



Recomendação com Similaridade Semântica Ponderada por
Links de Recursos na DBpedia

Por

Lucas Lara Marotta

Trabalho de Graduação



Universidade Federal da Bahia
wiki.dcc.ufba.br/DCC/

SALVADOR, Abril/2021



Universidade Federal da Bahia
Departamento de Ciência da Computação

Lucas Lara Marotta

Recomendação com Similaridade Semântica Ponderada por Links de Recursos na DBpedia

Trabalho apresentado ao Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Frederico Araujo Durão*

SALVADOR, Abril/2021

Dedico esta dissertação à minha família, amigos e professores que me deram todo o apoio necessário para chegar até aqui.

It matters not how strait the gate, how charged with punishments the scroll, I am the master of my fate, I am the captain of my soul

—WILLIAM ERNEST HENLEY

Resumo

Por facilitar a procura por informação no mar de dados da Internet, Sistemas de Recomendação (SR) são extremamente populares na Web. Usualmente SR tentam prever avaliações de usuários sobre um item desconhecido, para gerar recomendações personalizadas. Nesses sistemas, em especial os baseados em conteúdo, características de itens são processadas para identificar outros itens relacionados, mas é comum que sejam negligenciadas relações semânticas entre eles. Focar unicamente em dados sintáticos, favorece o efeito bolha, o que é caracterizado por usuários sendo menos expostos a itens relevantes e inesperados, algo desejável num SR. Encontrar itens com similaridades semânticas pode minimizar esse efeito, já que provê uma ainda relevante, porém mais abrangente similaridade. Nesse sentido, este trabalho propõe um Sistema de Recomendação, baseado em conteúdo, com Similaridade Semântica Ponderada por Links de Recursos (RLWS) na *DBpedia*. O objetivo é verificar que resultados são obtidos pela comparação de termos da sinopse dos filmes, onde RLWS analisa relações semânticas diretas e indiretas entre eles, usando o DBpedia. Sendo assim, foi conduzido um experimento comparando RLWS com a conhecida similaridade do cosseno. Considerando um conjunto de cinco itens ($k = 5$), o sistema proposto melhorou a precisão média (MAP) em 51%, quando privilegiado relacionamentos indiretos, e 27% para os diretos. Além disso, a proposta também melhorou o desempenho da métrica MRR em 26% privilegiando relacionamentos indiretos, e de 11% para diretos.

Palavras-chave: Sistema de Recomendação, Recomendação Baseada em Conteúdo, Web Semântica, Similaridade Semântica

Abstract

With the goal to facilitate the efforts when searching information on the Web, Recommender Systems (RS) have become extremely popular in recent years on the Web. Usually, RS try to predict the user's evaluation over an unknown item to generate personalized recommendations. Those systems, especially those content-based, process syntactic data (e.g., item features) to identify new related items, but often neglect the semantic similarities between them. Focusing only on syntactic data favors the “bubble filter effect” - an effect characterized by the user not being exposed to unexpected and relevant items, a desired feature for RS. Finding items with semantic similarities minimizes the “bubble filter effect” since it can provide a broader and more relevant similarity. In this sense, this work proposes a Recommender System (content-based) with a Resource Link-Weighted Similarity (RLWS), using *DBpedia*. The proposed system verifies which results are obtainable by comparing terms from film synopses, and then RLWS analyses the direct and indirect semantic relations between them, using the DBpedia. We conduct an experimental evaluation comparing the RLWS with the well-known cosine similarity. Considering a result set of five items ($k = 5$), the proposed system improves the MAP performance by 51% when weighting more indirect relationships between terms, and for the direct relationships by 27%. In addition, the proposal improves the MRR performance in 26% weighting more indirect relationships, and 11% using the direct ones.

Keywords: Recommender Systems, Content-Based Recomendation, Semantic Similarity, Semantic Web

Sumário

Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Acrônimos	xxi
Lista de Códigos Fonte	xxiii
1 Introdução	1
1.1 Motivação	2
1.2 Problema	4
1.3 Objetivos da Solução Proposta	4
1.4 Estrutura	5
2 Sistemas de Recomendação	7
2.1 Histórico	7
2.2 Conceitos	9
2.3 Tarefas de um Sistema de Recomendação	10
2.4 Técnicas de Recomendação	12
2.4.1 Filtragem Colaborativa	13
2.4.2 Filtragem Baseada em Conteúdo	14
2.4.3 Comparação das Técnicas de Recomendação	15
2.5 Aplicações de Sistemas de Recomendação	16
2.5.1 Netflix	16
2.5.2 Skoob	18
2.6 Sumário	19
3 Web Semântica	21
3.1 Arquitetura e formato de dados	22
3.1.1 RDF	23
3.1.2 SPARQL	24
3.1.3 OWL	24
Estrutura de um documento	26
3.1.4 Estrutura na rede semântica	28
3.2 Dados ligados	29

3.2.1	Linked Open Data	30
3.3	Similaridade Semântica	33
3.3.1	Medidas de Similaridade Semântica	34
Baseadas em estrutura	34	
Baseadas em conteúdo	35	
Baseadas em características ou recursos	36	
3.4	Projetos na Web Semântica	36
3.4.1	DBPedia	36
3.4.2	Google Knowledge Graph	38
3.5	Sumário	40
4	Recomendação com similaridade semântica ponderada por links de recursos na DBPedia	41
4.1	Arquitetura	42
4.2	Processo de Recomendação	42
4.3	Modelo de dados	44
4.3.1	Banco de dados	44
4.3.2	Modelo de filmes	45
4.3.3	Modelo de usuários	47
4.3.4	Preparação dos dados para recomendação	48
4.4	Modelo de Recomendação	51
4.4.1	Equação para similaridade semântica	52
4.4.2	Algoritmo da recomendação	54
4.5	Estrutura de Cache para Recomendação	59
4.6	Tecnologias	62
4.6.1	JAVA	62
4.6.2	Spring Boot	63
4.6.3	HTML, CSS, Javascript	63
4.6.4	MySQL	64
4.6.5	Apache Jena	64
4.6.6	Apache OpenNLP	65
4.6.7	Apache Lucene	65
4.7	Sumário	65
5	Avaliação	67
5.1	Metodologia	67

5.2	Conjunto de dados	71
5.3	Métricas de avaliação	73
5.3.1	Precision	73
5.3.2	Mean Average Precision (MAP)	75
5.3.3	Mean Reciprocal Rank (MRR)	75
5.4	Resultados	76
5.4.1	MAP	78
5.4.2	MRR	79
5.5	Discussão dos resultados	81
5.6	Pontos de melhorias	81
5.7	Sumário	83
6	Conclusão	85
6.1	Contribuições	86
6.2	Trabalhos Futuros	86
6.3	Sumário	87
Referências Bibliográficas		88

