANG; HENDLER, 2011). Diante desse enorme acervo, é necessário cada vez mais a disponibilização de ferramentas de busca para encontrar informações sobre os assuntos de interesse, e ainda assim, apesar da vasta cobertura sobre o que se procura, muitas vezes as informações não são bem estruturadas e aprofundadas (ALLEMANG; HENDLER, 2011).

Outro problema que se pode observar é a inconsistência de informações sobre um mesmo assunto em diferentes sites ou em páginas que não foram atualizadas diante de novos acontecimentos ou descobertas. Nota-se também a necessidade de uma conexão maior entre os Websites para se facilitar as buscas e dar maior profundidade às informações. Um exemplo de inconsistência de informações poderia ser observado se o site oficial da FIFA¹ e o site do fã clube de um determinado jogador de futebol divergissem ao informar o clube atual do jogador. Outro inconveniente ocorre quando se faz necessário visitar vários sites para planejar a rota para uma viagem de carro, considerando custo com combustível, pedágios e qualidade dos restaurantes no trajeto.

Para viabilizar o desenvolvimento de aplicativos mais eficientes, integrados e com maior carga semântica, desenvolvedores e pesquisadores têm demonstrado um crescente interesse em explorar os benefícios advindos da Web dos Dados, a conhecida Web Semântica. O propósito da Web Semântica é o desenvolvimento de estruturas de dados interpretáveis e compartilháveis por máquinas para representar os dados na Web. A ideia principal é oferecer uma estrutura favorável ao desenvolvimento de softwares capazes de buscar e filtrar informações completas e relevantes

¹ <http://www.fifa.com/world-match-centre/nationalleagues/nationalleague=brazil-serie-a-2000000078>

para os usuários (ANTONIOU; VANHARMELEN, 2004). Visando a criação de tal estrutura, surgiu o conceito de Dados Conectados, que pode ser definido como um conjunto de boas práticas para publicar e conectar conjuntos de dados usando tecnologias que permitam a leitura desses dados por agentes de software de modo automático. Quando os princípios propostos por Tim Berners-Lee, conhecidos como “Sistema de 5 estrelas”, são somados a esse conjunto de boas práticas, temos o conceito de Dados Abertos Conectados que, de maneira resumida, garantem o acesso público aos dados (ISOTANI; BITTENCOURT, 2015).

Este capítulo tem o objetivo de apresentar estruturas e conceitos específicos da área de Web Semântica que podem ser utilizados para aprimorar a representação de itens em SRs. Nas próximas subseções serão apresentados os principais recursos que podem contribuir no sistema proposto neste trabalho.

3.1 Representação de Dados na Web Semântica

Na Web Semântica cada informação é representada por uma tripla composta por sujeito, predicado e objeto (HEBELER *et al.*, 2009). Essa tripla, também conhecida como declaração, pode ser comparada à estrutura gramatical de uma frase, na qual o sujeito é algo que será descrito e o predicado é o relacionamento que o sujeito possui com o objeto. Visto que o objeto de uma declaração pode ser o sujeito de outra declaração, essa estrutura pode ser representada por um grafo, no qual os sujeitos e objetos são os vértices e os predicados são as arestas (Figura 2). Para formalizar a estrutura das declarações e possibilitar seu compartilhamento, foi desenvolvido o modelo de dados RDF (do Inglês, *Resource Description Framework*).

3.1.1 RDF

RDF é um padrão que define um modelo de dados para representar as declarações e estruturar um grafo por meio de suas conexões. Nesse modelo, os sujeitos são representados por recursos (objetos, fatos ou conceitos) e os objetos, comumente, por valores literais. Porém, como dito anteriormente, o objeto de uma declaração pode ser o sujeito de outra. Logo, algumas declarações possuem um recurso representando seu sujeito e outro representando seu objeto.

Um recurso é identificado unicamente por um IRI (Identificador de Recursos Internacionalizado, do Inglês, *Internationalized Resource Identifiers*), uma maneira de nomear recursos na Web. Apesar de muitos IRIs serem do tipo URL (Localizador Padrão de Recursos, do Inglês, *Uniform Resource Locator*), que incluem informações de como recuperar os respectivos recursos na Web, um IRI não é necessariamente resolúvel na Web, como por exemplo o IRI *urn:isbn:8576651858* na Figura 3. A unicidade de um IRI é o ponto chave para o compartilhamento de informações nesse ambiente, pois um recurso pode ser referenciado inequivocamente em qualquer contexto. Essa característica faz do RDF um formato mais portátil que muitos outros (HEBELER *et al.*, 2009).

Figura 2 – Ilustração de um grafo representando as declarações RDF, no qual os sujeitos e objetos são vértices e os predicados são as arestas.

Figura 3 – Ilustração de um grafo representando as declarações RDF, identificando sujeitos, predicados e objetos por IRIs.

Assim como os sujeitos, os predicados também são identificados por IRIs. O predicado *type* é um tipo especial usado para agrupar recursos semelhantes, permitindo a existência de recursos transversais, ou seja, com mais de um predicado do tipo *type*. Por exemplo, o recurso

“Pedro” pode ser do tipo “Pessoa” e do tipo “PersonagemBiblico” em um determinado documento.

Outra característica importante a ser observada é o fato de ser mais simples mesclar duas estruturas baseadas em grafos, do que mesclar estruturas baseadas em árvores, como o XML (do Inglês, *eXtensible Markup Language*) (HEBELER *et al.*, 2009). No caso dos grafos RDF, essa tarefa é ainda mais fácil devido à desobrigação de traduzir os nomes dos recursos, pois são únicos em qualquer contexto. Além disso, os recursos são autônomos e independentes de qualquer outro, não importando a ordem em que são encontrados. Essas características favorecem o desenvolvimento de sistemas que exploram mais de uma fonte de informação.

Para que possa haver troca de informações entre aplicativos é necessário que ocorra uma serialização dos dados, convertendo-o de uma forma abstrata (grafo) para uma concreta (documento). A serialização pode ser realizada nos formatos RDF/XML², Turtle (Terse RDF Triple Language)³ e N-Triples⁴, sendo que todas as ferramentas RDF adotam pelo menos o RDF/XML (HEBELER *et al.*, 2009). O RDF/XML é uma sintaxe XML para representar grafos RDF. Esse formato não possui uma representação canônica, o que torna possível que ferramentas diferentes gerem serializações diferentes para um mesmo grafo. Assim, documentos RDF/XML aparentemente diferentes podem ser referentes ao mesmo grafo RDF, o que não ocorre em Turtle, por exemplo, visto que esse formato foi projetado especialmente para serialização de RDF e não precisa representar um grafo como uma árvore (HEBELER *et al.*, 2009).

3.1.2 RDF Schema

O RDF provê meios que possibilitam a modelagem das informações, porém isso não é suficiente para incorporar a semântica necessária aos recursos descritos. Faltam ainda conceitos referentes a classes e tipos de dados que não são representáveis pelo predicado *type*. A definição de classes possibilita informar relações do tipo hierárquicas e hereditárias, mas, acima de tudo, configurar restrições de existência e de relacionamentos, ou seja, pode-se evitar que relações sem sentido sejam declaradas (ANTONIOU; VANHARMELEN, 2004).

A linguagem RDF Schema (RDFS) possibilita a criação de um vocabulário, no qual pode-se declarar as classes dos recursos, e suas propriedades, que serão usadas no modelo RDF. Também é possível organizar as classes em estruturas hierárquicas, definir intervalo de valores para propriedades, declarar relações de herança e especificar tipos de dados aceitos como objeto em uma declaração (HEBELER *et al.*, 2009).

Contudo, segundo Antoniou e vanHarmelen (2004) o RDFS apresenta as seguintes limitações:

- escopo local de propriedades: a definição de uma restrição de intervalo para uma proprie-

² <http://www.w3.org/TR/rdf-syntax-grammar/>

³ <http://www.w3.org/TR/2014/REC-turtle-20140225/>

⁴ <http://www.w3.org/TR/2014/REC-n-triples-20140225/>

dade é válida para todas as classes, ou seja, não é possível atribuir intervalos diferentes para uma mesma propriedade em classes diferentes;

- classes disjuntas: não é possível declarar que duas subclasses não podem ser atribuídas ao mesmo recurso;
- combinações booleanas de classes: não é possível definir uma classe a partir da união ou intersecção de outras duas, nem mesmo como complemento de outra. Isso impede dizer que todos os mamíferos que voam são morcegos;
- restrição de cardinalidade: não é possível estabelecer um número mínimo (nem um número máximo) de valores distintos para uma propriedade, como dizer que uma disciplina tem no mínimo um professor e no máximo 30 alunos;
- características especiais de propriedades: não é possível definir relações do tipo transitivas (como “é menor que”), únicas (como “é pai de”) ou inversas (como “conduz” e “é conduzido por”).

Essas limitações podem ser supridas por uma linguagem de ontologia, que possui mais recursos que o RDFS.

3.2 Ontologia

O termo ontologia tem sua origem na Filosofia, sendo que uma das definições aceitas refere-se a um ramo da metafísica que estuda a natureza e relações dos seres. A palavra ontologia (do latim *ont*, verbo ser, + *logia*, estudo) remete ao estudo da existência, que foi uma área sistematicamente estudada por Aristóteles na tentativa de classificar as coisas no mundo (Guizzardi, 2005).

A ciência da computação e informação tem demonstrado especial interesse no uso de ontologias nas últimas décadas visando, principalmente, representar e compartilhar o conhecimento de domínios. As primeiras pesquisas na área de computação a incorporar o uso de ontologias foram de Inteligência Artificial, em meados de 1990 (Guizzardi, 2005). Porém, atualmente, muitas outras áreas buscam superar desafios com o auxílio de ontologias.

Ao longo dos anos, a definição de ontologia em computação foi sendo construída a partir de contribuições de diferentes pesquisadores até chegar-se a uma versão aceita por grande parte da comunidade científica e que abrange todas as características já relatadas: uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada (STUDER; BENJAMINS; FENSEL, 1998). Uma conceitualização refere-se a um modelo abstrato de algum fenômeno no mundo, por meio da identificação dos conceitos relevantes desse fenômeno. Essa deve ser: explícita, pois os conceitos usados e as restrições de uso devem se definidas

explicitamente; formal, de modo a ser interpretada por máquina; e compartilhada, pois deve refletir o consenso de um grupo de pessoas sobre um determinado conhecimento.

Definido o conceito de ontologia, na sequência será apresentada a linguagem que possui os recursos necessários para modelar ontologias atendendo às exigências supracitadas, a OWL (do Inglês, *Web Ontology Language*).

3.2.1 OWL 2

A OWL é a linguagem usada para definir ontologias de maneira a representar a semântica de um domínio de conhecimento. Ela estende o vocabulário do RDFS com recursos adicionais que possibilitam a construção de ontologias mais complexas para a Web (HEBELER *et al.*, 2009). São adicionadas restrições em relação à estrutura e conteúdo de documentos RDF que possibilitam resolver, computacionalmente, problemas como:

- membros de uma classe: se x é uma instância da classe A , e A é uma subclasse de B , então x também é uma instância de B ;
- equivalência de classes: se a classe A é equivalente à classe B , e B é equivalente à classe C , então A também é equivalente à C ;
- consistência: dadas as seguintes afirmações: x é uma instância de A ; A é subclasse de $B \cap C$, A é subclasse de D ; e B e D são classes disjuntas. Tem-se uma inconsistência, pois A é um conjunto vazio;
- classificação: se um par propriedade-valor é condição suficiente para classificar uma classe A , um elemento x que satisfaça tal condição deve ser classificado como instância de A .

A OWL foi desenvolvida sobre a RDF e RDFS de modo que:

- Todos os tipos de OWL usam a sintaxe RDF;
- Instâncias são declaradas como em RDF; e
- Construtores OWL como `owl:Class`, `owl:DatatypeProperty` e `owl:ObjectProperty` são especializações das classes homólogas em RDF, conforme demonstrado graficamente na Figura 4.

A OWL 2, publicada em 2012, é uma extensão e revisão da OWL, referenciada a partir daqui como OWL 1. A compatibilidade com a OWL 1 é completa, ou seja, todas ontologias OWL 1 são ontologias OWL 2 válidas, com inferências idênticas em todos os casos práticos (OWL..., 2012).

Figura 4 – Relacionamentos entre as subclasses OWL e RDF/RDFS. Extraída de [Antoniou e vanHarmelen \(2004\)](#).

A OWL 1 definiu dois dialetos principais, OWL DL (Descrição Lógica, do Inglês, *Description Logic*) e OWL Full, e um subconjunto sintático (OWL Lite) ([ANTONIOU; VANHARMELLEN, 2004](#)). No entanto, verificou-se que isso não era suficiente para abordar requisitos posteriormente identificados por implantações de ontologias OWL, como escalabilidade, desempenho na resolução de inferências, interoperabilidade com Bancos de Dados Relacionais e interoperabilidade direta com triplas RDF ([GOLBREICH; WALLACE, 2012](#)).

Com o objetivo de atender à essa demanda, a OWL 2 define três subconjuntos sintáticos diferentes com propriedades computacionais ou possibilidades de implementação úteis ([GRAU et al., 2012](#)):

- OWL 2 EL (Linguagem Existencial, do Inglês, *Existential Language*): é baseada na EL++, uma linguagem formal de representação de conhecimento, e é particularmente usada em aplicações que empregam ontologias que contêm um número grande de propriedades ou classes. Esse subconjunto consegue explorar toda a carga semântica de uma ontologia e realizar inferências lógicas em tempo polinomial em relação ao tamanho da ontologia. Algoritmos de raciocínio específicos para esse subconjunto estão disponíveis e demonstraram ser implementáveis de forma altamente escalável;
- OWL 2 QL (Linguagem de Consulta, do Inglês, *Query Language*): é destinada a aplicativos que usam volumes muito grandes de dados de instância e onde a resposta às consultas é a tarefa de raciocínio mais importante. Na OWL 2 QL, a resposta de consultas conjuntivas podem ser implementadas usando sistemas de banco de dados relacionais convencionais. Esse subconjunto é bem mais limitado que o OWL 2 EL, apesar de incluir a maioria dos principais recursos de modelos conceituais como diagramas de classes UML e diagramas ER;
- OWL 2 RL (Regras de Linguagem, do Inglês, *Rule Language*): é destinada a aplicações que requerem escalabilidade de raciocínio sem muita perda de recursos da linguagem.

Esse subconjunto foi projetado para satisfazer aplicativos OWL 2 que podem trocar a expressividade total da linguagem pela eficiência na resposta. Os sistemas OWL 2 RL podem ser implementados usando mecanismos de raciocínio baseados em regras de linguagem.

Devido a todas as características apresentadas, a OWL é comumente utilizada para modelar as ontologias de bases de conhecimento, conforme descrito na próxima subseção.

3.2.2 Bases de Conhecimento

Bases de conhecimento são aplicações da Web Semântica que armazenam e recuperam informações descritas no formato RDF, assim como realizam interpretações semânticas definidas por OWL. Duas das bases de conhecimento mais conhecidas são a GeoNames⁵ e a DBpedia⁶.

GeoNames é uma base de conhecimento comunitária disponível para *download* que integra nomes de lugares, relevo, população entre outras características que podem ser adicionadas ou editadas por meio de uma interface amigável⁷ (AHLERS, 2013). Todas as coordenadas de latitude e longitude são descritas usando o padrão de coordenadas WGS84 (Sistema Geodésico Mundial 1984, do Inglês, *World Geodetic System* 1984).

DBpedia é um conjunto de bases de conhecimento resultantes de um projeto comunitário que extrai conhecimento em múltiplos idiomas das 111 diferentes edições da Wikipedia, e disponibiliza gratuitamente todas as bases para *download* e o acesso a 14 delas para consulta (LEHMANN *et al.*, 2015). Sua ontologia⁸ descreve classes como “write”, “musical work” e “book”, e propriedades como “birth place”, “running time” e “author” (MORSEY *et al.*, 2012). A principal vantagem da DBpedia é sua vasta cobertura sobre diversos domínios, na qual suas informações RDF apontam para IRIs de outras bases de conhecimento. Por sua vez, essa característica faz com que, cada vez mais, outras bases de conhecimento apontem para IRIs da DBpedia. Consequentemente, a DBpedia ocupa hoje uma posição de destaque na *Linked Data*.

Para que os objetivos propostos pela Web Semântica sejam alcançados, é necessário que, além da existência de bases de conhecimento, exista uma conexão entre elas, conhecida como *Linked Data*⁹. Ou seja, *Linked Data* é um conjunto de bases de conhecimento inter-relacionadas na Web. Desse modo é possível utilizar *Linked Data* para encontrar recursos que representem as entidades retratadas em um sistema. Além disso, é possível obter as informações desejadas por meio de consultas a *endpoints*, que são interfaces de acesso às bases de conhecimento inter-relacionadas (BAUER; KALTENBÖCK, 2012).

⁵ <http://www.geonames.org/>

⁶ <http://dbpedia.org/About>

⁷ <http://www.geonames.org/maps/cities.html>

⁸ http://mappings.dbpedia.org/index.php/Main_Page

⁹ <http://www.w3.org/standards/semanticweb/data>

Um SR pode, por exemplo, encontrar recursos em uma base de conhecimento que representem itens do seu acervo e explorar a estrutura da ontologia, sob a qual os recursos estão organizados, para encontrar relacionamentos entre tais itens.

Para se realizar consultas em bases de conhecimento, é necessário usar uma linguagem que reconheça fundamentalmente a sintaxe utilizada nos documentos RDF e, de acordo com a necessidade, dê suporte a recursos disponíveis em linguagens baseadas em RDF, como a OWL. SPARQL, um acrônimo recursivo para Protocolo SPARQL e Linguagem de Consulta RDF (do Inglês, *SPARQL Protocol and RDF Query Language*), é uma recomendação W3C¹⁰ para realizar consultas em sistemas que exploram Web Semântica. Existem outras linguagens como RDQL (do Inglês, *RDF Data Query Language*) e SeRQL (do Inglês, *Sesame RDF Query Language*) (HEBELER *et al.*, 2009), mas a linguagem SPARQL é uma recomendação W3C e a mais utilizada nos *endpoints* disponíveis. *Endpoint* é um serviço, não necessariamente Web, que aceita e processa consultas SPARQL, e retorna o resultado em diferentes formatos, dependendo do formato da consulta. Apesar de ser uma linguagem de consulta e um protocolo, o foco da maioria dos usuários está na sintaxe da linguagem para realizar consultas. O protocolo descreve como um cliente SPARQL (um navegador, por exemplo) conversa com um *endpoint* SPARQL (como <http://dbpedia.org/sparql>). O código listado na Figura 5 é um exemplo de consulta SPARQL. Consultas SPARQL específicas usando predicados como `rdfs:subClassOf`

```
SELECT ?location
WHERE {
    ?person <http://www.w3c.org/2000/01/rdf-schema#label>
        "George Washington"@en.
    ?location <http://dbpedia.org/property/namedFor> ?person
}
```

Figura 5 – Consulta SPARQL para obter lugares com nomes em homenagem a George Washington. Extraído de Hebel *et al.* (2009)

ou `rdfs:domain of` para um determinado conjunto de itens, por exemplo, podem ser realizadas com o objetivo de explorar a estrutura hierárquica definida pela ontologia para esse conjunto e calcular a similaridade semântica entre recursos distintos a partir dos resultados. Na próxima subseção serão apresentados dois métodos de calculá-la.

3.3 Similaridade Semântica

Os termos relacionamento semântico, similaridade e distância semântica são bastante usados em NLP (Processamento de Linguagem Natural, do Inglês, *Natural Language Processing*), e ultimamente seus conceitos têm despertado um crescente interesse dos pesquisadores da área de SRs para comparar itens e usuários (MENG; HUANG; GU, 2013). Contudo, é importante compreender a diferença entre os três conceitos e usá-los de maneira adequada. Ao se discutir o

¹⁰ http://www.w3.org/2009/sparql/wiki/Main_Page

nível de relacionamento semântico e a similaridade entre duas palavras, por exemplo, pode-se confundir os dois conceitos, visto que em muitos casos um nível de relacionamento semântico alto implicará em uma similaridade alta e vice-versa. Porém, quando analisamos palavras antônimas, observamos um alto relacionamento semântico e uma baixa similaridade. Um outro exemplo ocorre quando observamos um alto relacionamento semântico entre as palavras autor e livro, mas uma similaridade menor que a existente entre autor e compositor. Portanto, temos que similaridade é um conceito mais específico de relacionamento semântico.

Dada uma rede de conexões entre recursos, a distância semântica é calculada para mostrar o quanto dois recursos são próximos e pode ser usada tanto no cálculo do relacionamento semântico quanto no de similaridade entre eles.

Um exemplo de rede de conexões entre recursos é a WordNet¹¹, um enorme banco de dados léxico, no qual substantivos, verbos, adjetivos e advérbios estão agrupados em conjuntos de sinônimos cognitivos, que expressam conceitos distintos. Por meio dessa rede é possível encontrar relacionamentos de hipernímia e hiponímia, por exemplo, entre conceitos. Dentre as diferentes técnicas para se calcular a similaridade entre conceitos na WordNet, serão apresentadas nas próximas seções as que são baseadas em caminho, visto que serão usadas como baseline na comparação com o desempenho da proposta deste trabalho, que também é baseada em caminho.

Além das métricas de similaridade baseadas na WordNet, as subseções seguintes também apresentarão métricas de distância semântica para dados conectados.

3.3.1 Métrica Baseada no Menor Caminho

Essa métrica de similaridade é baseada na WordNet e considera apenas o menor caminho, dentre os possíveis na estrutura hierárquica, entre dois conceitos para calcular a similaridade entre eles. Na WordNet, denomina-se conceito um conjunto de sinônimos cognitivos composto por substantivos, verbos, adjetivos e advérbios. Visto que a grandeza distância remete a uma métrica, o cálculo da similaridade, $sim_{path}(c_1, c_2)$, entre dois conceitos, c_1 e c_2 , pode partir do cálculo da distância entre eles, $len(c_1, c_2)$, contabilizando o número de arestas que os separam na hierarquia, e considerando a similaridade dependente dessa distância. Assim, a similaridade entre dois conceitos depende do quão próximos eles estão na estrutura taxonômica (MENG; HUANG; GU, 2013):

$$sim_{path}(c_1, c_2) = 2 * deep_max - len(c_1, c_2) \quad (3.1)$$

onde $deep_max$ é a profundidade máxima da árvore que representa a hierarquia da estrutura taxonômica, que será um valor fixo diferente para cada nova estrutura.

Analisando a Equação 3.1, pode-se concluir que o valor da similaridade estará entre 0 e $2 * deep_max$. A similaridade terá seu valor máximo quando a distância entre os conceitos

¹¹ <https://wordnet.princeton.edu/>

for igual a 0, ou seja, quando $c_1 = c_2$, e terá seu valor mínimo quando a distância for igual a $2 * deep_max$. Usando essa métrica para calcular a similaridade entre os conceitos representados na Figura 6 pode-se concluir que, $len(\text{correio}, \text{veículo}) = len(\text{veículo automotivo}, \text{bicicleta}) = 2$, portanto, $sim_{path}(\text{correio}, \text{veículo}) = sim_{path}(\text{veículo automotivo}, \text{bicicleta})$. Porém, visualmente, observa-se uma similaridade maior entre veículo automotivo e bicicleta do que entre correio e veículo. Isso ocorre porque essa métrica baseia-se apenas na distância entre os conceitos para calcular a similaridade entre eles.

Figura 6 – Fragmento da estrutura de relação é-um da WordNet. Adaptada de Meng, Huang e Gu (2013)

3.3.2 Métrica Leacock & Chodorow

A métrica de similaridade apresentada por Leacock e Chodorow (1998), assim como a anterior, é baseada na WordNet e usa apenas a profundidade máxima da árvore que representa a hierarquia e o tamanho do menor caminho entre dois conceitos para calcular a similaridade entre eles. Contudo, o uso da função $\log()$ na equação garante um comportamento diferenciado no gráfico da equação completa. É possível observar que para caminhos cujos tamanhos são próximos do máximo, a punição no valor da similaridade é atenuada conforme os caminhos aumentam. Porém, o ganho no valor da similaridade nos caminhos mais próximos de 0 é acentuado à medida que os caminhos diminuem. Tal característica pode ser observada na Equação 3.2:

$$sim_{LC}(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2 * deep_max} \quad (3.2)$$

É comum adicionar 1 ao numerador ($len(c_1, c_2)$) e ao denominador ($2 * deep_max$) da fração para evitar $\log(0)$. Assim, os valores de $sim_{LC}(c_1, c_2)$ ficam no intervalo $(0, \log(2 * deep_max + 1)]$ (MENG; HUANG; GU, 2013).

Comparando a métrica do menor caminho com a porposta por Leacock e Chodorow (1998) no fragmento de estrutura representado na Figura 6, tem-se

$$sim_{path}(\text{veículo automotivo}, \text{bicicleta}) = 6$$

e

$$\text{sim}_{\text{path}}(\text{bicicleta}, \text{ônibus}) = 3,$$

enquanto que

$$\text{sim}_{LC}(\text{veículo automotivo}, \text{bicicleta}) = 0,6$$

e

$$\text{sim}_{LC}(\text{bicicleta}, \text{ônibus}) = 0,2.$$

Contudo, essa métrica também baseia-se apenas na distância entre os conceitos para calcular a similaridade entre eles e pode não revelar diferenças no cálculo da similaridade entre diferentes pares de conceitos. Ou seja, usando essa métrica também se tem $\text{sim}_{LC}(\text{correio}, \text{veículo}) = \text{sim}_{LC}(\text{veículo automotivo}, \text{bicicleta})$.

3.3.3 Métrica Wu & Palmer

Wu e Palmer (1994) propuseram uma métrica de similaridade baseada na WordNet que, além de considerar o tamanho do menor caminho entre dois conceitos, c_1 e c_2 , leva em consideração a profundidade do conceito mais específico que os abrange, $\text{depth}(\text{lso}(c_1, c_2))$, na estrutura hierárquica da taxonomia (Equação 3.3):

$$\text{sim}_{WP}(c_1, c_2) = \frac{2 * \text{depth}(\text{lso}(c_1, c_2))}{\text{len}(c_1, c_2) + 2 * \text{depth}(\text{lso}(c_1, c_2))} \quad (3.3)$$

Note que, se o ancestral comum entre dois conceitos, c_1 e c_2 , for a raiz da estrutura, $\text{depth}(\text{lso}(c_1, c_2)) = 1$ e $\text{sim}_{WP}(c_1, c_2) > 0$. Quando c_1 e c_2 forem iguais, $\text{lso}(c_1, c_2)$ será o mesmo conceito e $\text{sim}_{WP}(c_1, c_2) = 1$. Contudo, nos casos em que c_1 é diferente de c_2 , tem-se $0 < \text{depth}(\text{lso}(c_1, c_2)) < \text{deep_max}$, $0 < \text{len}(c_1, c_2) < 2 * \text{deep_max}$ e $0 < \text{sim}_{WP}(c_1, c_2) < 1$ (MENG; HUANG; GU, 2013).

A profundidade do ancestral comum mais próximo adiciona à função a informação referente à posição dos conceitos na estrutura e garante uma similaridade maior entre conceitos com profundidade maior. Assim, ao usar a Equação 3.3 para calcular $\text{sim}_{WP}(\text{correio}, \text{veículo})$ e $\text{sim}_{WP}(\text{veículo automotivo}, \text{bicicleta})$, conforme a Figura 6, os resultados mostrarão que $\text{sim}_{WP}(\text{veículo automotivo}, \text{bicicleta}) > \text{sim}_{WP}(\text{correio}, \text{veículo})$.

3.3.4 Distância Semântica em Dados Conectados

No que diz respeito a Sistemas de Recomendação, pesquisas recentes têm explorado ontologias de domínios específicos a fim de obterem descrições de itens mais detalhadas e com maior carga semântica para representar os itens do acervo e os perfis de usuários, principalmente em sistemas baseados em conteúdo (LOPS; GEMMIS; SEMERARO, 2011). Além disso, os relacionamentos entre recursos de uma ontologia podem ser medidos por meio do cálculo da distância semântica (PASSANT, 2010b). Uma vez calculada a distância semântica, pode-se dizer

quão similares são dois recursos. A similaridade entre dois recursos pode, ainda, ser calculada por meio da distância de edição de árvores (BILLE, 2005), onde a estrutura das características de um recurso em uma ontologia pode ser manipulada como uma árvore.

Similaridade e distância semântica são métricas definidas para um conjunto de termos, ou recursos, onde a ideia de distância entre eles é baseada na semelhança de seus conteúdos semânticos: recursos com conteúdo semântico similar têm distâncias pequenas, enquanto que recursos com conteúdo semântico dissimilares têm distâncias grandes (PASSANT, 2010b). Concretamente, similaridade semântica pode ser estimada por similaridades topológicas entre termos (recursos) em uma ontologia. Ela pode ser utilizada como um modo de calcular o relacionamento existente entre dois recursos.

A distância semântica é útil em diversos estudos das mais diversas áreas de pesquisa, como para buscar semelhanças entre genes de doenças na biomedicina (PESQUITA *et al.*, 2009), por exemplo. Em sistemas de recomendação o cálculo da distância semântica entre recursos é muito útil, pois, dado um item de referência, eles buscam itens similares no acervo para recomendar ao usuário. A LDSD (Distância Semântica em *Linked Data*, do Inglês, *Linked Data Semantic Distance*) apresentada por Passant (2010b) depende apenas dos *links* entre os recursos para calcular a distância entre eles, mas possui variações dependendo se a ligação é direta ou indireta e se serão atribuídos pesos diferentes a elas dependendo da frequência com que o tipo de ligação ocorre no conjunto de dados. A distância calculada varia entre 0 e 1, sendo que os recursos são mais próximos conforme esse valor se aproxima de 0. Primeiramente é necessário contabilizar os *links* diretos e distintos entre os recursos, que são representados por uma função C_d . $C_d(l_i, r_a, r_b)$ será igual a 1 se existir um *link* do tipo l_i entre os recursos r_a e r_b , e 0 caso contrário. No exemplo ilustrado na Figura 7, tem-se: $C_d(a:l_1, a:r_1, a:r_2) = 1$ e $C_d(a:l_2, a:r_1, a:r_3) = 1$, entre outros. Após contabilizar os *links* diretos, pode-se medir a distância direta pela Equação 3.4 (PASSANT, 2010b):

$$LDS D_d(r_a, r_b) = \frac{1}{1 + C_d(n, r_a, r_b) + C_d(n, r_b, r_a)} \quad (3.4)$$

De maneira semelhante, pode-se contabilizar os *links* indiretos e distintos entre os recursos, por meio das funções C_{io} e C_{ii} . $C_{io}(l_i, r_a, r_b)$ será igual a 1 se existir um *link* do tipo l_i que saia de r_a para um recurso n e outro *link* do tipo l_i que saia de r_b para o mesmo recurso n ; caso contrário, será igual a 0. $C_{ii}(l_i, r_a, r_b)$ será igual a 1 se existirem dois *links* do tipo l_i saindo de um recurso n , um para r_a e outro para r_b , caso contrário será igual a 0. No exemplo ilustrado na Figura 7, temos: $C_{io}(a:l_2, a:r_3, a:r_4) = 1$ (por meio de $a:r_5$). Na sequência, a distância indireta é calculada pela Equação 3.5 (PASSANT, 2010b):

$$LDS D_i(r_a, r_b) = \frac{1}{1 + C_{io}(n, r_a, r_b) + C_{ii}(n, r_a, r_b)} \quad (3.5)$$

Passant (2010b) ainda mostra como combinar os cálculos da distância direta e indireta para medir a distância semântica entre dois recursos.