Lista de Figuras

2.1	Exemplo de lista de vídeos em alta no YouTube (2017)	8
2.2	Recomendação de Filmes no serviço Netflix. Figura elaborada pelo autor (2017).	17
2.3	Página de avaliação do livro no Skoob. Figura elaborada pelo autor (2017).	18
3.1	Exemplo do grafo RDF (RDF, 2017)	23
3.2	Exemplo do grafo da tripla sujeito predicado objeto (Web, 2009)	24
3.3	Camadas na rede semântica. (Júnio César de Lima, 2005)	28
3.4	Camadas na rede semântica. Figura elaborado pelo autor de acordo com a publicação de Berners-Lee (2008)	29
3.5	Sistema de avaliação do LOD (Berners-Lee, 2008)	31
3.6	Diagrama da nuvem dos dados ligados (Andrejs Abele, 2020)	32
3.7	Ilustração da arquitetura do DBPèdia (DBPedia, 2017)	38
3.8	Ilustração do sumário de dados mapeados no Google Knowledge Graph.	39
4.1	Fluxo das camadas do sistema de recomendação.	43
4.2	Diagrama da modelagem dos dados extraído do banco MySQL	46
4.3	Segmentação de tarefas no NLP. (Guts, 2016)	49
4.4	Imagen que retrata os links diretos saintes e entrantes de um recurso.	53
4.5	Imagen que retrata os links indiretos saintes de um recurso.	55
4.6	Imagen que exemplifica o fluxo da recomendação.	61
5.1	Gráfico da relação do tamanho do modelo do usuário (quantidade de termos usados) com o tempo de processamento para recomendação de um usuário. Todas as comparações de termos estão em <i>cache</i> . Execução numa máquina com processador <i>i7 6700K</i> , 16GB RAM, <i>Windows 10</i>	69
5.2	Tabela de tipos de erros baseada na ilustração de Jannach <i>et al.</i> (2010).	74
5.3	Exemplo do cálculo do MAP.	75
5.4	Exemplo do cálculo do MRR.	76
5.5	MAP - Gráfico com linhas dos três experimentos nos testes online e offline	79
5.6	MAP - Gráfico de caixa dos três experimentos nos testes online e offline	79
5.7	MRR - Gráfico com linhas dos três experimentos nos testes online e offline	80
5.8	MRR - Gráfico de caixa dos três experimentos nos testes online e offline	80

Lista de Tabelas

3.1	Recorte da tabela de dados de triplas de entidades mapeadas no DBPedia. (DBPedia, 2014)	37
4.1	Relação das tags das partes do discurso	50
4.2	Exemplos da geração de tokens	50
5.1	Contagem dos dados utilizados durante os testes.	71
5.2	Estatística da cobertura dos dados dos links de recursos na DBPedia . .	72
5.3	Contagem da relação entre recursos utilizados durante os experimentos.	73
5.4	Tabela de amostra de comparações entre termos usando Resource Link-Weighted Similarity (RLWS).	77

Lista de Acrônimos

NFC	Need For Cognition
API	Application Programming Interface
SR	Sistema de Recomendação
CF	Collaborative Filtering
CBF	Content Based Filtering
DVD	Digital Video Disc
RMSE	Root Mean Square Error
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
OWL	Ontology Web Language
URI	Universal Resource Identifier
SQL	Structured Query Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
LOD	Linked Open Data
MIS	Most Informative Subsume
MVC	Model View Controller
VM	Virtual Machine
SO	Sistema Operacional

OOP	Object Oriented Programming
IOC	Inversion of Control
ORM	Object Relational Mapping
CSS	Cascading Style Sheets
SGBD	Sistema de Gerenciamento de Banco de Dados
NER	Name Entity Recognition
NLP	Natural Language Processing
RLWS	Resource Link-Weighted Similarity
IDF	Inverse Document Frequency
TFIDF	Term Frequency and Inverse Document Frequency
RAM	Random Access Memory
MAP	Mean Average Precision
RESIM	Resource Similarity
LDSD	Linked Data Semantic Distance
POS	Part of Speech
MRR	Mean Reciprocal Rank
WMD	Word Mover's Distance

Lista de Códigos Fonte

3.1	Exemplo de consulta na linguagem SPARQL	25
3.2	Exemplo do topo de um documento OWL	26
3.3	Exemplo do cabeçalho XML de um documento OWL	27
3.4	Exemplo de propriedades transitivas no OWL	27
3.5	Exemplo do cabeçalho de uma ontologia	27
4.1	Consulta SPARQL para contagem de links diretos	54
4.2	Consulta SPARQL para contagem de links indiretos.	55
4.3	Consulta SPARQL para contagem de links diretos (saíntes e entrantes) entre dois recursos.	56
4.4	Consulta SPARQL para contagem de links indiretos (saíntes) entre dois recursos.	56

1

Introdução

I'm so fast with technology. People think it all seems too much, but we'll get used to it. I'm sure it all seemed too much when we were learning to walk.

—YOKO ONO

A expansão dos meios de comunicação através da Internet possibilitou o rápido acesso a todo tipo de informação de diversas áreas do mundo a todo lugar. Consumir conteúdo digital tornou-se atividade comum no dia das pessoas. Conforme mais se expande o acesso as mídias digitais mais conteúdo é gerado e mais está disponível para ler, ver, ouvir e interagir. Segundo [Walker \(2014\)](#) chegamos a uma era em que trafegamos uma quantidade enorme de dados que rapidamente perde-se a escala e cognição para o humano. Qual o significado de 400 milhões de tweets¹ por dia? Usar o pensamento empírico de grandes matemáticos como “to measure is to know” (William Thomson) torna-se especialmente difícil com o volume de informações produzidas neste século. Com a quantidade de dados disponíveis não é irônico ouvir “não sei qual filme assistir”, pois apesar do fácil acesso existe uma grande sobrecarga a qual expõe o usuário a um mar de dados ([Wellman, 2013](#)), dificultando o acesso ao conteúdo que seja mais relevante.

O volume de informações apresenta-se como um obstáculo ao usuário que deseja consumir algum tipo conteúdo. Compras online possuem milhares de opções e nem todos estão dispostos a passar um grande tempo olhando o catálogo disponível. Uma das razões pela preferência de compra pela Internet é justamente a “falta de tempo”, conforme revela análise de [Baubonienė and Gulevičiūtė \(2015\)](#). Dessa forma, é natural que o usuário

¹Tweet é o nome utilizado para designar as publicações feitas na rede social do Twitter (<https://www.merriam-webster.com/dictionary/tweet>)

CAPÍTULO 1. INTRODUÇÃO

recorra a alternativas para se guiar pelas informações e encontrar mais facilmente aquilo que lhe é mais útil. Para minimizar o obstáculo que o volume de informações se opõem, é comum apelar para ajuda de conhecidos, parentes, amigos, como apontado pela pesquisa de [Baubonienė and Gulevičiūtė \(2015\)](#), onde um dos fatores relacionados ao consumidor que influenciam a opção pela compra pela Internet são as recomendações de outros usuários.