Figura 7 – Ilustração de um grafo representando os recursos vinculados (*Linked Data*), onde $a:r_1$, $a:r_2$, $a:r_3$, $a:r_4$ e $a:r_5$ são recursos (vértices do grafo) e $a:l_1$ e $a:l_2$ são links (arestas do grafo)

Em um sistema de recomendação, o cálculo da distância semântica entre recursos é importante para retornar a lista de recomendações aos usuários, ordenada pela semelhança dos itens retornados em relação ao item dado como referência. Ou seja, quanto mais semelhante, mais relevante é o item retornado em uma determinada pesquisa.

É importante que se escolha um conjunto de dados bastante representativo para o domínio trabalhado, pois não é possível calcular a distância semântica entre dois recursos usando as equações apresentadas aqui, se um deles não for encontrado no conjunto de dados escolhido. Eventualmente, pode-se trabalhar com mais de um conjunto de dados, mas se mesmo assim o recurso não for encontrado em um dos conjuntos de dados, não será possível usufruir da contribuição apresentada neste trabalho para refinar a lista retornada pelo sistema de recomendação.

Para buscar relacionamentos mais específicos, que podem ser inferidos a partir da análise hierárquica de uma estrutura, uma abordagem mais específica pode ser utilizada. A estrutura de classes e subclasses de uma ontologia pode ser comparada a uma estrutura de árvore rotulada não ordenada com raiz, que por definição é uma árvore que possui um vértice raiz, cujos vértices são rotulados e somente a relação pai-filho entre eles é significativa, ou seja, a ordem entre os vértices irmãos não importa (ZHANG, 1996). Devido ao interesse específico nesse tipo de árvore, nessa seção ele será citado por, simplesmente, árvores, exceto quando for mencionado explicitamente o contrário.

A partir dessa comparação, é possível manipular a estrutura hierárquica de um item (formada por suas classes e respectivas classes pais, por exemplo) como uma árvore e calcular sua similaridade com a estrutura hierárquica de outro item, segundo uma ontologia diferente ou não, por meio da distância de edição de árvores. O cálculo da distância de edição de árvores é baseado no custo de cada operação necessária para transformar uma árvore T_1 em outra árvore

T_2 . Segundo Bille (2005), as operações que podem ocorrer na transformação de uma árvore T_1 em uma árvore T_2 , sendo $T_1 \neq T_2$, são as seguintes:

- **Substituição:** alterar o rótulo de um vértice v em T_1 ;
- **Remoção:** remover um vértice v , diferente do vértice raiz e com pai v' , em T_1 , de modo que os filhos de v se tornem filhos de v' ;
- **Inserção:** inserir um vértice v como filho de um vértice v' em T_1 , de modo que o vértice v se torne o pai dos filhos de v' .

Visto que cada operação de edição e_i possui um custo c_{e_i} e que um *script* de edição S , composto pela sequência de e_i necessárias para transformar T_1 em T_2 , possui custo equivalente à soma dos c_{e_i} dos respectivos $e_i \in S$, o S ótimo é aquele que apresenta menor custo. Assim, o valor da distância de edição de árvores (DEA) entre T_1 e T_2 , será o custo do S ótimo entre T_1 e T_2 (BILLE, 2005) (Equação 3.6):

$$DEA = \min_S \sum_{e_i \in S} c_{e_i} \quad (3.6)$$

O cálculo da distância de edição de árvores pode ser aplicado em sistemas de recomendação com o objetivo de encontrar itens ou usuários semelhantes a partir da estrutura de categorias que definem um item ou os interesses de um usuário, por exemplo.

3.4 Trabalhos relacionados

Nesta seção são apresentados trabalhos que estão relacionados com o proposta apresentada neste trabalho por aplicarem um ou mais conceitos desenvolvidos na área de Web Semântica em sistemas de recomendação, a fim de alcançar melhor desempenho com o uso de informações semânticas.

Um aplicativo para dispositivos móveis que recomenda filmes e peças de teatro, o Cinemappy, é apresentado no trabalho de Ostuni *et al.* (2013). Esse aplicativo refina as recomendações geradas por um sistema de recomendação baseado em conteúdo a partir de informações contextuais relacionadas a posição espacial e temporal do usuário. O módulo que desempenha a função de sistema de recomendação é alimentado com informações extraídas da DBpedia para construir as descrições dos filmes. Então, a similaridade entre os filmes e o perfil do usuário é calculada usando-se vetores de características. O sistema ainda coleta dados do site Trovacinema¹² sobre cinemas, teatros e suas respectivas programações, e a geolocalização dos usuários por meio da Google Geocoding API (Interface de Programação de Aplicativos, do Inglês, *Application Programming Interface*)¹³. Com essas informações em mãos, filmes e peças de teatro em cartaz são recomendados ao usuário, conforme seus interesses, posição atual e horário.

¹² <http://trovacinema.repubblica.it/>

¹³ <https://developers.google.com/maps/documentation/geocoding/>

Passant (2010a) descreve um sistema de recomendação de músicas, dbrec, que recomenda músicas de artistas e bandas semelhantes às preferências dos usuários utilizando uma abordagem baseada em conteúdo e uma ontologia para a descrição dos artistas e bandas. Os artistas e bandas relacionados no sistema são definidos por entidades descritas no DBpedia, via uma URI. Assim, para retornar uma lista de recomendações ao usuário, o sistema calcula a similaridade entre os artistas, ou bandas, de preferência do usuário e os demais disponíveis no sistema explorando as propriedades e relacionamentos observados na ontologia da DBpedia. A similaridade entre artistas é calculada por meio da distância semântica e o sistema retorna uma lista ranqueada com as músicas dos artistas e bandas mais semelhantes às preferências do usuário.

Noia *et al.* (2012) criaram um sistema de recomendação baseado em conteúdo, testado sob o domínio de filmes, que usa bases de conhecimento públicas inter-relacionadas, a conhecida *Linked Open Data* (LOD)¹⁴. O trabalho propõe o modelo espaço vetorial para calcular a similaridade entre os recursos RDF encontrados nas bases de conhecimento DBpedia, LinkedMDB¹⁵ e Freebase¹⁶. O uso desses recursos RDF tornou mais fácil a realização da fase de pré-processamento das informações sobre os itens (extração, desambiguação, expansão das palavras-chave, discutidas no trabalho), visto que os dados já estão estruturados em uma ontologia. Também são apresentadas maneiras de detectar similaridade entre recursos explorando se o grafo RDF, como pelas propriedades *dbpedia-owl:subsequentWork* e *dbpedia-owl:previousWork*, que indicam que um filme é sequência de outro, ou quando recursos diferentes possuem algumas triplas RDF iguais, entre outros indicadores.

O artigo publicado por Jntema *et al.* (2010) descreve o *framework* Athena, extensão do *framework* Hermes, que fornece vários métodos para sistemas de recomendação baseados em conteúdo focados em notícias. O sistema possibilita construir perfis de usuários, construir descrições de itens (notícias) e disponibiliza várias medidas de similaridade. O *framework* conta com uma ontologia específica para estruturar conceitos e relacionamentos, descritos por triplas RDF, armazenados na base de conhecimento. Essa base de conhecimento é construída e mantida por um especialista, a partir de informações do Yahoo! Finance¹⁷. O artigo também apresenta os resultados obtidos na comparação de desempenho de sistemas de recomendação baseados em conteúdo, com implementações diferentes de medidas de similaridade. Os resultados apontam que houve uma maior satisfação dos usuários em relação às recomendações oriundas do sistema com métricas de similaridade baseada em conceitos do que do sistema com métrica de similaridade baseada em TF-IDF (Frequência do Termo–Inverso da Frequência nos Documentos, do Inglês, *Term Frequency–Inverse Document Frequency*), por exemplo.

Maidel *et al.* (2008) apresentam um sistema de recomendação baseado em conteúdo que seleciona notícias eletrônicas, o ePaper. A filtragem baseada em conteúdo do sistema utiliza

¹⁴ <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

¹⁵ <http://www.linkedmdb.org/>

¹⁶ <https://www.freebase.com/>

¹⁷ <http://finance.yahoo.com/>

a hierarquia de uma ontologia específica para itens de notícias. Nele, um conceito presente no perfil de um usuário e outro presente na descrição de um item, são considerados comuns, de acordo com a distância hierárquica medida entre os dois conceitos. Baseado no número de conceitos comuns e suas respectivas distâncias, a similaridade entre itens e usuários é calculada e as notícias são ranqueadas de acordo com a relevância para cada usuário. Esse trabalho mostra que é possível alcançar melhorias significativas explorando-se a distância hierárquica entre conceitos em uma ontologia, ao invés de usar apenas a similaridade entre conceitos comuns.

O trabalho desenvolvido por [Harispe et al. \(2013\)](#) foi motivado pela necessidade de sistemas de recomendação baseados em conteúdo em mensurar a distância semântica entre dois recursos definidos em bases de conhecimento RDF. [Harispe et al. \(2013\)](#) apresentam um *framework* para tal finalidade. O *framework* se baseia na criação de uma classe chamada contexto de projeção, da qual são geradas instâncias para cada recursos utilizados. O contexto de projeção é um conjunto de projeções, que por sua vez é um conjunto propriedades relevantes para definir uma informação, tida como importante, para o cálculo da distância entre recursos. O *framework* foi avaliado por meio da implementação de um sistema de recomendação de músicas baseado em conteúdo, no qual o contexto de projeção é composto por três projeções: i) seus nomes; ii) seus tipos (classes); iii) a distância de seus lugares de formação; e iv) a proximidade dos gêneros relacionados. Uma métrica diferente pode ser utilizada para o cálculo de similaridade em cada projeção, para serem combinadas posteriormente e apresentar um valor final. Desse modo, o *framework* possibilita um cálculo de similaridade que vai além dos aplicados nos sistemas estudados até então.

A pesquisa apresentada por [Ostuni et al. \(2014\)](#) ataca o problema relacionado a dificuldade de extrair características semânticas dos itens do acervo e o problema da sobre-especialização, inerentes à SRs Baseados em Conteúdo, explorando Dados Abertos Conectados. Esse trabalho sugere a construção de um grafo a partir dos dados extraídos de bases de conhecimento para explorá-lo e identificar relacionamentos entre os itens do acervo, que estão representados nesse grafo. Nesse sistema, cada item possui um grafo local. Isso possibilita a comparação entre itens a partir da contabilização ponderada das entidades comuns em seus respectivos grafos, e o aprendizado do perfil dos usuários para lhes recomendar itens alinhados aos seus interesses com maior nível de satisfação e novidade.

[Zhang et al. \(2016\)](#) apresentam um *framework*, CKE (do Inglês, *Collaborative Knowledge Base Embedding*), que extrai dados heterogêneos de uma base de conhecimento e os modela de modo a gerar representações semânticas dos itens do acervo para usá-las em um sistema de recomendação colaborativo incorporado, que aprende as representações latentes. O *framework* explora, em um primeiro momento, a base de conhecimento e constrói a representação semântica dos itens a partir dos conteúdos estrutural, textual e visual encontrados. Essa representação semântica é composta de três vetores independentes, um para cada tipo de conteúdo extraído da base. No passo seguinte, o sistema de filtragem colaborativa constrói o

vetor de características latentes de cada item a partir de uma combinação dos vetores da sua representação semântica. Então, o vetor de características latentes passa por um processo de aprendizagem das variáveis da equação para, enfim, realizar a predição de notas para cada item.

O trabalho apresentado por [Rowe \(2014\)](#) mostra que as abordagens comuns de sistemas de recomendação não são capazes de lidar com eventuais mudanças nas preferências dos usuários. Esse trabalho apresenta o modelo SemanticSVD⁺⁺, que é baseado em fatoração de matrizes, mas ameniza a limitação supracitada e inerente ao modelo tradicional usando categorias semânticas para inferir a afinidade de um determinado usuário por fatores latentes. O modelo é capaz de analisar a evolução das preferências dos usuários ao longo do tempo, sob uma perspectiva semântica, e verificar o como tais mudanças são influenciadas por fatores dinâmicos globais. Dessa maneira é possível observar como as preferências de um usuário evoluem em relação ao seu próprio histórico e em relação às preferências globais. Foram observadas melhoras significativas na comparação com o modelo SVD⁺⁺ (do Inglês, *Singular Value Decomposition++*) ([KOREN, 2008](#)).

3.5 Considerações finais

Ao longo deste capítulo foram discutidas as sucessivas evoluções na busca em agregar semântica às representações de dados na Web. O estudo do estado da arte mostra uma grande evolução na área, porém as soluções mais completas e atuais nem sempre são fáceis ou viáveis de se implementar. São poucas as ontologias modeladas a partir da linguagem OWL full devido à sua complexidade e custo computacional.

Ao analisar os trabalhos mencionados na seção anterior, que representam parte das pesquisas atuais, é possível observar aplicações ou combinações de técnicas ainda não exploradas. Um sistema de recomendação capaz de extrair todo o potencial em informação disponibilizado por bases de conhecimento modeladas por OWL e que represente os perfis dos usuários em documentos RDF, pode alcançar ganhos significativos em relação à satisfação dos usuários. Devido ao grande nível de detalhamento que pode ser definido por uma OWL, sistemas com tais características podem calcular mais precisamente a similaridade entre descrições de itens e perfis de usuários, entre perfis de usuários, e entre descrições de itens. A modelagem de uma regra que represente a ideia de oposição entre recursos, por exemplo, pode ajudar a eliminar itens de uma lista de recomendações. Contudo, a comparação aprofundada e o cálculo da distância com mais detalhamentos entre recursos descritos em ontologias diferentes ainda é um grande desafio.

Pode-se ainda explorar a estrutura hierárquica das categorias semânticas usadas para categorizar recursos em uma ontologia, a fim de encontrar relacionamentos semânticos entre eles. Isso possibilita uma análise profunda dos relacionamentos e a atribuição de pesos a cada um deles, dependendo da distância semântica calculada em cada relacionamento. Este trabalho explora essa alternativa, não pesquisada nos trabalhos mencionados acima, com o propósito

de beneficiar o cálculo de similaridade entre itens em sistemas de recomendação. O próximo capítulo descreve como o cálculo de similaridade entre recursos pode ser baseado na distância semântica entre as categorias semânticas que os classificam.

PROPOSTA DE SOLUÇÃO

Os capítulos anteriores apresentaram conceitos e técnicas sobre sistemas de recomendação e dados abertos conectados que fornecem suporte à proposta de solução apresentada neste capítulo. A proposta visa, basicamente, propor o uso da ontologia de uma ou mais bases de conhecimento em sistemas de recomendação para se construir a representação dos itens disponíveis com maior carga semântica. Consequentemente, será possível alcançar melhor desempenho ao comparar itens entre si devido ao alto nível de detalhes disponíveis. Um bom desempenho na comparação de itens também favorece o uso de algoritmos como o k -NN, mencionado na Seção 2.1.2.1 do Capítulo 2. Adicionalmente, pretende-se beneficiar a recomendação de novos itens, situação caracterizada como partida fria, conforme mencionado no Capítulo 1. A proposta é adaptável para sistemas de qualquer domínio, bastando apenas encontrar uma base de conhecimento adequada.

A Seção ?? aponta alguns desafios inerentes à área de sistemas de recomendação e destaca a justificativa para este trabalho. Na Seção ?? são enumerados os objetivos que este trabalho pretende atingir. A arquitetura do sistema é apresentada em detalhes na Seção 4.1, na qual também são discutidas as técnicas que poderão ser empregadas na construção do sistema. Por fim, a Seção 4.2 encerra este capítulo destacando as características principais da proposta.

4.1 Arquitetura

A arquitetura proposta para cumprir os objetivos mencionados na seção anterior é ilustrada na Figura 8. Nessa proposta, uma lista de recomendações pode ser gerada para um usuário após sete tarefas, basicamente, que estão rotuladas de “a” a “g” na Figura 8.

Figura 8 – Visão geral da arquitetura do sistema de recomendação proposto.

4.1.1 Base de Dados

O primeiro passo é criar uma base de dados, seja um arquivo ou um Sistema de Gerenciamento de Banco de Dados (SGBD), para armazenar os dados dos itens do acervo e dos usuários do sistema. É recomendado o uso de um SGBD para se armazenar as descrições sintáticas a fim de simplificar a manipulação de dados quando for necessário recuperá-los ou atualizar os registros dos itens com suas respectivas IRIs e outros dados obtidos de bases de conhecimento. Além desses dados, também será necessário armazenar os valores de similaridade entre itens, que serão calculados posteriormente. Conforme apresentado na Seção 2.1.2 do Capítulo 2, um sistema de recomendação com filtragem colaborativa baseada em vizinhança de itens recomenda itens a um determinado usuário que sejam semelhantes aos itens bem avaliados por ele e desconhecidos pelo mesmo. Portanto, é imprescindível para o bom desempenho do sistema que a base de dados contenha registros que relatem o grau de satisfação dos usuários em relação a cada item acessado por eles. Tal informação pode ser observada por meio de um comentário, recomendações a outros usuários ou pela nota atribuída. Contudo, pretende-se que a abordagem proposta alcance melhor desempenho que outras nos casos em que o item tiver poucas interações no sistema. Isso ocorre em situações em que o item é novo no sistema, por exemplo. Tal fenômeno é conhecido na área como partida fria. Para efeito de pesquisa, pode-se obter descrições sintáticas de itens dos mais variados domínios e registros de avaliação sobre os mesmos, produzidos por usuários anônimos, em bases de dados públicas como Grouplens¹ ou Amazon².

¹ <http://grouplens.org/datasets/movielens/>

² <https://aws.amazon.com/pt/datasets/>

4.1.2 Base de Conhecimento

As descrições sintáticas armazenadas na base de dados, tais como nome, título e ano, são insuficientes e com pouca carga semântica para se analisar e calcular com precisão a similaridade entre dois itens. Para suprir a necessidade de dados com maior carga semântica, a proposta apresentada neste capítulo sugere a adição de uma IRI ao registro de cada item, de modo a indicar a sua representação em uma base de conhecimento. As IRIs são únicas e identificam entidades que são descritas por dados estruturados capazes de fornecer descrições com maior carga semântica. Dessa maneira, o vínculo entre os itens do acervo e as entidades das bases de conhecimento é realizado por meio das IRIs, que podem ser obtidas a partir de consultas SPARQL (Figura 8b) construídas com as descrições sintáticas da base (Figura 8a). As bases de conhecimento podem ser de acesso restrito, públicas ou desenvolvidas sob demanda para o domínio desejado. Geralmente, a escolha da base de conhecimento é guiada por características como sua abrangência sobre os itens do domínio trabalhado, carga semântica oferecida nas descrições dos itens, popularidade e padrão adotado para estruturar os dados (BAUER; KALTENBÖCK, 2012).

Na proposta apresentada aqui, é importante que a base de conhecimento seja estruturada de tal modo que as entidades de interesse sejam categorizadas por outras entidades com propriedades hierárquicas. Essas últimas entidades, denominadas daqui em diante por categorias (Figura 8c), devem compor um grupo hierárquico, cujas relações de hiperonímia³ também estejam representadas na base de conhecimento. Essa característica é importante porque o cálculo da similaridade entre os itens do acervo será baseado na busca por categorias comuns, mesmo que mais genéricas, entre eles. Dependendo do domínio, pode existir mais de uma propriedade que pode categorizar os recursos sob aspectos diferentes. Nesses casos, pode-se calcular a distância semântica nos diferentes tipos de categorização e obter uma média ponderada de acordo com a importância de cada categorização dentro do domínio trabalhado.

4.1.3 Cálculo da Distância Semântica

O ponto principal desta proposta está no cálculo da similaridade entre itens. Como dito na seção anterior, ele depende da estrutura hierárquica das categorias das entidades associadas aos itens.

Assim, o próximo passo consiste em estruturar as categorias semânticas obtidas, de modo que se possa comparar recursos entre si. Tais comparações serão úteis para que se possa aplicar os algoritmos tradicionais de filtragem colaborativa e baseados em conteúdo. Esta proposta sugere que as categorias semânticas sejam representadas em um grafo RDF, pois suas ligações podem ser exploradas no processo de comparação entre itens, conforme mostrado na Seção 3.3.4 do Capítulo 3. Também é possível representar as descrições semânticas em vetores de características, de tal modo que cada posição represente uma categoria presente ou ausente no

³ Relação semântica entre uma palavra de significado mais geral ou abrangente e outra ou outras com significado mais específico em relação à primeira.

recurso em questão. A Figura 9 demonstra graficamente como dois recursos diferentes podem ser representados por dois vetores de categorias. O valor armazenado em cada posição indica a presença ou não da categoria correspondente no recurso representado pelo vetor.

Figura 9 – Demonstração gráfica da estrutura hierárquica das categorias e da relação de categorias associadas a dois itens.

A Figura 9 também mostra que a estrutura hierárquica das categorias pode ser percorrida por meio da propriedade que a define, a fim de encontrar *links* entre as categorias e, assim, calcular a distância semântica entre duas categorias. O tipo de estrutura modelado pelas categorias irá definir a melhor técnica para o cálculo da distância semântica, conforme visto na Seção 3.3.4 do Capítulo 3.

O cálculo da distância semântica proposto aqui é derivado do apresentado por [Passant \(2010b\)](#), que contabiliza o número de *links* diretos (C_d) ou de *links* indiretos (C_i), distintos entre dois recursos para calcular a distância semântica direta ($LDSD_d$) e indireta ($LDSD_i$), respectivamente, entre eles. Cada *link* l representa um tipo de relacionamento entre os recursos, e ele pode ser direto quando o caminho percorrido através do *link* l não passa por nenhum outro recurso, ou indireto quando passa por um ou mais recursos ao longo do caminho.

A proposta também consiste em calcular a distância semântica entre dois recursos ($LDSD_{iExt}$) contabilizando os vínculos indiretos que chegam (C'_{ii}) ou saem (C'_{io}) deles, porém são contabilizados somente as ocorrências do *link* l que define a estrutura hierárquica das categorias. Para cada ocorrência é atribuído um peso que dependerá do número de recursos

(categorias) existentes no caminho entre os recursos, conforme observado nas Equações 4.1, 4.2 e 4.3:

$$C'_{io}(l, r_a, r_b) = \sum_{n \in N_{io}(r_a, r_b)} \frac{1}{1 + \log_2 k_n} \quad (4.1)$$

$$C'_{ii}(l, r_a, r_b) = \sum_{n \in N_{ii}(r_a, r_b)} \frac{1}{1 + \log_2 k_n} \quad (4.2)$$

$$LDSD_iExt(r_a, r_b) = \frac{1}{1 + C'_{io}(l, r_a, r_b) + C'_{ii}(l, r_a, r_b)} \quad (4.3)$$

onde $N_{io}(r_a, r_b)$ é o conjunto de caminhos indiretos do *link* l que saem de r_a e r_b em direção ao ancestral comum mais próximo, enquanto $N_{ii}(r_a, r_b)$ é conjunto dos caminhos indiretos do *link* l que chegam a r_a e r_b , vindos do ancestral comum mais próximo. k_n é o número de categorias existentes em cada caminho. A função $\log_2$ é aplicada sobre k_n para amenizar a punição sobre os caminhos que passam por muitas categorias.

Desse modo é possível observar que as Equações 4.1 e 4.2 atribuem maior significância aos caminhos com menor comprimento, diferentemente do que é observado na proposta apresentada por [Passant \(2010b\)](#), na qual todos os caminhos são contabilizados com a mesma importância. Essa ponderação possibilita que, além da identificação do relacionamento, seja inferida a intensidade do mesmo.

A Figura 10 retrata algumas das categorias associadas a dois filmes representados na ontologia da DBpedia e o *link* `skos:broader` que estrutura a hierarquia das categorias. Aplicando a fórmula descrita acima, pode-se contabilizar os caminhos representados e calcular a distância semântica entre os dois filmes conforme segue:

$$\begin{aligned} C'_{io}(l, r_a, r_b) &= \frac{1}{1 + \log_2 4} + \frac{1}{1 + \log_2 3} + \frac{1}{1 + \log_2 3} \\ C'_{io}(l, r_a, r_b) &= \frac{1}{3} + \frac{1}{2.585} + \frac{1}{2.585} \\ C'_{io}(l, r_a, r_b) &= 1.107 \end{aligned}$$

$$C'_{ii}(l, r_a, r_b) = 0$$

$$LDSD_iExt(r_a, r_b) = \frac{1}{1 + 1.107 + 0}$$

$$LDSD_iExt(r_a, r_b) = 0.475$$

Para efeito de comparação, considerando o *link* `skos:broader` como um *link* entre os filmes e observando que esse é o único entre eles, pode-se aplicar a fórmula apresentada por

Passant (2010b) para calcular a distância semântica entre esses mesmos filmes e obter o seguinte resultado:

$$C_{io}(l, r_a, r_b) = 3$$

$$C_{ii}(l, r_a, r_b) = 0$$

$$LDSD_i(r_a, r_b) = \frac{1}{1 + 3 + 0}$$

$$LDSD_i(r_a, r_b) = 0.25$$

A partir da comparação acima é possível observar que, segundo o cálculo da $LDSD_{iExt}(r_a, r_b)$, os recursos r_a e r_b estão mais distantes semânticamente do que é inferido no cálculo da $LDSD_i(r_a, r_b)$. Isso se deve aos pesos atribuídos a cada caminho ao contabilizá-los (Equações 4.1 e 4.2), que possibilitam calcular a distância semântica entre os recursos com maior precisão. Consequentemente, o cálculo da similaridade semântica será beneficiado.

Figura 10 – O ancestral comum “*Space_in_fiction*” das categorias “*Space_adventure_films*” e “*Extraterrestrial_life_in_popular_culture*” revela um link entre o filme *Alien* e o episódio *Lost in Space*.

4.1.4 Cálculo da Similaridade Semântica

O cálculo da distância semântica entre dois recursos revela a similaridade entre eles. Portanto, o cálculo da similaridade semântica entre itens pode ser efetuado a partir da distância semântica (Equação 4.3) entre os respectivos recursos, ao invés de buscar usuários com perfis semelhantes ou usuários que já avaliaram o item candidato à recomendação para observar os itens que receberam a mesma nota.

Analisando o grafo gerado a partir do relacionamento das categorias e a equação para o cálculo da distância semântica (Equação 4.3), pode-se concluir que quanto maior a distância semântica entre dois recursos, menor será a similaridade semântica. Diante dessa conclusão, esta proposta sugere a seguinte equação para o cálculo da similaridade semântica, $sim(a, b)$, entre dois itens, a e b :

$$sim(a, b) = 1 - \frac{LDSD_i Ext(r_a, r_b)}{1 + cat_{r_a} + cat_{r_b}} \quad (4.4)$$

onde cat_{r_a} e cat_{r_b} representam o total de categorias associadas aos recursos r_a e r_b , respectivamente. Dessa maneira, $sim(a, b)$ é calculada a partir da distância semântica entre os recursos r_a e r_b , usando os totais de categorias associadas a cada recurso para penalizar recursos com muitas categorias. Quanto menos categorias um recurso tem, mais significativas elas se tornam.

Esse cálculo é realizado par-a-par entre todos os itens do acervo (Figura 8d) e o valor obtido é armazenado na base de dados (Figura 8e) para que esse esforço computacional seja necessário apenas uma vez. A cada inserção de um novo item, será necessário calcular apenas a similaridade entre ele e cada um dos itens já existentes. Essa característica pode amenizar o problema do novo item em sistemas de recomendação baseados em filtragem colaborativa, nos quais os itens novos são recomendados apenas após obterem uma certa quantia de avaliações boas no sistema, conforme discutido na Seção 2.1.2

4.1.5 Sistemas de Recomendação Baseados em Filtragem Colaborativa

O cálculo da similaridade entre itens está entre as tarefas mais importantes em um sistema de filtragem colaborativa e pode ser efetuado usando diferentes técnicas (DESROSIERS; KARYPIS, 2011). Especialmente falando de sistemas de filtragem colaborativa kNN (k Vizinhos mais Próximos, do Inglês, k -Nearest-Neighbors) baseados em itens, a recomendação de novos itens a um determinado usuário é baseada na predição de notas para os itens mais similares aos melhores avaliados por esse usuário (Figura 8g). A predição da nota, $\hat{r}_{ui}$, de um item i para um usuário u pode ser uma média ponderada das notas atribuídas aos itens do conjunto $S_i^k(u)$, que é

composto pelos k itens mais similares a i que foram avaliados por u (Equação 4.5):

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S_i^k(u)} (r_{uj} - b_{uj}) w_{ij}}{\sum_{j \in S_i^k(u)} |w_{ij}|} \quad (4.5)$$

onde b_{ui} é o baseline do usuário u e item i , conforme apresentado na Seção 2.1.2.1 do Capítulo 2, e w_{ij} é o valor que representa a similaridade entre o item i , para o qual queremos prever a nota que o usuário u daria, e o item j , que já foi avaliado pelo usuário u . O valor de w_{ij} pode ser calculado usando-se diferentes técnicas, inclusive correlação por notas. Entretanto a correlação por notas enfrenta dificuldades no cenário de partida fria que são amenizadas no cálculo da similaridade semântica, como ocorre na proposta deste trabalho (Equação 4.3). Para selecionar os itens do conjunto $S_i^k(u)$, primeiro se seleciona os itens avaliados por u e, desse subconjunto, extrai-se os k itens mais semelhantes a i para compor $S_i^k(u)$.

A predição da nota $\hat{r}_{ui}$ também pode ser baseada no conjunto de itens $R_i^k(u)$, que é composto pelos itens que já foram avaliados por u e estão entre os k itens mais similares a i (Equação 4.6):

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in R_i^k(u)} (r_{uj} - b_{uj}) w_{ij}}{\sum_{j \in R_i^k(u)} |w_{ij}|} \quad (4.6)$$

Para selecionar os itens do conjunto $R_i^k(u)$, primeiro se seleciona os k itens mais semelhantes a i e, desse subconjunto, extrai-se os que já foram avaliados por u para compor $R_i^k(u)$.

Na prática, pode-se observar que, quando se utiliza o conjunto $S_i^k(u)$, o cálculo da predição da nota $\hat{r}_{ui}$ pode considerar notas atribuídas a itens pouco similares a i , com pesos menores, se o usuário u avaliou poucos itens similares a i . Assim, a quantidade de itens selecionados para prever $\hat{r}_{ui}$, frequentemente é igual a k . Por outro lado, o conjunto $R_i^k(u)$ é mais restrito e, frequentemente, a quantidade de itens selecionados é menor que k (KOREN, 2008). Ambas versões são comparadas no capítulo de avaliação do sistema proposto.