A larga difusão da Internet, principalmente pela Web, também cria um desafio pela busca de informação. Sistemas populares de recuperação de informação, como Google, amenizam o problema ([Isinkaye et al., 2015](#)), mas são deficientes quanto a personalização e priorização da informação em relação as preferências e interesses do usuário. Essa é uma das razões pelo grande aumento do desenvolvimento e procura por sistemas de recomendação. Sistemas de recomendação são sistemas de filtragem de itens que possuem objetivo de prever a avaliação e preferência do usuário ([Ricci et al., 2011](#)). Tais soluções contribuem ainda mais com a experiência do usuário no que diz ao conceito do Need For Cognition ([NFC](#)) que reflete na tendência de indivíduos em se engajar e aproveitar numa atividade ([Baubonienė and Gulevičiūtė, 2015](#)). Esses sistemas filtram os dados para reduzir o problema da sobrecarga de informação ([Konstan and Riedl, 2012](#)), podendo ser utilizados em diversos domínios como livros, filmes, músicas até para construir experiências em jogos online ([Alexandra, 2017](#)).

Os sistemas de recomendação tipicamente possuem três tipos de abordagens para as sugestões: filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida que leva em consideração as duas anteriores. Filtragem baseada em conteúdo são fundamentadas na descrição dos dados e nas preferências dos usuários ([Aggarwal, 2016b](#)). Desse modo, o objetivo desse trabalho é construir um sistema de recomendação baseado em conteúdo, utilizando dados não estruturados para definir as preferências de usuários, de tal forma a explorar relações semânticas entre eles, criando novas possibilidades para recomendar novos itens.

1.1 Motivação

Na similaridade em termos associados aos itens de comparação, é comum em sistemas de recomendação para domínios como de livros e filmes seja comparado termos como gênero e autor. Nesse caso, pode-se analisar se já foi demonstrado interesse pelo usuário em filmes com esses termos, para que assim o sistema aprenda e recomende novos itens com essas mesmas características. Entretanto, pode ser interessante para o usuário encontrar filmes que não sejam necessariamente do mesmo gênero ou autor, mas que

1.1. MOTIVAÇÃO

possuam narrativas mais similares ou relacionadas. Como exemplo considere os filmes *Sucker Punch* ([Snyder and Snyder, 2011](#)) e *Labirinto do Fauno* ([del Toro et al., 2006](#)), possuem diferentes diretores e apesar de terem um tema de fantasia, se diferem bastante, pois o primeiro é um filme orientado para ação enquanto que o segundo é um drama que se passa num período de guerra. Na narrativa dos filmes é possível encontrar pontos de similaridade, como os dois tratarem de jovens garotas que entram num mundo fantasioso onde precisarão vencer uma série de desafios para superar dificuldades em tempos difíceis. Nesse sentido, analisar a similaridade do conteúdo da descrição de um filme que contenha um trecho da sua narrativa, pode levar ao usuário a sair do seu círculo tradicional de preferência, podendo contribuir com o NFC no uso de um sistema. Uma das propostas desse trabalho é explorar os resultados analisando a sinopse de filmes, buscando relações que vão além de uma relação sintática, mas que possuam um contexto semântico relacionável.

Para analisar a similaridade de filmes observando a descrição da narrativa, será utilizado um serviço presente na web semântica, o DBpedia², para traçar formas de definir uma similaridade entre palavras presentes nas sinopses dos filmes, assim extraíndo relações semânticas de termos presentes nos textos. A ideia é de que a análise entre termos possa se estender de uma comparação simples e sintática, mas para uma comparação que envolva-os em contexto cujo seja possível definir um relacionamento ontológico. Como exemplo simples, quando compara-se uma frase como "O homem comprou aquele carro", com "A mulher adquiriu aquela moto", nota-se que embora tenham palavras diferentes, são semelhantes, podendo intuitivamente traçar uma proximidade entre termos de cada uma.

Como motivação para criar novas oportunidades e maneiras de descobrir filmes, sendo útil na busca por novos títulos pelos serviços na internet, um Sistema de Recomendação ([SR](#)) torna-se uma ferramenta valiosa. Na concepção dessas recomendações, é necessário criar um perfil do usuário para sugerir novos itens, que para a proposta deste trabalho se basearão nas suas preferências, ou seja em outros filmes que demonstrou interesse. Dessa forma, o foco deste trabalho é explorar a similaridade entre textos pequenos, um dado não estruturado, através da análise da relação semântica de termos extraídos, buscando suas referências em entidades nos serviços de dados ligados na web semântica (apresentados no Capítulo 3). O propósito dessa similaridade é para criar um sistema de recomendação, utilizando o domínio de filmes como ponto de interesse do público, embora as definições na construção desse sistema não sejam exclusivamente desenvolvidas para tal domínio.

²<http://wiki.dbpedia.org>

1.2 Problema

O problema deste trabalho trata-se da deficiência e dificuldade quanto a sistemas de recomendação sugerir itens quando apenas o conteúdo sintático é analisado desprezando relações semânticas presentes do conteúdo. É tradicional construir um SR apenas observando as características discretas dos itens, como propriedades e categorias, ou até de uma análise estatística, mas existe uma lacuna de informações que são desprezadas que podem ser extraídas analisando-as numa rede semântica de relações que as envolva.

Um problema também muito comum trata-se de como esses algoritmos de filtragem e personalização afetam as pessoas. O livro “The Filter Bubble” [Pariser \(2011\)](#) levanta preocupações sobre tais sistemas, onde o usuário fica fortemente sujeito a apenas ao mesmo tipo de conteúdo, ou informação que não venha criar conflitos de ponto de visão, o efeito bolha. Assim, utilizando um SR que apenas analisasse termos de gênero e título poderia deixar o usuário “preso” no círculo tradicional de preferência. Essa preocupação pode também ter um impacto negativo no sistema, já que é possível que os usuários viriam a encontrar outros conteúdos que poderiam ter interesse, mas são apenas encorajados àqueles mais tradicionais.

Comumente a busca de informação em sistemas de recuperação, como o Google³, possui dados dispersos e por muitas vezes desorganizados, além da carência de dados personalizados e priorizados que considere os interesses do usuário para encontrar o item desejado. Somando a isso, propondo um sistema em que também seja possível extrair a similaridade da descrição das narrativas dos filmes, analisando e buscando outras relações semânticas com as entidades presentes, pode-se trazer resultados que amenizem o efeito bolha. Esse trabalho tem um dos objetivos de explorar que decorrências podem ser obtidas levando em consideração essa abordagem.