Uma vez predita a nota de cada recurso em relação aos melhores avaliados por um determinado usuário, os respectivos itens poderão ser ordenados por nota em uma lista. Após o preenchimento da lista com itens relevantes, os n primeiros itens serão retornados ao usuário no formato de uma lista de recomendações, conforme ilustrado na Figura 8h.

Assim, esta proposta apresenta um sistema de recomendação híbrido, que segue os procedimentos básicos de um sistema de filtragem colaborativa, porém calcula a similaridade entre itens baseando-se em suas características. Essa combinação ameniza o problema do item novo, inerente tanto a sistemas de filtragem colaborativa como a sistemas baseados em conteúdo, conforme discutido nas Seções 2.1.1 e 2.1.2 do Capítulo 2.

4.2 Considerações Finais

Este capítulo apresentou como descrições semânticas dos itens de um acervo podem ser obtidas de bases de conhecimento e usadas em sistemas de recomendação para encontrar os itens avaliados por um usuário u que são mais similares a um item i , ainda não avaliado por u , e prever a nota $\hat{r}_{ui}$ que u daria a i . Qualquer base de conhecimento pode ser usada desde que tenha pelo menos um grupo de recursos, estruturados hierarquicamente, que categorizem os recursos que representam os itens do acervo. Essa característica é essencial para o cálculo da similaridade entre os itens.

A proposta tem sua aplicação voltada para sistemas baseados em vizinhança, mais especificamente baseados em itens, e sugere uma nova técnica para o cálculo da similaridade semântica. Assim, tomamos como referência uma das abordagens mais comuns de sistemas de recomendação com essas características, a k NN, para adaptá-la e substituir o cálculo da similaridade baseada na correlação de notas pelo cálculo da similaridade semântica (Equação 4.3).

No próximo capítulo será detalhada a metodologia empregada nos experimentos e apresentados os resultados obtidos pela proposta, comparando esses resultados com os de outras técnicas de cálculo de similaridade.

AVALIAÇÃO DO SISTEMA

5.1 Metodologia

O capítulo anterior descreveu em termos gerais a arquitetura do sistema proposto neste trabalho e apresentou possibilidades de implementação. Nesta seção será descrita a implementação da arquitetura apresentada no capítulo anterior, usando a Figura 11 para ilustrar as sete tarefas do processo de recomendação configuradas para o domínio de filmes e os detalhes de implementação definidos de acordo com os experimentos realizados.

Figura 11 – Visão geral da arquitetura implementada nos experimentos do SR proposto.

5.1.1 Base de Dados

O sistema utiliza como base de dados um SGBD, devido à simplicidade para manipular os dados, seja recuperando ou inserindo registros de itens do acervo. O SGBD também permite modelar uma estrutura de tabelas e relacionamentos que torna mais eficiente a recuperação de dados desconexos inicialmente, além de possibilitar o armazenamento de resultados parciais de maneira relacionada e prática. Assim, os dados sobre os itens do acervo de filmes ficaram armazenados em um banco de dados MySQL¹. Tais dados foram obtidos a partir da base de dados MovieLens 100k, disponibilizada gratuitamente pelo GroupLens². Essa base de dados contém informações sobre filmes e as respectivas avaliações recebidas por usuários do site MovieLens³. A MovieLens 100k possui 100.000 avaliações (de 1 a 5) de 943 usuários sobre 1682 filmes, sendo que cada usuário avaliou pelo menos 20 filmes. Os dados coletados correspondem ao período de 19 de setembro de 1997 até 22 de abril de 1998 e compreendem informações sobre filmes, como título, ano de lançamento e gêneros associados. Também são compreendidos dados demográficos sobre os usuários, como idade, sexo, ocupação e cep, que não foram explorados neste projeto. Essa base foi escolhida devido ao perfeito atendimento aos requisitos do sistema e ao grande número de referências em trabalhos da área de SR, o que facilita a comparação dos resultados obtidos com outros trabalhos. O uso de uma base de dados maior mostrou-se inviável devido ao alto custo computacional para se calcular a similaridade entre todos os filmes em tempo aceitável.

Definida a base de dados, o sistema usa os dados referentes ao título, ano e diretor de cada filme para buscar os respectivos recursos na base de conhecimento.

5.1.2 Base de Conhecimento

Duas das maiores bases de conhecimento para o domínio de filmes são a DBpedia e a YAGO⁴, que possuem ontologias que atendem às necessidades do projeto e já são bastante difundidas, fator que facilita a integração com outras bases e sistemas. A DBpedia foi a base escolhida por ter um nível de detalhamento superior no conjunto de dados para filmes, principalmente na quantidade de categorias associadas, que é a característica usada no cálculo da similaridade.

Mais de 97% dos filmes da MovieLens 100k foram encontrados na DBpedia. Além dos dados obtidos da MovieLens 100k, é necessário armazenar a URI do recurso correspondente a cada filme na base de conhecimento. Para isso, foi adicionado mais um campo na tabela de dados dos filmes no banco de dados. As URIs são retornadas por consultas SPARQL formuladas a partir do título e ano dos filmes (Figura 11a), que buscam tais dados nas propriedades name e abstract dos recursos do tipo <http://dbpedia.org/ontology/Film> presentes na DBpedia

¹ <https://www.mysql.com/>

² <http://grouplens.org/datasets/movielens/>

³ <https://movielens.org/>

⁴ <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

(Figura 11b). Uma vez encontrada a URI do recurso que representa um filme, é possível acessar todos os dados da base de conhecimento relacionados a ele. Por meio da propriedade `dct:subject` o sistema obtém as categorias associadas a cada filme (Figura 11c), por exemplo.

A base DBpedia usa diferentes esquemas de classificação para as entidades representadas em seu conjunto de dados, dentre as quais podemos destacar o vocabulário SKOS (Sistema de Organização do Conhecimento Simples, do Inglês, *Simple Knowledge Organization System*)⁵, um modelo de dados comum para compartilhar e vincular sistemas de organização de conhecimento na Web dos Dados⁶. A propriedade `skos:broader` do vocabulário SKOS está presente no conjunto de dados da DBpedia e pode revelar relacionamentos do tipo hiperônimo entre duas entidades.

Nesta proposta, a propriedade `skos:broader` é usada para encontrar as categorias genéricas de cada categoria associada a um filme, a fim de encontrar categorias comuns entre dois filmes. Ou seja, podemos observar uma hierarquia entre as categorias de filmes, que é estruturada pela propriedade `skos:broader`, na qual é possível encontrar um ancestral comum entre duas categorias diferentes. Desse modo, pode-se afirmar que quanto mais específico for o ancestral comum entre categorias de filmes diferentes, maior é a similaridade entre esses filmes em relação ao vínculo analisado.

Contudo, a propriedade `skos:broader` não é transitiva. É recomendado o uso da propriedade `skos:broaderTransitive`⁷ para garantir a transitividade entre um conjunto de entidades, em relação a um relacionamento hiperônimo. Porém, não foi encontrada a propriedade `skos:broaderTransitive` no conjunto de dados da DBpedia trabalhado no projeto e foi observada a presença de alguns ciclos na estrutura quando a percorremos em busca dos ancestrais de algumas categorias por meio da propriedade `skos:broader`. Outra característica relevante da estrutura de dados da DBpedia é o fato de uma entidade possuir mais de um vínculo a partir da propriedade `skos:broader`. Ou seja, uma entidade pode ter dois ou mais ancestrais distintos ligados diretamente a ela. Essas duas características foram determinantes para a adoção de uma estrutura de grafo, ao invés de árvore, para representar as categorias no sistema. Foi utilizada a biblioteca JGraphT⁸ para representar e manipular o grafo das categorias semânticas no sistema em tempo de execução. Essa biblioteca possui objetos e algoritmos estudados na teoria de grafos implementados na linguagem Java. Dessa maneira, o algoritmo de Dijkstra (CHERKASSKY; GOLDBERG; RADZIK, 1996) foi usado para encontrar o menor caminho entre as categorias semânticas, visto que ele trata grafos com ciclos e o fato de uma entidade ter mais de um recurso ligado diretamente a ela, não ser um problema na representação de grafos.

Além disso, pode-se observar que quando o volume de dados a ser recuperado da DBpedia é grande, não é possível obtê-los a partir de uma única consulta. Essa restrição e o longo tempo de resposta nas consultas, tornaram o acesso à base online inviável. Considerando

⁵ <http://wiki.dbpedia.org/services-resources/datasets/dbpedia-datasets#h434-6>

⁶ <https://www.w3.org/TR/skos-reference/>

⁷ <https://www.w3.org/TR/skos-reference/#L2413>

⁸ <http://jgrapht.org/>

que apenas a estrutura de categorias e o relacionamento delas com os filmes são utilizados no sistema, foi desenvolvido um sistema auxiliar que extraiu os dados desejados da DBpedia e os inseriu em uma base de conhecimento local. A base de conhecimento local foi construída usando o software BlazeGraph⁹, que é uma base de dados em grafos com suporte à API RDF/SPARQL. Os dois sistemas desenvolvidos utilizam a biblioteca Jena¹⁰ para recuperar os dados a partir de consultas SPARQL na DBpedia (sistema auxiliar) e na base de conhecimento local (sistema principal). Portanto, o sistema representa parte do conjunto de dados RDF da DBpedia (filmes e categorias) localmente, onde cada tripla sujeito-predicado-objeto existente nesse conjunto RDF é representada por uma aresta entre dois vértices em um grafo direcionado e não valorado.

Desse modo, o sistema proposto pode recuperar as triplas necessárias da base de conhecimento local de maneira rápida e irrestrita, para calcular a similaridade semântica entre filmes conforme ilustrado na Figura 12.

Figura 12 – Representação de parte dos dados da base MovieLens e da base de conhecimento local projetada para o sistema proposto.

⁹ <https://www.blazegraph.com/>

¹⁰ <https://jena.apache.org/>

Apenas 47 dos 1681 filmes da base MovieLens 100k não tiveram seus respectivos recursos encontrados na DBpedia e foram excluídos dos experimentos. Após a extração dos dados desejados da DBpedia, foram armazenadas 4.012.746 categorias semânticas na base de conhecimento local, sendo que 4.505 estão ligadas diretamente a filmes. Cada filme possui, em média, 15 categorias semânticas associadas diretamente a ele.

5.1.3 Cálculo da Similaridade Semântica

Os experimentos, cujos resultados serão apresentados nas próximas seções, compararam os desempenhos obtidos por diferentes SRs que calculam a similaridade entre os filmes do acervo de modos diferentes. Porém, todos eles precisam recuperar os dados da base de conhecimento local e reconstruir o grafo das categorias instanciando objetos na memória principal em tempo de execução, para percorrê-lo e calcular a similaridade semântica entre os filmes. As classes para a construção e manipulação desse grafo na memória principal foram obtidas da biblioteca JGraphT¹¹.

Além do SR que usa o cálculo da similaridade de itens proposto neste trabalho na Seção 4.1.4 do Capítulo 4, foram desenvolvidos outros 6 SRs com filtragem colaborativa baseada em itens, que usam técnicas diferentes para calcular a similaridade entre os filmes, para terem os desempenhos comparados. As técnicas escolhidas foram as seguintes: i) similaridade baseada em correlação de avaliações; ii) similaridade baseada em correlação de atributos; iii) LDSDi (PASSANT, 2010b); iv) WordNet - menor caminho; v) WorNet - Leacock & Chodorow (LEACOCK; CHODOROW, 1998); e vi) WordNet - Wu & Palmer (WU; PALMER, 1994).

A correlação usada nas técnicas i e ii é a correlação de cosseno, que calcula a correlação entre os itens baseando-se em seus respectivos vetores de características (DESROSIERS; KARYPIS, 2011). A primeira técnica constrói vetores, nos quais as posições representam os usuários e os valores representam as notas desses usuários para o item em questão. A segunda técnica, por sua vez, constrói vetores, nos quais as posições representam características e os valores representam a presença ou ausência da respectiva característica no item em questão. Essa última pode usar diferentes tipos de característica fornecidos como parâmetro de entrada no sistema. Nos experimentos realizados, o SR que aceita características dos itens como parâmetro de entrada foi executado com os gêneros dos filmes, obtidos da base MovieLens, e com as categorias semânticas diretas, obtidas da base de conhecimento local.

O cálculo da similaridade usando LDSDi, conforme apresentado por Passant (2010b), contabiliza apenas *links* distintos. Porém, a implementação usada nos experimentos contabiliza todos os *links*, visto que no contexto trabalhado temos apenas um tipo de *link*. Essa adaptação tem o objetivo de avaliar o ganho de desempenho obtido quando ponderamos os *link* de acordo com o tamanho do caminho, ao invés de apenas contabilizá-los.

¹¹ <http://jgrapht.org/>

Para que as técnicas iv, v e vi, baseadas na WordNet, possam ser utilizadas, foram inseridas arestas no grafo que vinculam todos os vértices fontes com um vértice adicional nomeado como `Category:Thing` (Figura 13). Assim, o vértice `Category:Thing` desempenha um papel de raiz enquanto os vértices sumidouros podem ser considerados como as folhas de uma estrutura de árvore. Conforme descrito nas Seções 3.3.1, 3.3.2 e 3.3.3 do Capítulo 3, tais elementos são necessários para se obter o valor da profundidade máxima da árvore, que na implementação apresentada equivale ao número de arestas do maior caminho existente entre um vértice sumidouro e o vértice `Category:Thing` do grafo.

Figura 13 – Representação de parte da hierarquia de categorias semânticas, após a adição da categoria `Category:Thing`.

A próxima seção descreve como as diferentes versões de SRs foram adaptadas para adotar os cálculos de similaridade supracitados.

5.1.4 Sistemas Implementados

Todos os experimentos foram executados usando o software livre MyMediaLite que oferece várias implementações de SRs¹². A partir de seu algoritmo ItemKNN, que usa similaridade baseada em correlação de avaliações, foram criadas outras versões que adaptam o algoritmo original para usar os cálculos de similaridade: LDSDi; LDSDiExt, Sim. LDSDiExt; WordNet - menor caminho; WorNet - Leacock & Chodorow; e WordNet - Wu & Palmer.

¹² <http://www.mymedialite.net/>

Os SRs que calculam apenas a distância semântica LDSDi e LDSDiExt para determinar a similaridade entre os filmes exploram a relação inversamente proporcional que existe entre as duas grandezas. Tal relação pode ser observada quando se verifica que quanto maior é a distância semântica entre dois recursos, menor é a similaridade entre eles. O SR Sim. LDSDiExt também calcula a distância semântica LDSDiExt, mas implementa o cálculo de similaridade proposto no trabalho (Equação 4.4).

Também foram executados dois SRs com o algoritmo ItemAttributeKNN implementado. Esse algoritmo usa o cálculo de similaridade baseado na correlação de atributos. O SR ItemKNN_g, que implementa o algoritmo ItemAttributeKNN original do MyMediaLite, recebeu como entrada de dados a relação dos filmes com os 19 gêneros relacionados na base MovieLens e o SR ItemKNN_c, adaptado do anterior, recebeu a relação dos filmes com as 4.505 categorias semânticas diretas extraídas da base de conhecimento local.

Os algoritmos, inclusive o ItemKNN e o ItemAttributeKNN, foram adaptados para trabalhar com o cálculo da predição de notas baseado no conjunto de itens $S_i^k(u)$ ou $R_i^k(u)$ (Capítulo 4, Seção 4.1.5), os quais foram executados e avaliados conforme descrito na seção a seguir.

5.2 Método de Avaliação

As configurações de cada SR variam em relação ao número máximo de filmes semelhantes, denominados vizinhos, que são usados para predizer uma nota, em relação aos critérios para a formação do conjunto dos filmes vizinhos, $S_i^k(u)$ ou $R_i^k(u)$. O número de vizinhos varia entre 50 e 600.

A avaliação de cada configuração foi realizada sobre experimentos *offline*, pois foram executadas simulações com um conjunto de dados fornecido por usuários, e focada na predição de notas, visto que é a técnica adotada por todos os SRs que foram avaliados. Além disso, todas as propostas de alterações nos algoritmos ou nas configurações dos SRs geram impacto direto na predição de notas. Portanto, foi escolhida a métrica RMSE para avaliar o desempenho na predição de notas porque é comumente usada para avaliar esses tipos de SRs e penaliza mais os erros maiores do que a métrica MAE.

Diante das características mencionadas, foi escolhida a técnica de validação cruzada em 5 partições, para garantir que cada versão de SR seja executada 5 vezes com conjuntos de teste diferentes, conforme descrito na Seção 2.2.2 do Capítulo 2. Também foram realizados testes de significância estatística sobre os resultados obtidos, a fim de avaliar o nível de confiança dos valores observados.

Além de executar as versões dos SRs no cenário normal, eles também foram executados no cenário de partida fria, no qual é simulado o desempenho dos SRs com um conjunto de teste

com apenas itens novos. Ou seja, é avaliado o desempenho da predição de notas dos SRs para itens que não estavam no conjunto de treinamento. Sendo assim, as análises referentes a um cenário normal levam em consideração o RMSE de todos os itens do conjunto de teste, enquanto as referentes a um cenário de partida fria consideram apenas o RMSE de itens novos, presentes no mesmo conjunto de teste. O desempenho diferente dos SRs em cada cenário, juntamente com os resultados dos testes de significância, pode ser observado na próxima seção.

5.3 Resultados

Nesta seção são apresentados os resultados obtidos na execução dos sistemas descritos até aqui com diferentes configurações e em diferentes cenários a fim de se comparar seus desempenhos e entender em quais circunstâncias o uso de descrições semânticas pode favorecer um SR.

Na apresentação dos resultados, serão denominados LDSDi e LDSDiExt os SRs que adotam os cálculos de distância LDSDi (Equação 3.5) e LDSDiExt (Equação 4.3), respectivamente. O SR que também usa o cálculo de distância LDSDiExt, mas calcula a similaridade conforme a Equação 4.4, apresentada no Capítulo 4, será denominado Sim. LDSDiExt. Os SRs que adotam os cálculos de similaridade WordNet - menor caminho, WordNet - Leacock & Chodorow e WordNet - Wu & Palmer serão denominados Menor Caminho, L&C e W&P, respectivamente. Serão denominados ItemKNN_n, ItemKNN_g e ItemKNN_c os SRs que calculam a similaridade usando correlação por cosseno, baseando-se em avaliações, gêneros e categorias semânticas diretas, respectivamente.

Serão considerados baselines para comparação os SRs LDSDi, Menor Caminho, L&C, W&P, ItemKNN_n e ItemKNN_g, sendo que os quatro primeiros usam dados semânticos como o SR Sim. LDSDiExt que está sendo proposto.

Inicialmente, todos os algoritmos usaram o conjunto $S_i^k(u)$, que é composto dos k itens mais similares a i que foram avaliados por u , para prever a nota $\hat{r}_{ui}$ que o usuário u daria ao item i (Equação 4.5).

5.3.1 SR LDSDiExt vs. SRs baselines

Os primeiros experimentos investigaram se apenas o cálculo da distância semântica proposto (Equação 4.3) seria suficiente para um SR alcançar desempenho melhor que os SRs baselines.

Nos resultados plotados na Figura 14a, observamos que, no cenário normal, o SR LDSDiExt só obteve desempenho melhor que o SR LDSDi. Porém, essa diferença de desempenho é extremamente significativa (Tabela 1) de acordo com os critérios convencionais ($p\text{-value} < 0,0001$).

Figura 14 – LDSDiExt vs. baselines

No cenário de partida fria (Figura 14b), o SR LDSDi e o SR LDSDiExt obtiveram desempenho melhor que todos os outros SRs a partir de 450 vizinhos, porém essa melhora foi estatisticamente significativa a partir de 500 vizinhos contra 4 algoritmos: ItemKNN_n; Menor Caminho; L&C; e W&P (Tabela 2).

Esses resultados revelam a tendência do SR LDSDiExt, que trabalha com descrições semânticas, obter melhor desempenho no cenário de partida fria do que os SRs ItemKNN_n e ItemKNN_g, que trabalham apenas com notas e descrições sintáticas da base de dados, respectivamente. Quando o SR precisa prever uma nota para um item novo, tem-se poucas ou nenhuma avaliação sobre ele na base de dados. Isso explica a baixa eficiência de SRs que adotam similaridade baseada em correlação de avaliações nesse cenário. Entretanto, o uso de descrições semânticas e um número maior de itens semelhantes para as predições de notas podem melhorar sua eficiência.

Tabela 1 – LDSDiExt vs. baselines - Resultados obtidos no cenário normal. Os melhores resultados estão em destaque.

k	LDSDi Ext	Item KNN_n	Item KNN_g	LDSDi	Menor Caminho	L&C	W&P
50	0,958005	0,921063	0,937162	0,961152***	0,945468	0,944153	0,944562
100	0,953892	0,926740	0,936671	0,959668***	0,942991	0,942443	0,942073
150	0,952401	0,929378	0,936566	0,958993***	0,942953	0,942422	0,942538
200	0,953469	0,930523	0,936539	0,964518***	0,942850	0,942736	0,942889
250	0,953883	0,931202	0,936521	0,966296***	0,942935	0,943035	0,943156
300	0,953872	0,931518	0,936503	0,966702***	0,943146	0,943293	0,943324
350	0,954030	0,931661	0,936500	0,967273***	0,943240	0,943423	0,943347
400	0,954054	0,931747	0,936495	0,967321***	0,943311	0,943442	0,943403
450	0,953987	0,931791	0,936493	0,967313***	0,943351	0,943416	0,943470
500	0,953948	0,931815	0,936493	0,967264***	0,943451	0,943430	0,943522
550	0,953989	0,931822	0,936493	0,967321***	0,943562	0,943443	0,943538
600	0,954277	0,931823	0,936493	0,967943***	0,943598	0,943451	0,943531

A seguinte notação foi usada para a significância estatística: (***) quando $p\text{-value} < 0,001$.

Tabela 2 – LDSDiExt vs. baselines - Resultados obtidos no cenário de partida fria. Os melhores resultados estão em destaque.

k	LDSDi Ext	Item KNN_n	Item KNN_g	LDSDi	Menor Caminho	L&C	W&P
50	1,23252	1,10677	1,09350	1,23883	1,11774	1,08265	1,05505
100	1,19192	1,10677	1,09506	1,19695	1,10848	1,08488	1,07463
150	1,14280	1,10677	1,10401	1,14154	1,09390	1,08945	1,08343
200	1,11827	1,10677	1,10891	1,12957*	1,08765	1,09410	1,08811
250	1,10395	1,10677	1,11287	1,10552	1,08197	1,10195	1,09343
300	1,08979	1,10677	1,11451	1,08937	1,08160	1,10317	1,09346
350	1,08438	1,10677	1,11495	1,08416	1,08404	1,10457	1,09587
400	1,08537	1,10677	1,11515	1,08307	1,08342	1,10219*	1,09474
450	1,07555	1,10677*	1,11517	1,07698	1,08292	1,09797**	1,09690*
500	1,06694	1,10677**	1,11517	1,06869	1,08500*	1,09575**	1,09760**
550	1,05735	1,10677**	1,11517	1,05865	1,09073**	1,09413**	1,09567**
600	1,04111	1,10677**	1,11517	1,03088	1,09340**	1,09266**	1,09456**

A seguinte notação foi usada para a significância estatística: (*) quando $p\text{-value} < 0,1$; e (**) quando $p\text{-value} < 0,05$.

5.3.2 SR Sim. LDSDiExt vs. SRs baselines

Quando a Equação 4.4, que potencializa o peso das categorias de um item na medida que diminui o número de categorias associadas a ele, é adotada para calcular a similaridade semântica, observa-se uma pequena perda de desempenho no cenário normal, porém um ganho significativo no cenário de partida fria, em relação ao SR LDSDiExt. Consequentemente, observa-se um desempenho pior na comparação com os SRs baselines no cenário normal (Figura 15a e Tabela 3), mas um desempenho melhor no cenário de partida fria (Figura 15b). A significância estatística

Figura 15 – Sim. LDSDiExt vs. baselines

dos resultados do SR Sim. LDSDiExt em relação aos do SR ItemKNN e os do SR LDSDi é maior que a obtida pelos resultados do SR LDSDiExt (Tabela 4). Além disso, para alcançar essa melhora de desempenho, o SR Sim. LDSDiExt precisa de apenas 300 vizinhos para prever notas. A redução no número de vizinhos implica em maior desempenho do SR no tempo de execução, visto que será usado um conjunto menor de notas para calcular a média e prever uma nota.

Dado o melhor desempenho do SR Sim. LDSDiExt em relação ao SR LDSDiExt, pode-se concluir que usar apenas o valor da distância semântica entre dois itens para estimar a similaridade entre eles não é eficiente. A ponderação do conjunto de categorias contribui no processo de comparação entre itens, dando pesos distintos conforme a relevância de cada categoria do conjunto para categorizá-lo.

Tabela 3 – Sim. LDSDiExt vs. baselines - Resultados obtidos no cenário normal. Os melhores resultados estão em destaque.

k	Sim. LDSDi Ext	Item KNN_n	Item KNN_g	LDSDi	Menor Caminho	L&C	W&P
50	0,961171	0,921063	0,937162	0,961152	0,945468	0,944153	0,944562
100	0,960242	0,926740	0,936671	0,959668	0,942991	0,942443	0,942073
150	0,960156	0,929378	0,936566	0,958993	0,942953	0,942422	0,942538
200	0,967421	0,930523	0,936539	0,964518	0,942850	0,942736	0,942889
250	0,972711	0,931202	0,936521	0,966296	0,942935	0,943035	0,943156
300	0,975311	0,931518	0,936503	0,966702	0,943146	0,943293	0,943324
350	0,976476	0,931661	0,936500	0,967273	0,943240	0,943423	0,943347
400	0,976752	0,931747	0,936495	0,967321	0,943311	0,943442	0,943403
450	0,976737	0,931791	0,936493	0,967313	0,943351	0,943416	0,943470
500	0,976726	0,931815	0,936493	0,967264	0,943451	0,943430	0,943522
550	0,977125	0,931822	0,936493	0,967321	0,943562	0,943443	0,943538
600	0,978420	0,931823	0,936493	0,967943	0,943598	0,943451	0,943531

Tabela 4 – Resultados obtidos no cenário de partida fria. Os melhores resultados estão em destaque.

k	Sim. LDSDi Ext	Item KNN_n	Item KNN_g	LDSDi	Menor Caminho	L&C	W&P
50	1,2122	1,10677**	1,093502*	1,238836*	1,117746**	1,082656**	1,055056**
100	1,192404	1,10677**	1,095062**	1,196952	1,108482**	1,084886**	1,07463**
150	1,123772	1,10677	1,104014	1,141546	1,093906	1,089456	1,08343
200	1,11637	1,10677	1,10891	1,12957	1,087654	1,094108	1,08811
250	1,081594	1,10677*	1,11287	1,105526*	1,081974	1,101956	1,093438
300	1,07279	1,10677**	1,114518	1,089376**	1,081604	1,103176	1,093464
350	1,061558	1,10677*	1,114952	1,084164	1,084046	1,104578*	1,095876
400	1,054004	1,10677**	1,115158	1,083078	1,083424	1,102196*	1,094746*
450	1,047746	1,10677**	1,115172	1,076982	1,082922	1,097976*	1,096906*
500	1,0361	1,10677**	1,115172	1,068692	1,085*	1,095758**	1,097602**
550	1,038742	1,10677**	1,115172	1,05865	1,090738*	1,094134*	1,095678*
600	0,995954	1,10677**	1,115172	1,030888	1,093404*	1,092666*	1,094566*

A seguinte notação foi usada para a significância estatística: (*) quando $p\text{-value} < 0,1$; e (**) quando $p\text{-value} < 0,05$.

5.3.3 SR Sim. LDSDiExt usando o conjunto $S_i^k(u)$ vs. SRs baselines usando o conjunto $R_i^k(u)$

Os experimentos com os SRs configurados para trabalhar com o conjunto de vizinhos $R_i^k(u)$ mostraram que, no cenário normal, não ocorrem grandes alterações no desempenho médio dos SRs na predição de notas (Figura 16a). Porém, no cenário de partida fria o SR Sim. LDSDiExt, usando o conjunto $R_i^k(u)$, obteve um desempenho pior que o observado com o conjunto $S_i^k(u)$. Fazendo uma comparação direta entre o SR Sim. LDSDiExt com o conjunto $R_i^k(u)$ e com o conjunto $S_i^k(u)$ por meio da Figura 16a, é possível observar que o sistema que usou

Figura 16 – Sim. LDSDiExt vs. baselines

o conjunto $R_i^k(u)$ obteve melhores resultados no cenário normal (Tabela 5). Contudo, nenhum dos dois teve desempenho melhor que os SRs ItemKNN_n e ItemKNN_g. Por outro lado, no cenário de partida fria, o sistema que usou o conjunto $S_i^k(u)$ manteve-se melhor que todos SRs comparados a partir de 350 vizinhos (Figura 16b e Tabela 6).

Os resultados dos experimentos no cenário de partida fria que visam analisar a influência do uso do conjunto de vizinhos $S_i^k(u)$ ou $R_i^k(u)$ nos SRs comparados mostram que o uso do conjunto $S_i^k(u)$ pode suprir a ausência de dados sobre um item i para predizer a nota $\hat{r}_{ui}$. Isso se deve ao fato do conjunto $R_i^k(u)$ ser mais restritivo e fazer com que o SR trabalhe com um conjunto menor de itens semelhantes, somente os avaliados pelo usuário u dentre os k mais similares a i , para predizer a nota $\hat{r}_{ui}$.

Tabela 5 – Sim. LDSDiExt vs. baselines - Resultados obtidos no cenário normal, comparando o desempenho do SR Sim. LDSDiExt usando o conjunto de vizinhos $S_i^k(u)$ com os SRs baselines usando o conjunto de vizinhos $R_i^k(u)$. Os melhores resultados estão em destaque.

k	Sim. LDSDi Ext $S_i^k(u)$	Sim. LDSDi Ext $R_i^k(u)$	Item KNN_n	Item KNN_g
50	0,961171	1,022183***	0,926272	1,013380***
100	0,960242	0,986890***	0,921660	0,984827***
150	0,960156	0,976237**	0,920919	0,968849***
200	0,967421	0,968400	0,921877	0,961914
250	0,972711	0,961869	0,923103	0,956992
300	0,975311	0,958146	0,924212	0,952430
350	0,976476	0,956111	0,925224	0,950032
400	0,976752	0,954168	0,926186	0,947561
450	0,976737	0,953242	0,927006	0,945333
500	0,976726	0,951504	0,927769	0,943619
550	0,977125	0,950719	0,928482	0,942123
600	0,978420	0,949551	0,929029	0,940137

A seguinte notação foi usada para a significância estatística: (**) quando $p\text{-value} < 0,05$; e (***) quando $p\text{-value} < 0,001$.