1.3 Objetivos da Solução Proposta

Este trabalho propõem a criação de um SR baseado em conteúdo que também utilize uma análise da similaridade semântica (ver capítulo 3) entre os itens envolvidos. Para isso será proposto um modelo de recomendação que leve em consideração a descrição da narrativa do item. O objetivo é explorar que resultados podem ser obtidos realizando consultas ao serviço de dados ligados na web semântica, o DBpedia⁴. Para a construção do SR foi

³<https://www.google.com>

⁴<http://wiki.dbpedia.org>

escolhido o domínio de filmes, como motivador e exemplo de aplicação que tire proveito desse sistema. Através de uma pequena análise empírica na rede de relacionamento do autor, percebeu-se que pessoas tendem a informar mais das preferências de filmes do que de livros, outro fator para a escolha do domínio.

Com o acesso a esse serviço da web semântica, serão analisadas entidades procurando ontologias e relações presentes nas sinopses dos filmes, através dos dados ligados na DBpedia. Assim, pode ser comparada à similaridade de dois filmes através da presença ou relação de ontologias extraídas dos termos das sinopses dos filmes. Como exemplo, caso um filme possua na sinopse o termo *Morfeu* e o outro possua outras entidades sobre deuses mitológicos, como *Zeus*, poderá ser criado um nível de similaridade e relevância entre as sinopses dos filmes.

Os dados dos filmes e de preferências dos usuários serão obtidos através de uma coleta do projeto MovieLens⁵, um banco de dados o qual possui 20 milhões de avaliações, por 138.000 usuários em 27.000 filmes, sendo comumente utilizados para construção de um SR, o que facilita a realização de comparações. De posse dos dados serão definidos os modelos do usuário e dos itens para recomendação (ver 4.3), além de também propor uma nova métrica de similaridade semântica (ver 4.4) para a comparação de termos dos filmes e utilização no algoritmo de recomendação (ver 4.4.2).

1.4 Estrutura

Neste capítulo foi introduzido o problema e a motivação deste trabalho. Os próximos capítulos estão organizados da seguinte maneira: O Capítulo 2 apresenta os conceitos teóricos usados neste trabalho referentes a SR. O Capítulo 3 apresenta conceitos sobre a web semântica. O Capítulo 4 apresenta a proposta do SR com a resolução de um modelo de usuário que leve em consideração a descrição de itens, discutindo sua implementação. O Capítulo 5 apresenta a avaliação do sistema, conclusões e considerações finais.

⁵<https://movielens.org>

2

Sistemas de Recomendação

A Internet disponibiliza um enorme volume de informação para o usuário, o que cria um desafio pela busca de informação. Por esse problema, empresas cresceram construindo sistemas de recuperação e filtragem, para contornar a sobrecarga de informação, como é o caso do Google¹. Neste capítulo será apresentado um panorama sobre SR, introduzindo os principais conceitos, tarefas e processos que o caracterizam.

2.1 Histórico

Em razão da crescente dificuldade de usuários administrar a quantidade de informação, é comum decidir baseado em opiniões e recomendações de outros, especialmente quando há pouca experiência no assunto (Resnick and Varian, 1997). Conforme mais se expandia a tendência do uso de meios digitais de comunicação, mais rapidamente pessoas migraram de cartas para e-mails. A grande quantidade de e-mails acabava deixando o usuário imerso em documentos, dificultando o consumo do conteúdo. Em 1992, Xerox Palo Alto Research Center apresentou o sistema Tapestry (Goldberg *et al.*, 1992) na revista mensal ACM Communications², como proposta para lidar com o problema quantidade de e-mails.

O objetivo do sistema era prover listas de e-mails permitindo a inscrição dos usuários naquelas que fossem mais importantes. Alguns sistemas daquela época suportavam filtragem de e-mails baseado no seu conteúdo, mas os autores acreditavam que uma maneira mais eficiente seria com ajuda da avaliação de outros usuários. Interessante ressaltar que o termo “filtragem colaborativa” apresentado no artigo tornou-se comum, e só alguns anos depois surgiu a defesa do termo sistemas de recomendação, mais genérico,

¹<https://www.google.com>

²<https://cacm.acm.org/>

CAPÍTULO 2. SISTEMAS DE RECOMENDAÇÃO



Figura 2.1: Exemplo de lista de vídeos em alta no YouTube (2017)

como defende [Resnick and Varian \(1997\)](#) em seu artigo.

O sistema do Tapestry foi concebido para a filtragem colaborativa, onde colaborações de outras pessoas auxiliam a outros filtrarem, gravando suas avaliações dos itens. Uma das vantagens da aplicação da filtragem colaborativa é que não depende da análise do conteúdo o que é especialmente útil para a análise de itens complexos como vídeos, amplamente usado em serviços como o YouTube³. Um exemplo das recomendações no YouTube é na página “em alta” que mostra os vídeos em alta tendência baseada no feedback e visualizações. Em geral, as recomendações personalizadas são dispostas como uma lista de itens ranqueados. O termo “item” é o mais comum a ser denotado por SR para usuários, o que pode designar para diversos tipos, como filmes, livros, músicas etc.

Para construir o ranque os SRs tentam predizer qual é o item mais adequado àquele usuário ([Ricci et al., 2011](#)). Para realizar a tarefa o SR coleta das usuárias suas preferências que podem ser informadas de forma explícita, como avaliação de produtos, ou implícita interpretando suas ações como o histórico de navegação. O princípio do SR é da dependência existente entre o usuário e sua atividade em torno dos itens ([Aggarwal, 2016a](#)). Como exemplo, se um usuário comprou um filme de ficção científica, é mais provável que também tenha interesse em outro filme de ficção científica. Dessa forma, o sistema lida com o problema da sobrecarga filtrando itens que sejam menos prováveis de serem gostados, baseando-se nas demonstrações de interesse prévio em outros itens, seja por outros usuários ou não.

O aumento da importância da Web como meio eletrônico, especialmente para o

³<https://www.youtube.com>

e-commerce, também se mostrou como força para o desenvolvimento de sistemas de recomendação. Na Web o usuário pode facilmente informar o seu feedback de produtos sobre o que gostou ou não. Nesse contexto, a aplicação do SR não somente beneficia o usuário, mas também para aqueles que o provem (Isinkaye *et al.*, 2015). Estudos (Baubonienė and Gulevičiūtė, 2015) demonstram que usuários optam por realizar compras online para poupar tempo contudo, com a explosão da variedade de informação disponível, em vez de agir em benefício começa a prejudicar a experiência. É bem aceito que ter escolha é bom, mas ter mais nem sempre é melhor (Ricci *et al.*, 2011).