5.3.4 SR ItemKNN_g vs. SR ItemKNN_c

Os benefícios que as descrições semânticas podem trazer para um SR também são observáveis quando se compara o desempenho do SR ItemKNN_g, que usa gêneros, e do SR ItemKNN_c, que usa categorias semânticas diretas. Os filmes podem ter um ou mais gêneros de um total de 19 extraídos da base de dados MovieLens. As categorias semânticas diretas, por

Tabela 6 – Sim. LDSDiExt vs. baselines - Resultados obtidos no cenário de partida fria, comparando o desempenho do SR Sim. LDSDiExt usando o conjunto de vizinhos $S_i^k(u)$ com os SRs baselines usando o conjunto de vizinhos $R_i^k(u)$. Os melhores resultados estão em destaque.

k	Sim. LDSDi Ext $S_i^k(u)$	Sim. LDSDi Ext $R_i^k(u)$	Item KNN_n	Item KNN_g
50	1,21220	1,40571***	1,10677**	1,07429**
100	1,19240	1,30051**	1,10677**	1,10686**
150	1,12377	1,29177**	1,10677	1,07263
200	1,11637	1,26603**	1,10677	1,12415
250	1,08159	1,23445**	1,10677*	1,12755
300	1,07279	1,21395**	1,10677**	1,13227
350	1,06156	1,20324**	1,10677*	1,14043
400	1,05400	1,19534**	1,10677**	1,10657
450	1,04775	1,19012**	1,10677**	1,09256
500	1,03610	1,17212**	1,10677**	1,09728
550	1,03874	1,15441**	1,10677**	1,09762
600	0,99595	1,14622**	1,10677**	1,09833

A seguinte notação foi usada para a significância estatística: (*) quando $p\text{-value} < 0,1$; (**) quando $p\text{-value} < 0,05$; e (***) quando $p\text{-value} < 0,001$.

Figura 17 – ItemKNN_c vs. ItemKNN_g e ItemKNN_n

sua vez, foram obtidas por meio dos recursos que representam os filmes do acervo na base de conhecimento DBpedia. São 4505 categorias semânticas diretas que vão além da descrição do gênero do filme e fornecem informações geográficas, temporais e tipos de mídia entre outras.

Os resultados apresentados nas Figuras 17a e 17b mostram que o SR ItemKNN_c obteve melhor desempenho que o SR ItemKNN_g nos dois cenários, normal e de partida fria, apesar de haver significância estatística apenas no cenário normal (Tabela 7).

Também é possível observar na Figura 17b que SR ItemKNN_c superou o desempenho do SR ItemKNN_n no cenário de partida fria, com significância estatística (Tabela 8). Isso reforça a afirmação de que os SRs podem se beneficiar usando descrições semânticas para calcular a similaridade entre itens a fim de amenizar o problema enfrentado com novos itens no acervo.

Tabela 7 – Resultados obtidos no cenário normal. Os melhores resultados estão em destaque.

k	Item KNN_c	Item KNN_g	Item KNN_n
50	0,920931	0,93716212***	0,9210632
100	0,925850	0,93667148***	0,92673996
150	0,928155	0,93656632***	0,92937804
200	0,929220	0,9365387***	0,93052312
250	0,929703	0,93652094**	0,9312018*
300	0,929922	0,93650324**	0,9315182*
350	0,930044	0,93649976**	0,9316609*
400	0,930103	0,93649466**	0,93174694*
450	0,930134	0,93649258**	0,93179078*
500	0,930145	0,93649254**	0,9318146*
550	0,930146	0,93649254**	0,9318224*
600	0,930146	0,93649254**	0,93182254*

A seguinte notação foi usada para a significância estatística: (*) quando $p\text{-value} < 0,1$; (**) quando $p\text{-value} < 0,05$; e (***) quando $p\text{-value} < 0,001$.

Tabela 8 – Resultados obtidos no cenário de partida fria. Os melhores resultados estão em destaque.

k	Item KNN_c	Item KNN_g	Item KNN_n
50	1,040186	1,093502	1,10677***
100	1,052794	1,095062	1,10677**
150	1,058968	1,104014	1,10677**
200	1,066856	1,10891	1,10677**
250	1,070208	1,11287	1,10677*
300	1,072804	1,114518	1,10677
350	1,074178	1,114952	1,10677
400	1,075416	1,115158	1,10677
450	1,076382	1,115172	1,10677
500	1,076408	1,115172	1,10677
550	1,076408	1,115172	1,10677
600	1,076408	1,115172	1,10677

A seguinte notação foi usada para a significância estatística: (*) quando $p\text{-value} < 0,1$; (**) quando $p\text{-value} < 0,05$; e (***) quando $p\text{-value} < 0,001$.

5.4 Considerações Finais

Neste capítulo foram avaliados os experimentos realizados para comparar o desempenho de versões de SRs que usam ou não descrições semânticas para calcular a similaridade entre filmes. Foi ainda analisado o impacto que o modo como o SR seleciona os filmes similares para prever notas causa no seu desempenho, quando o mesmo adota o cálculo de similaridade proposto neste trabalho.

De modo geral, os experimentos mostraram que os SRs que usam descrições semânticas alcançam melhores resultados no cenário de partida fria, chegando a superar o desempenho de SRs que não usam descrições semânticas na medida que aumentam o número de itens

semelhantes avaliados para prever notas.

O próximo capítulo conclui este documento, elencando as principais contribuições, trabalhos realizados e apresentando algumas propostas de trabalhos futuros.

CONCLUSÕES E TRABALHOS FUTUROS

O capítulo anterior apresentou todos os experimentos realizados no período deste trabalho e uma análise detalhada dos resultados obtidos. Assim, este capítulo conclui este documento, retomando um resumo do trabalho, suas contribuições, apresentando os artigos gerados e publicados no período e algumas propostas de trabalhos futuros.

6.1 Resumo do Trabalho

Este trabalho teve como objetivo investigar a influência que as descrições semânticas construídas a partir de informações extraídas de bases de conhecimento na Web podem ter na resolução de problemas provenientes da ausência de semântica na descrição de itens em sistemas de recomendação.

A arquitetura projetada teve como referência o domínio de filmes e contempla, entre outras características, um módulo de extração de categorias semânticas da DBpedia, e um módulo que realiza o cálculo de similaridade entre os filmes do acervo. As categorias semânticas da DBpedia constituem uma estrutura hierárquica que, nessa arquitetura, é representada por um grafo. Isso possibilita que a similaridade semântica seja calculada por meio da distância semântica entre as categorias associadas a dois filmes distintos.

Durante o desenvolvimento deste trabalho foram realizados experimentos que compararam o desempenho de sistemas que utilizaram ou não informações semânticas. Os experimentos foram executados de modo a possibilitar a análise do desempenho de cada sistema na predição de notas no cenário normal e de partida fria. Além disso, foi possível avaliar o impacto causado no desempenho do sistema, conforme é alterada a maneira de selecionar os filmes semelhantes para a predição de notas.

De modo geral, verificou-se que os sistemas que usaram descrições semânticas obtiveram melhor desempenho no cenário de partida fria, podendo alcançar uma melhora de até 10% no

RMSE, quando comparados com outros sistemas cuja predição de notas é baseada na correlação de notas. Observou-se também uma tendência de melhoria no desempenho desses sistemas na medida que o número de filmes semelhantes usados na predição de notas aumenta, chegando a ultrapassar o desempenho dos sistemas que não usam descrições semânticas.

Os resultados geram expectativas positivas em relação ao uso de descrições semânticas em sistemas de recomendação e do uso de técnicas para calcular a similaridade semântica, incluindo a proposta deste trabalho.

6.2 Contribuições da Pesquisa

As principais contribuições desta pesquisa são:

- avaliação de diferentes técnicas de cálculo de similaridade entre itens e comparação com heurísticas consagradas na área, aplicadas diretamente em sistemas de recomendação;
- investigação da contribuição que dados com maior carga semântica podem trazer para um sistema de recomendação, em particular a filtragem colaborativa baseada em vizinhança;
- apresentação de uma nova técnica de cálculo de similaridade entre itens que explora descrições semânticas hierárquicas extraídas de uma base de conhecimento;
- aplicação das técnicas de similaridade semântica em um cenário de partida fria, visando reduzir problemas existentes oriundos da pouca disponibilidade de avaliações sobre novos itens; e
- disponibilização de uma infra-estrutura de recomendação semântica contendo uma base de filmes associada a uma base de conhecimento, favorecendo novas pesquisas na área.

Nas seções seguintes apresentam-se as conclusões finais e possíveis aplicações para o produto deste trabalho, e as publicações realizadas durante seu desenvolvimento.

6.2.1 Conclusões e Aplicações

Neste trabalho foram realizados experimentos que analisaram sete algoritmos de recomendação, sendo que cinco deles são derivações do ItemKNN que calculam a similaridade entre itens com técnicas diferentes. Dentre os sete algoritmos, apenas dois não trabalham com descrições semânticas.

Os SRs desenvolvidos a partir desses algoritmos foram submetidos a experimentos que os avaliaram em dois cenários distintos: normal e de partida fria. Verificou-se que no cenário normal a técnica de cálculo da similaridade baseada em correlação de avaliações é melhor que as demais comparadas neste trabalho para a predição de notas.

Contudo, no cenário de partida fria o SR ItemKNN tende a predizer a nota média global para os itens novos porque esses itens não possuem nenhuma avaliação e, conseqüentemente, o cálculo de similaridade baseado em correlação de avaliações não poderá ser realizado. Nesse cenário, pode-se observar que sempre houve pelo menos um dos SRs que usam descrições semânticas com melhor desempenho que o SR ItemKNN em parte do intervalo de itens vizinhos analisados. Verificou-se que o desempenho dos SRs que usam descrições semânticas são sensíveis à variação do conjunto de itens vizinhos, usado na predição de notas, no cenário de partida fria. Em especial o SR Sim. LDSDiExt apresentou uma tendência de melhoria a medida que o conjunto de vizinhos aumenta. Essa característica explica o seu melhor desempenho ao trabalhar com o conjunto de vizinhos $S_i^k(u)$ à trabalhar com o $R_i^k(u)$, visto que o último é mais restritivo e frequentemente possui menos itens.

Também foi observado como um SR que implementa o algoritmo ItemAttributeKNN pode ser beneficiado, recebendo como entrada um volume maior de atributos relacionados aos itens, advindos de uma base de conhecimento.

Diante das análises realizadas, é possível concluir que os sistemas de recomendação podem se beneficiar do uso de descrições semânticas advindas de bases de conhecimento no cenário de partida fria. Assim sendo, a maior contribuição deste trabalho está na proposta de uma nova técnica de cálculo de similaridade no cenário de partida fria.

6.2.2 Limitações do Sistema Proposto

O cálculo da similaridade semântica proposto na Equação 4.4 do Capítulo 4, exige um grande esforço computacional. Para realizar o cálculo entre dois itens distintos, a e b , é necessário listar todas as categorias semânticas diretas de cada item e encontrar o menor caminho entre a primeira categoria de a e a primeira categoria de b , a primeira categoria de a e a segunda categoria de b , sucessivamente. Após encontrar todos os menores caminhos entre todas as categorias de a e b , eles são contabilizados e a distância semântica entre os itens a e b é calculada conforme a Equação 4.3 para que, enfim, a similaridade semântica seja calculada. Para realizar esse processo entre todos os itens da base de dados MovieLens 100k foram necessárias algumas semanas com um computador de 12 núcleos (core i7 @ 3,4 GHz) e 16 GB de memória RAM, dedicado para os cálculos.

Conseqüentemente não foi viável realizar o cálculo da similaridade semântica explorando mais informações disponíveis na base de conhecimento. Entretanto, novas estratégias podem ser estudadas a fim de filtrar as categorias relevantes para o cálculo de modo a reduzir o número de categorias envolvidas e, conseqüentemente, o custo computacional. As técnicas existentes sobre predição de *links* em redes complexas podem ajudar na filtragem de categorias semânticas (GAO *et al.*, 2015).

O sistema de recomendação que teve implementado o cálculo de similaridade semântica

proposto neste trabalho alcançou desempenho superior aos demais somente em casos específicos, como na partida fria a partir de um determinado número de vizinhos, por exemplo. Foram realizados alguns experimentos nos quais o cálculo de similaridade proposto foi combinado com a correlação por notas com o objetivo de obter melhores desempenhos no cenário normal e diminuir a sobre-especialização na recomendação de itens. Porém, não se obteve sucesso. Sendo assim, é necessário o estudo de combinações diferentes entre as duas técnicas para que se alcance melhores desempenhos em casos gerais.

6.2.3 *Trabalhos Submetidos e Publicados*

- Publicação de artigo curto:

Título: Collaborative Filtering Based on Semantic Distance Among Items.

Autores: Silva Júnior, Salmo Marques e Manzato, Marcelo Garcia

Conferência: WebMedia '15 Proceedings of the 21st Brazilian Symposium on Multimedia and the Web

Data: 27 a 30 de outubro de 2015

Local: Manaus, AM, Brasil

Abstract: This paper proposes a new method to compute the semantic distance among items in collaborative filtering based on k -nearest neighbors. This approach predicts the rating that a user u would give to an item i calculating the similarity between i and other items rated by u . This items' similarity is obtained using a semantic distance metric proposed in this paper. The technique exploits ontologies available on the Web through the Linked Open Data. This is possible because they have semantic descriptions, structured by links, that define a knowledge domain. The equation to calculate the semantic distance is an extension of a related work. We propose to assign weight to links to show the specificity of item's categories. Our proposal was evaluated with a movies dataset and it was shown that significant improvements can be achieved when compared to the baseline without weighted links.

- Submissão de artigo completo (sob avaliação):

Título: Addressing the Cold Start Problem in Recommender Systems with Semantic Information.

Autores: Silva Júnior, Salmo Marques e Manzato, Marcelo Garcia

Conferência: 2017 IEEE/WIC/ACM International Conference on Web Intelligence

Data: 23 a 26 de agosto de 2017

Local: Leipzig, Alemanha

Abstract: This paper proposes a new method for computing semantic items' similarities in collaborative filtering based on neighborhood models. Items' similarity is calculated from their semantic distance, which in turn, is based on the hierarchical structure of semantic categories associated with items. These semantic categories are available in open knowledge bases, which use ontologies to structure data in a way to be automatically interpretable by machines. Particularly in this paper, we investigate how semantic information obtained from knowledge bases can support the recommendation under the cold start scenario, in which new items have very low information to be used by the recommender algorithm.

6.3 Trabalhos Futuros

Findada a realização deste trabalho, há espaço para a sua continuação e aperfeiçoamento. Esta seção aponta sugestões de trabalhos futuros que poderão aprofundar o estudo iniciado neste trabalho:

- construção de perfis de usuários a partir das descrições semânticas dos itens avaliados pelos usuários;
- pesquisar modos de comparar diretamente os perfis semânticos dos usuários com as descrições semânticas dos itens;
- realização de uma avaliação *online* de um sistema usando as técnicas propostas neste trabalho, analisando-se outras variáveis relacionadas à avaliação de um usuário real;
- combinar o uso de descrições semânticas com a correlação baseada em avaliações no cálculo da similaridade em busca de melhor desempenho no cenário normal e amenizar a sobre-especialização, inerente a sistemas que adotam o modelo de vizinhança baseado em itens;
- estudar técnicas para filtrar as categorias semânticas de maior relevância para a descrição dos itens a fim de viabilizar o cálculo da similaridade em bases de dados maiores;
- aplicação do sistema proposto em outros domínios de dados, como músicas, livros e eventos.

REFERÊNCIAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, 2005. Citado nas páginas 23, 30, 33 e 36.

_____. Context-aware recommender systems. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 217–253. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_7>. Citado na página 36.

AGGARWAL, C. C. Neighborhood-based collaborative filtering. In: _____. **Recommender Systems: The Textbook**. Cham: Springer International Publishing, 2016. p. 29–70. ISBN 978-3-319-29659-3. Disponível em: <http://dx.doi.org/10.1007/978-3-319-29659-3_2>. Citado na página 34.

AHLERS, D. Assessment of the accuracy of geonames gazetteer data. In: **Proceedings of the 7th Workshop on Geographic Information Retrieval**. New York, NY, USA: ACM, 2013. (GIR '13), p. 74–81. ISBN 978-1-4503-2241-6. Disponível em: <<http://doi.acm.org/10.1145/2533888.2533938>>. Citado na página 48.

ALLEMANG, D.; HENDLER, J. **Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL**. 2. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 9780123859655, 9780123859662. Citado na página 41.

AMATRIAIN, X.; JAIMES*, A.; OLIVER, N.; PUJOL, J. M. Data mining methods for recommender systems. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 39–71. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_2>. Citado na página 39.

ANTONIOU, G.; VANHARMELEN, F. **A Semantic Web Primer**. Cambridge, MA, USA: MIT Press, 2004. ISBN 0262012103. Citado nas páginas 15, 25, 42, 44 e 47.

ARAÚJO, R. D. **Distribuição e Personalização de Conteúdo Multimídia em Ambientes Educacionais Ubíquos**. Tese (Doutorado) — Faculdade de Computação da Universidade Federal de Uberlândia, Brasil, 2013. Citado nas páginas 36 e 37.

BAUER, F.; KALTENBÖCK, M. **Linked Open Data: The Essentials**. Vienna, Austria: edition mono/monochrom, 2012. Citado nas páginas 25, 48 e 63.

BILLE, P. A survey on tree edit distance and related problems. **Theoretical Computer Science**, v. 337, p. 217–239, 2005. Citado nas páginas 53 e 55.

BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; GUTIÉRREZ, A. Recommender systems survey. **Knowledge-Based Systems**, v. 46, p. 109–132, 2013. Citado nas páginas 32, 34 e 38.

BURKE, R.; RAMEZANI, M. Matching recommendation technologies and domains. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 367–386. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_11>. Citado na página 24.

- CHERKASSKY, B. V.; GOLDBERG, A. V.; RADZIK, T. Shortest paths algorithms: Theory and experimental evaluation. **Mathematical Programming**, v. 73, n. 2, p. 129–174, 1996. ISSN 1436-4646. Disponível em: <http://dx.doi.org/10.1007/BF02592101>. Citado na página 73.
- DESROSIERS, C.; KARYPIS, G. A comprehensive survey of neighborhood-based recommendation methods. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 107–144. ISBN 978-0-387-85820-3. Disponível em: http://dx.doi.org/10.1007/978-0-387-85820-3_4. Citado nas páginas 24, 33, 34, 35, 67 e 75.
- FELFERNIG, A.; FRIEDRICH, G.; JANNACH, D.; ZANKER, M. Developing constraint-based recommenders. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 187–215. ISBN 978-0-387-85820-3. Disponível em: http://dx.doi.org/10.1007/978-0-387-85820-3_6. Citado na página 36.
- GAO, F.; MUSIAL, K.; COOPER, C.; TSOKA, S. Link prediction methods and their accuracy for different social networks and network metrics. **Scientific Programming**, v. 2015, 2015. Disponível em: <http://dx.doi.org/10.1155/2015/172879>. Citado na página 91.
- GOLBREICH, C.; WALLACE, E. **OWL 2 Web Ontology Language New Features and Rationale (Second Edition)**. [S.l.], 2012. <http://www.w3.org/TR/2012/REC-owl2-new-features-20121211/>. Citado na página 47.
- GRAU, B. C.; WU, Z.; FOKOUE, A.; HORROCKS, I.; MOTIK, B. **OWL 2 Web Ontology Language Profiles (Second Edition)**. [S.l.], 2012. <http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/>. Citado na página 47.
- Guizzardi, G. **Ontological foundations for structural conceptual models**. Tese (Doutorado) — University of Twente, Enschede, 2005. Disponível em: <http://doc.utwente.nl/50826/>. Citado na página 45.
- HARISPE, S.; RANWEZ, S.; JANAQI, S.; MONTMAIN, J. Semantic measures based on rdf projections: Application to content-based recommendation systems. In: _____. **On the Move to Meaningful Internet Systems: OTM 2013 Conferences: Confederated International Conferences: CoopIS, DOA-TrusteD Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 606–615. ISBN 978-3-642-41030-7. Disponível em: http://dx.doi.org/10.1007/978-3-642-41030-7_44. Citado nas páginas 25 e 57.
- HEBELER, J.; FISHER, M.; BLACE, R.; PEREZ-LOPEZ, A. **Semantic Web Programming**. [S.l.]: Wiley Publishing, 2009. ISBN 047041801X, 9780470418017. Citado nas páginas 15, 25, 42, 44, 46 e 49.
- IJNTEMA, W.; GOOSSEN, F.; FRASINCAR, F.; HOGENBOOM, F. Ontology-based news recommendation. In: **Proceedings of the 2010 EDBT/ICDT Workshops**. New York, NY, USA: ACM, 2010. (EDBT '10), p. 16:1–16:6. ISBN 978-1-60558-990-9. Disponível em: <http://doi.acm.org/10.1145/1754239.1754257>. Citado na página 56.
- ISOTANI, S.; BITTENCOURT, I. I. **Dados Abertos Conectados**. [S.l.]: Novatec, 2015. Citado na página 42.
- KAGIE, M.; WEZEL, M. van; GROENEN, P. J. Map based visualization of product catalogs. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 547–576.

ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_17>. Citado na página 23.

KOREN, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: **Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2008. (KDD '08), p. 426–434. ISBN 978-1-60558-193-4. Disponível em: <<http://doi.acm.org/10.1145/1401890.1401944>>. Citado nas páginas 58 e 68.

KOREN, Y.; BELL, R. Advances in collaborative filtering. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 145–186. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_5>. Citado nas páginas 23 e 36.

LEACOCK, C.; CHODOROW, M. Combining local context and wordnet similarity for word sense identification. **WordNet: An electronic lexical database**, v. 49, n. 2, p. 265–283, 1998. Citado nas páginas 51 e 75.

LEHMANN, J.; ISELE, R.; JAKOB, M.; JENTZSCH, A.; KONTOKOSTAS, D.; MENDES, P. N.; HELLMANN, S.; MORSEY, M.; KLEEF, P.; AUER, S.; BIZER, C. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. **Semantic Web Journal**, v. 6, n. 2, p. 167–195, 2015. Citado na página 48.

LOPS, P.; GEMMIS, M. de; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 73–105. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_3>. Citado nas páginas 15, 30, 31, 32 e 52.

MAIDEL, V.; SHOVAL, P.; SHAPIRA, B.; TAIEB-MAIMON, M. Evaluation of an ontology-content based filtering method for a personalized newspaper. In: **Proceedings of the 2008 ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2008. (RecSys '08), p. 91–98. ISBN 978-1-60558-093-7. Disponível em: <<http://doi.acm.org/10.1145/1454008.1454024>>. Citado na página 56.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. **An Introduction to Information Retrieval**. Cambridge, UK: Cambridge University Press, 2009. Citado na página 32.

MANZATO, M. G. **Uma arquitetura de personalização de conteúdo baseada em anotações do usuário**. Tese (Doutorado) — Universidade de São Paulo, Brasil, 2010. Citado na página 31.

MENG, L.; HUANG, R.; GU, J. A review of semantic similarity measures in wordnet. **International Journal of Hybrid Information Technology**, v. 6, n. 1, p. 1–12, 2013. Citado nas páginas 15, 49, 50, 51 e 52.

MORSEY, M.; LEHMANN, J.; AUER, S.; STADLER, C.; HELLMANN, S. Dbpedia and the live extraction of structured data from wikipedia. **Program: electronic library and information systems**, v. 46, n. 2, p. 157–181, 2012. Citado na página 48.

NOIA, T. D.; MIRIZZI, R.; OSTUNI, V. C.; ROMITO, D.; ZANKER, M. Linked open data to support content-based recommender systems. In: **Proceedings of the 8th International Conference on Semantic Systems**. New York, NY, USA: ACM, 2012. (I-SEMANTICS '12), p. 1–8. ISBN 978-1-4503-1112-0. Disponível em: <<http://doi.acm.org/10.1145/2362499.2362501>>. Citado na página 56.

OSTUNI, V. C.; GENTILE, G.; NOIA, T. D.; MIRIZZI, R.; ROMITO, D.; SCIASCIO, E. D. Mobile movie recommendations with linked data. In: _____. **Availability, Reliability, and Security in Information Systems and HCI: IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2013, Regensburg, Germany, September 2-6, 2013. Proceedings.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 400–415. ISBN 978-3-642-40511-2. Disponível em: <http://dx.doi.org/10.1007/978-3-642-40511-2_29>. Citado nas páginas 24, 25 e 55.

OSTUNI, V. C.; NOIA, T. D.; MIRIZZI, R.; SCIASCIO, E. D. A linked data recommender system using a neighborhood-based graph kernel. In: _____. **E-Commerce and Web Technologies: 15th International Conference, EC-Web 2014, Munich, Germany, September 1-4, 2014. Proceedings.** Cham: Springer International Publishing, 2014. p. 89–100. ISBN 978-3-319-10491-1. Disponível em: <http://dx.doi.org/10.1007/978-3-319-10491-1_10>. Citado na página 57.

OWL 2 Web Ontology Language Document Overview (Second Edition). [S.l.], 2012. [Http://www.w3.org/TR/2012/REC-owl2-overview-20121211/](http://www.w3.org/TR/2012/REC-owl2-overview-20121211/). Citado na página 46.

PASSANT, A. Dbrec: Music recommendations using dbpedia. In: **Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II.** Berlin, Heidelberg: Springer-Verlag, 2010. (ISWC'10), p. 209–224. ISBN 3-642-17748-4, 978-3-642-17748-4. Disponível em: <<http://dl.acm.org/citation.cfm?id=1940334.1940349>>. Citado na página 56.

_____. **Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations.** 2010. Disponível em: <<http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1147/1456>>. Citado nas páginas 52, 53, 64, 65, 66 e 75.

PESQUITA, C.; FARIA, D.; FALÇÃO, A. O.; LORD, P.; COUTO, F. M. Semantic similarity in biomedical ontologies. **PLoS Computational Biology**, v. 5, n. 7, 2009. Citado na página 53.

RENDLE, S. Item recommendation. In: _____. **Context-Aware Ranking with Factorization Models.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 69–84. ISBN 978-3-642-16898-7. Disponível em: <http://dx.doi.org/10.1007/978-3-642-16898-7_6>. Citado na página 35.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: _____. **Recommender Systems Handbook.** Boston, MA: Springer US, 2011. p. 1–35. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_1>. Citado nas páginas 29, 30 e 36.

ROWE, M. Semanticsvd++: Incorporating semantic taste evolution for predicting ratings. In: **Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01.** Washington, DC, USA: IEEE Computer Society, 2014. (WI-IAT '14), p. 213–220. ISBN 978-1-4799-4143-8. Disponível em: <<http://dx.doi.org/10.1109/WI-IAT.2014.36>>. Citado nas páginas 24 e 58.

SHANI, G.; GUNAWARDANA, A. Evaluating recommendation systems. In: _____. **Recommender Systems Handbook.** Boston, MA: Springer US, 2011. p. 257–297. ISBN 978-0-387-85820-3. Disponível em: <http://dx.doi.org/10.1007/978-0-387-85820-3_8>. Citado nas páginas 37, 38 e 40.

SNOEK, C. G.; WORRING, M. Multimodal video indexing: A review of the state-of-the-art. **Multimedia Tools and Applications**, v. 25, n. 1, p. 5–35, 2005. ISSN 1573-7721. Disponível em: <http://dx.doi.org/10.1023/B:MTAP.0000046380.27575.a5>. Citado na página 31.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: Principles and methods. **Data and Knowledge Engineering**, v. 25, p. 161–197, 1998. Citado na página 45.

VICTOR, P.; COCK, M. D.; CORNELIS, C. Trust and recommendations. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 645–675. ISBN 978-0-387-85820-3. Disponível em: http://dx.doi.org/10.1007/978-0-387-85820-3_20. Citado na página 36.

WU, Z.; PALMER, M. Verbs semantics and lexical selection. In: **Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994. (ACL '94), p. 133–138. Disponível em: <http://dx.doi.org/10.3115/981732.981751>. Citado nas páginas 52 e 75.

ZHANG, F.; YUAN, N. J.; LIAN, D.; XIE, X.; MA, W.-Y. Collaborative knowledge base embedding for recommender systems. In: **Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2016. (KDD '16), p. 353–362. ISBN 978-1-4503-4232-2. Disponível em: <http://doi.acm.org/10.1145/2939672.2939673>. Citado nas páginas 24, 25 e 57.

ZHANG, K. A constrained edit distance between unordered labeled trees. **Algorithmica**, v. 15, p. 205–222, 1996. Citado na página 54.

CONSULTAS SPARQL

Este apêndice relaciona as consultas SPARQL realizadas pelo sistema de recomendação proposto para extrair as categorias semânticas associadas aos recursos da DBpedia que representam os filmes da base MovieLens.

Prefixos usados nas consultas SPARQL:

```
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX dbpprop: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX bds: <http://www.bigdata.com/rdf/search#>
```

Consulta SPARQL para retornar todos os recursos do tipo Film da DBpedia cuja IRI inicia com a letra “a” :

```
SELECT ?film WHERE{
    ?film rdf:type dbpedia-owl:Film.
    FILTER regex(?film, '^http://dbpedia.org/resource/a', 'i')
}
```

Consulta SPARQL para obter a propriedade name e foaf:name de um recurso a partir de sua IRI:

```
SELECT DISTINCT ?name WHERE{
    <IRI> <http://dbpedia.org/property/name> ?name.
```

```

}
SELECT DISTINCT ?name WHERE{
    <IRI> <http://xmlns.com/foaf/0.1/name> ?name.
}

```

Consulta SPARQL para obter a propriedade abstract de um recurso a partir de sua IRI:

```

SELECT DISTINCT ?abstract WHERE{
    <IRI> <http://dbpedia.org/ontology/abstract> ?abstract.
}

```

Consulta SPARQL para obter a propriedade release de um recurso a partir de sua IRI:

```

SELECT DISTINCT ?release WHERE{
    <IRI> <http://dbpedia.org/property/released> ?release.
}

```

Consulta SPARQL associar filme “Top Gun” ao recurso correspondente por meio das propriedades name, foaf:name, release e abstract:

```

SELECT DISTINCT ?film ?score ?scoreAbstract WHERE{
{
    ?name bds:search "Top Gun".
    ?name bds:relevance ?score.
    ?film dbpprop:name ?name.
    ?abstract bds:search "Top Gun 1986".
    ?abstract bds:relevance ?scoreAbstract.
    ?film dbpedia-owl:abstract ?abstract.
} UNION {
    ?name bds:search "Top Gun".
    ?name bds:relevance ?score.
    ?film foaf:name ?name.
    ?abstract bds:search "Top Gun 1986".
    ?abstract bds:relevance ?scoreAbstract.
    ?film dbpedia-owl:abstract ?abstract.
}
} ORDER BY DESC(?score) DESC(?scoreAbstract)

```

Consulta SPARQL para obter as categorias semânticas diretas de um recurso a partir de sua IRI:

```

SELECT DISTINCT ?category WHERE{
    <IRI> dcterms:subject ?category.
}

```