É importante ressaltar que por fornecer uma informação individualizada, que esteja mais alinhada com o perfil do usuário é o que diferencia os sistemas de recomendação de sistemas de recuperação de informação. Tradicionalmente o motor de buscas deve retornar tudo correspondente a um termo de pesquisa, porém cada vez mais o usuário entra no fator desses sistemas (Burke, 2002). Sistemas como o Google⁴, vão além de retornar termos que batem com a consulta, mas também com a quantidade de outras páginas referentes, histórico de buscas, localização, compatibilidade com dispositivos móveis, além de introduzir informações extra a busca, com os quadros do knowledge graph⁵.

2.2 Conceitos

Sistemas de recomendação são sistemas de processamento de informação que lidam com diversos tipos de dados para construir recomendações que tentam prever a preferência do usuário (Ricci *et al.*, 2011). Os dados tratam-se de basicamente de itens que serão apresentados a usuários na forma de recomendações. Técnicas de recomendação variam com dependência do tipo de conhecimento que pode ser extraído de um dado (Ricci *et al.*, 2011). Dados de avaliações possuem pouca informação, o que resulta em técnicas diferentes em relação daquelas que dependem mais da descrição de um item ou relações com as atividades do usuário. Generalizando, SRs referem-se a três tipos de objetos: itens, usuários e transações que são as relações entre usuários e itens.

- **Itens:** Objetos que são recomendados. Podem ser caracterizados pela complexidade valor ou utilidade. O valor de um item pode ser positivo se é útil para o usuário, ou negativo se não é apropriado ou foi uma decisão errada de seleção por parte do mesmo. O usuário pode ser modelado e representado de diferentes formas, variando

⁴<https://www.google.com>

⁵https://www.google.com/intl/pt_br/insidesearch/features/search/knowledge.html

CAPÍTULO 2. SISTEMAS DE RECOMENDAÇÃO

bastante em relação do domínio operado pelo SR. Toda vez que um usuário interage com um item constrói-se um custo cognitivo, o que pode entrar na relevância na construção do sistema, mesmo se o usuário não chega a adquirir o item interagido. Alguns exemplos de itens são: livros, notícias (baixa complexidade), computadores, viagens, vagas de trabalho (alta complexidade).

- **Usuários:** Usuários de um SR, pode ter uma variedade de objetivos e características, possibilitando explorar uma série de informações variadas para personalizar as recomendações. A informação pode ser estruturada de diversas formas de acordo com o seu tipo, e a seleção de um modelo depende das técnicas a serem utilizadas. Modelos para sistemas de filtragem colaborativa pode usar apenas listas de avaliações de itens por usuários. O modelo de usuário cria o seu perfil, ou seja, armazena suas preferências e necessidades. Usuários também podem ser descritos baseados num padrão de comportamento, como o histórico de navegação na Web sua ou localização.
- **Transações:** Genericamente refere-se a transações gravadas das interações entre usuários e o SR. Transações podem ser vistas como um histórico de registros, um log de dados que armazena importantes informações geradas das interações com o sistema. Um registro pode conter a descrição do que foi consultado para uma recomendação particular de um item.

2.3 Tarefas de um Sistema de Recomendação

Sistemas de recomendação são vistos como mais do que uma ferramenta de prover sugestões de itens que o usuário possa desejar. Em seu artigo, [Ricci et al. \(2011\)](#) introduziu uma série de funções que podem ser aplicadas a SRs.

- **Aumento do número de itens vendidos:** Uma das funções mais importantes para aplicações comerciais. O princípio é ser capaz de vender outros itens comparados àqueles que são vendidos sem qualquer tipo de recomendação. O objetivo é geralmente alcançado devido a itens que são prováveis de serem úteis a necessidade do usuário.
- **Vender itens mais diversos:** Também outra função de alta importância, na qual permite o usuário a selecionar itens que podem ser difíceis de encontrar. Num

2.3. TAREFAS DE UM SISTEMA DE RECOMENDAÇÃO

serviço de recomendações de filmes, como o Netflix⁶, o provedor estará interessado que os usuários encontrem conteúdos diversos, não somente os mais populares.

- **Aumentar a satisfação do usuário:** Quando um usuário encontra recomendações que sejam de seu interesse, impacta na experiência com o sistema. Um SR bem desenvolvido permite uma combinação precisa de recomendações que juntos a uma interface com boa operabilidade, pode aumentar a noção subjetiva da avaliação de um sistema.
- **Aumentar a fidelidade:** Um usuário costuma ser leal a um site que, quando visitado, o reconhece como um consumidor reincidente e o trata como um visitante de valor. É muito comum para um SR levar em consideração as informações obtidas em prévias interações com o usuário. Consequentemente, por quanto mais tempo o usuário interage com o site, mais refinado seu modelo fica, tornando cada vez mais efetivo e personalizado o resultado da recomendação.
- **Melhor entendimento do que o usuário quer:** Outra função importante, na qual pode ser influenciada por outras aplicações, é a descrição das preferências do usuário, seja coletada de forma explícita ou prevista pelo sistema. Um serviço pode decidir reutilizar esses dados do usuário para anunciar um produto em específico, derivado da coleta das informações de transações do SR.

Usuários também podem desejar um SR quando oferecer suporte a suas tarefas ou objetivos. *Herlocker et al.* (2004) é uma clássica referência no assunto, e define onze tarefas comuns que SR podem ajudar a implementar.

- **Encontrar bons itens:** Recomendar a usuários alguns itens em ranque, junto a uma predição de o quanto o usuário possa gostar deles. Também comum no uso em sistemas comerciais.
- **Encontrar todo os bons itens:** Recomendar todos os itens que satisfazem as preferências do usuário. Neste caso é insuficiente apenas encontrar alguns bons itens. Esta função torna-se útil quando existe um número reduzido de itens, ou quando há uma razão crítica para fornecer informação, como em contextos de uso médico ou financeiro.
- **Anotações em contexto:** Dado um contexto, enfatizar alguns itens de uma lista a depender das preferências do usuário.

⁶<https://www.netflix.com>

CAPÍTULO 2. SISTEMAS DE RECOMENDAÇÃO

- **Recomendar uma sequência:** Recomendar uma sequência de itens invés de gerar uma única recomendação.
- **Recomendar um grupo:** Sugerir grupos de itens bem relacionados que possam ser da preferência do usuário.
- **Apenas navegando:** Mesmo que o usuário não possua a intenção de comprar um item, o SR deverá ajudá-lo a navegar pelos catálogo de maneira que encaixe no escopo de interesse do usuário.
- **Encontrar um sistema de recomendação confiável:** Nem todos os usuários podem confiar no sistema, dessa forma é importante oferecer testes de suas funcionalidades.
- **Melhorar o perfil:** Relativo a capacidade de o usuário prover dados ao SR sobre suas preferências. Tarefa fundamental para personalizar o sistema, caso contrário apenas seria possível oferecer recomendações que fossem relativa ao usuário comum.
- **Expressar-se:** Usuários podem não se importar com as recomendações, mas o sistema pode permiti-lo a contribuir com as avaliações e expressão de suas opiniões.
- **Ajudar outros:** Para alguns é importante contribuir com informações de suas opiniões e avaliações, pois compartilhando sua experiência pode ajudar outros formarem uma opinião.
- **Influenciar outros:** Alguns usuários podem ter apenas o objetivo de influenciar outros, ou até usar o SR para difamar a imagem de alguns itens.

2.4 Técnicas de Recomendação

As recomendações utilizadas no sistema são alcançadas através de algumas técnicas que possuem o objetivo de prever informações sobre itens e preferências de usuários. O SR irá produzir recomendações individualizadas como saída, ou será capaz de guiar o indivíduo de forma personalizada a modo de encontrar itens úteis (Burke, 2002). Apresentadas não somente como técnicas de filtragem colaborativa, Resnick and Varian (1997), introduz o termo mais genérico de sistema de recomendação, uma vez que tais sistemas podem explicitamente não utilizar recipientes que talvez sejam desconhecidos uns aos outros.

2.4. TÉCNICAS DE RECOMENDAÇÃO

Para alcançar as principais funções de um SR, é necessário que o sistema seja capaz de identificar que itens possuem alguma utilidade para o usuário (Ricci *et al.*, 2011). O sistema deve prever ou comparar a utilidade de itens, para decidir como recomendá-los. Dessa forma, as recomendações podem variar conforme os dados conhecidos de usuários e itens, podendo ter maior ou menor influência em uma função específica. Como exemplo, durante a etapa da predição pode ser considerado uma informação que não seja necessariamente personalizada, como apenas recomendar itens mais populares. De posse de poucas informações, ou não conclusivas, a premissa é basear-se num item que tem boa aceitação, ou seja, que é útil para muitos, com uma recomendação provável ao usuário genérico.

Ampliando ao já apresentado Tapestry (Goldberg *et al.*, 1992), nem todas as técnicas precisam ser baseadas nas informações de preferências de outros usuários. Na literatura já foram discutidos diversas técnicas, como as apresentadas nos trabalhos de (Ricci *et al.*, 2011) e (Burke, 2002). Dentre essas abordagens estão:

- **Filtragem Colaborativa:** O sistema agrupa avaliações ou recomendações, reconhecendo características comuns entre usuários baseando-se nos itens de suas avaliações.
- **Baseada em conteúdo:** Objetos de interesse são definidos pela associação de suas características. O sistema aprende e recomenda itens similares ao que usuário demonstrou interesse no passado.
- **Demográfico:** Objetivam categorizar o usuário baseado nas informações pessoais dos usuários. Recomendações são baseadas nas classes demográficas dos usuários.
- **Baseada em conhecimento:** Realizam sugestões de itens baseadas em inferências das preferências do usuário.

Abaixo será apresentado em maiores detalhes o funcionamento das técnicas de filtragem colaborativa e baseada em conteúdo.

2.4.1 Filtragem Colaborativa

Recomendação com Collaborative Filtering (CF) é uma das técnicas mais familiares e já implementadas (Ricci *et al.*, 2011). A similaridade das preferências e desejos de dois usuários é calculada baseada na similaridade do histórico de avaliações dos usuários. A premissa do método é de que a opinião de outros usuários pode ser selecionada e agregada

CAPÍTULO 2. SISTEMAS DE RECOMENDAÇÃO

de forma a prover previsões razoáveis ao usuário alvo ([Ekstrand et al., 2011](#)). Como exemplo, intuitivamente assume-se que usuários que concordam sobre a qualidade de um filme que João gosta, então João provavelmente gostará de outros filmes que outros usuários avaliaram, mas não assistiu.

O perfil de um usuário na CF pode ser continuamente aprimorado conforme o usuário interage com sistema, podendo levar o tempo de uso como fator de avaliação. Em alguns casos a avaliação pode ser apenas binária (*like* ou *deslike*), ou então de valor real que determina um grau de utilidade. Nesse caso, nas avaliações do usuário, o sistema deverá modelar uma função $R(u, i)$ representando o grau de utilidade do item i para o usuário u . Basicamente, a tarefa do sistema é estimar um valor de R baseado nos pares de usuário e item. Dessa forma, avaliando os dados dessas previsões de R para o usuário alvo, o sistema recomendará uma quantidade de itens com as maiores utilidades previstas.

Tipicamente, conforme apresentado por [Burke \(2002\)](#), CF divide-se em dois métodos principais: vizinhança e baseados em modelo. No método da vizinhança o foco é no relacionamento entre itens ou usuários, conhecidos como de *item-item* ou *usuário-usuário* ([Ricci et al., 2011](#)), utilizando informações armazenadas com o tempo. O método aborda modelos através da análise da preferência armazenada das classificações de usuário-item, pela avaliação de outros itens similares. Já o método baseado em modelo é criado diretamente do histórico das avaliações para aprender as preferências do usuário, podendo-se usar uma quantidade diversa de técnicas para o aprendizado, como redes neurais. O objetivo é compreender e extrair das interações usuário-item características de destaque para o sistema, podendo criar classes de preferências dos itens.

2.4.2 Filtragem Baseada em Conteúdo

Ao contrário da filtragem colaborativa, sistemas de recomendação baseados em Content Based Filtering ([CBF](#)), seleciona itens baseados entre as relações de seus conteúdos e as preferências do usuário. A CBF é uma continuação natural das pesquisas nos sistemas de filtragem de informação, [Burke \(2002\)](#). O método utiliza-se da intuição de que se o usuário demonstrou interesse em certos itens com determinados atributos, é provável de também ter interesse em outros itens de mesmo atributo ou semelhante. Como exemplo, se João gostou dos filmes com o ator *Tom Cruise*, é provável que vá gostar de outros filmes com o mesmo ator. Os sistemas de CBF foram desenhados para explorar cenários com itens que podem ser descritos com um conjunto de propriedades ou atributos ([Aggarwal, 2016b](#)).

Nessa abordagem, o sistema deverá aprender do perfil do usuário seus interesses base-

2.4. TÉCNICAS DE RECOMENDAÇÃO

ados na combinação das características presentes nos objetos que ele avaliou ou marcou. O tipo do perfil utilizado no sistema dependerá do método aplicado. A informação das preferências do usuário pode manifestar-se de forma explícita, onde existem avaliações ou indicações dos itens favoritos, ou de forma implícita como itens que o usuário comprou. Nos métodos aplicados na CBF, as descrições dos itens avaliados são usadas como dados de treinamento para criar uma classificação específica para o usuário ([Aggarwal, 2016b](#)). Os perfis da filtragem baseada no conteúdo são modelos de longo prazo, onde mais dados são atualizados conforme mais evidências do usuário são observadas, [Burke \(2002\)](#).

Apesar da descrição do conteúdo, ou seja, atributos particulares dos itens, sejam o centro da análise da utilidade de novos itens para recomendação, a avaliação de outros usuários tem significativo impacto no sistema ([Aggarwal, 2016b](#)). Essa característica apresenta tanto vantagens como desvantagens. Por um lado, num contexto da *partida a frio*, onde há pouca informação disponível sobre as avaliações dos usuários, há margem de utilização enquanto houver outras suficientes informações das preferências do usuário. Mesmo quando um item é novo ou desconhecido, o sistema ainda pode aproveitar suas características para recomendar novos itens, algo que não é possível apenas baseando-se nas avaliações de outros usuários.

Assim, sistemas de CBF são tipicamente utilizados quando há suficiente informação das preferências do usuário disponíveis. Particularmente, são de mais fácil utilização quando usados em domínios com dados não estruturados e ricos em textos, como páginas da Web.

2.4.3 Comparação das Técnicas de Recomendação

Todas as abordagens dos SR possuem vantagens e desvantagens, dependendo de questões como novos itens, usuários, e quantidade de informação disponível sobre os dois. Em relação a novos usuários, como recomendações partem da comparação de informações do usuário alvo e outros usuários, quanto menos avaliações o sistema possuir, mais difícil será a classificação. Já para novos itens, o problema surge em domínios em constante atualização e novas informações e onde cada usuário pouco avalia. Também pode ser visto como o problema do *early rater*, uma vez que a pessoa que avalia primeiro, pouco se beneficia.

[Burke \(2002\)](#) apresentou alguns pontos comuns das diferenças desses sistemas:

- **Sistemas baseados em filtragem colaborativa:** Dependem da sobreposição de avaliações através dos usuários e possuem dificuldades quando há escassez dessas

avaliações dos itens. O problema resalta que as técnicas colaborativas melhor servem quando a densidade de interesses de usuários é alta através de um universo de itens que não mudam rapidamente.

- **Sistemas baseados em conteúdo:** Possuem o problema da partida a frio, onde o sistema não acumulou dados suficientes para construir uma recomendação confiável. Também são limitados pela quantidade de informações disponíveis e associadas aos itens. Isto acaba colocando a técnica muito dependente da descrição dos dados. Uma grande desvantagem em relação a abordagem colaborativa é que a abrangência de gêneros, onde deixa o usuário sujeito ao mesmo tipo de conteúdo. A depender da CF, pela a avaliação de outros usuários é possível recomendar itens “fora da caixa”.

2.5 Aplicações de Sistemas de Recomendação

O sistema Tapestry ([Goldberg et al., 1992](#)) foi um marco inicial no desenvolvimento de aplicações, introduzindo a filtragem colaborativa. Hoje, SR são quase que obrigatórios para muitas lojas online e serviços de entretenimento, tornou-se algo comum e já disseminado entre usuários. A seguir será apresentado algumas aplicações em destaque que usam sistemas de recomendação.

2.5.1 Netflix

Com a evolução da Internet, as mídias físicas para consumo de entretenimento começaram a decair, especialmente para filmes. O avanço na conexão da banda larga trouxe o modelo do *streaming*⁷ que possibilita o usuário a assistir o conteúdo a qualquer momento, lugar, sem ter que necessariamente sair de sua residência para ir à uma locadora, por exemplo. Embora o Netflix⁸, tenha iniciado no ramo de aluguel de Digital Video Disc (DVD)s ([Keating, 2012](#)), a companhia rapidamente abandonou este modelo e partiu para a transmissão de filmes e em seguida para produção de seus próprios filmes e séries. Dessa forma, o serviço de filmes e séries cresceu, ocupou espaço das televisões, cinemas e alcançou diversos países.

Com a crescente quantidade de títulos disponíveis na plataforma e também de usuários, logo o serviço desenvolveu seu próprio sistema de recomendações de vídeos, baseado

⁷Transmissão contínua de mídia pela Internet, (<https://directradios.com/streaming>)

⁸<https://www.netflix.com>

2.5. APLICAÇÕES DE SISTEMAS DE RECOMENDAÇÃO



Figura 2.2: Recomendação de Filmes no serviço Netflix. Figura elaborada pelo autor (2017).

nas avaliações de usuários. Em outubro de 2006 a companhia publicou um concurso pelo melhor sistema de filtragem colaborativa que poderia superar a precisão de seu SR, o Cinematch ([Bennett et al., 2007](#)). Neste ponto o serviço já tinha lançado um banco de dados contendo 100 milhões de avaliações de usuários e 18 mil títulos. O Cinematch analisava as avaliações acumuladas dos usuários semanalmente usando uma variante da correlação de Pearson, com todos os outros filmes para determinar uma lista de filmes similares. Sendo assim, conforme o usuário provia avaliações, o sistema computava uma regressão baseada nessa correlação para determinar uma predição única personalizada. Caso não houvesse nenhuma predição personalizada a média de todas as avaliações é usada. As predições eram apresentadas como conjunto de 5 estrelas.

O desempenho do Cinematch é medido principalmente pelo cálculo da raiz do erro quadrático médio, Root Mean Square Error (RMSE) ([Herlocker et al., 2004](#)), das previsões do sistema contra as avaliações que os usuários informam. Com os sistemas propostos no concurso, a companhia propôs um prêmio para aqueles que conseguissem melhorar a precisão em 10%. Nesse ano de 2017, a companhia migrou seu sistema de avaliação das tradicionais 5 estrelas para uma avaliação binária, o *Like* e *Dislike* ([Var, 2017](#)). Segundo a companhia, os usuários confundiam a avaliação de 5 estrelas, pois na verdade eram sempre as previsões avaliadas para o filme, assim agora as previsões aparecem no formato de porcentagem de relevância e a avaliação do usuário é indicada pelos símbolos do gostei ou não gostei. As previsões também passaram a serem baseadas apenas no histórico e comportamento do usuário e não mais na média em relação às outras pessoas.

CAPÍTULO 2. SISTEMAS DE RECOMENDAÇÃO

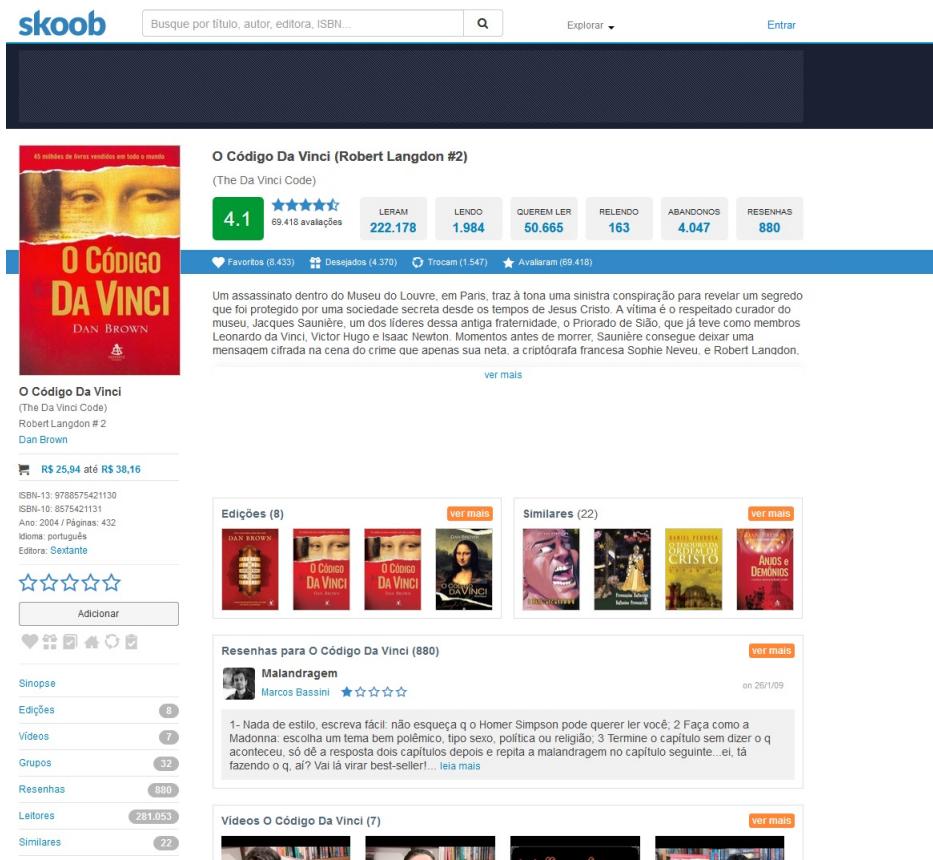


Figura 2.3: Página de avaliação do livro no Skoob. Figura elaborada pelo autor (2017).

2.5.2 Skoob

Em janeiro de 2009, o analista de sistemas Lindeberg Moreira realizou sua ideia de criar uma plataforma em que pessoas socializassem o ato da leitura ([Sko, 2009](#)), o Skoob⁹. O sistema criado trata-se de uma rede social para leitores no Brasil ([Sko, 2017](#)). Na plataforma, o usuário montará uma estante virtual realizando buscas pelos livros e em seguida indicar o que já fez com o livro, se já leu, se lerá ou está relendo. Após a seleção dos livros os usuários poderão avaliar seus livros, podendo até escrever resenhas completas ou de capítulos dos livros, compartilhando com outras pessoas na plataforma.

A rede social, conta com algumas mecânicas para ajudar usuários a encontrar livros, com um sistema busca de livros, recomendação com filtragem colaborativa baseada nas avaliações de usuários, dos marcados como mais lidos, lendo, quero ler entre outros. A plataforma também conta com um sistema que indica livros similares. Todos esses processos não somente levam a questão da socialização da leitura e escrita entre indivíduos

⁹<https://www.skoob.com.br>

que compartilham interesses, surgidas a partir da aplicação, mas passam a influenciar a forma como usuários passam a tratar a leitura fora do ambiente da comunidade virtual, é o que aponta VIANA NETO (2010).

2.6 Sumário

Neste capítulo, foi apresentado um panorama geral sobre os sistemas de recomendação. Inicialmente abordando o histórico envolvido e motivações na criação dos conceitos envolvidos do tema. Em sequência foi aprofundado e explicado os conceitos utilizados nesses sistemas. Então, foi apresentado as tarefas e técnicas utilizadas. Também foi aprofundado algumas diferenças e dificuldades entre as principais técnicas de recomendação. Por fim, foi mostrado exemplos de aplicações que utilizam esses sistemas de recomendação. No capítulo 3 será discutido sobre os conceitos envolvidos na Web Semântica, bem como os princípio dos dados ligados e o serviço da DBpedia¹⁰.

¹⁰<http://wiki.dbpedia.org>

3

Web Semântica

I found myself answering the same questions asked frequently of me by different people. It would be so much easier if everyone could just read my database

—TIM BERNERS-LEE

A introdução e expansão da *World Wide Web* possibilitou acessar e publicar uma grande variedade de conteúdo, seja para o consumo de entretenimento, exposição de opiniões, compras online etc. O crescimento da rede tornou-se tão grande que é latente a dificuldade dos usuários encontrar informações. Para eles, foram criados e desenvolvidos os indexadores de páginas, como o Google¹, Yahoo², Bing³. Tais sistemas facilitam encontrar informações em serviços populares na Internet. Entretanto, e se quiséssemos encontrar algum médico de confiança para marcar uma consulta, levando em consideração uma agenda de compromissos? Ou então se estamos realizando um trabalho escolar e queremos encontrar os reis do século XV? Essas pesquisas certamente são mais complicadas, e resultados de buscas tradicionais levam a informações fragmentadas, com uma série de outras buscas separadas para alinhar todo o conhecimento e semântica envolvidos nessas tarefas. É nesse ponto que entra o conceito da Web Semântica, como uma extensão da já existente.

O conteúdo da Web tradicional é fundamentalmente desenvolvido para humanos lerem, não para máquinas manipularem de forma produtiva e significante (Berners-Lee *et al.*, 2001). Originalmente desenvolvida para compartilhar e apresentar conteúdo de forma que fosse possível interagir e navegar entre hipertextos e hiper mídia, a World

¹<https://www.google.com>

²<https://www.yahoo.com>

³<https://www.bing.com>

Wide Web ([WWW](#)) torna fácil a apresentação de layouts. É possível estruturar um documento com um cabeçalho, um link para outra página, entretanto, dificilmente as máquinas poderão processar semanticamente que informações estão disponíveis e que podem ser organizadas naquela página ou site. Como exemplo, uma página de João com link para seu currículo informando que possui especialização em cardiologia. Todas essas informações podem até serem compreendidas por humanos ao associar a semântica das entidades presentes numa página e analisando links relacionados, mas para a máquina não há uma estrutura comum e eficiente que leve a essas mesmas conclusões.

O objetivo da Web Semântica é de estender a WWW, aproveitando a enorme variedade de dados já existente, mas agregando uma nova camada de metadados que possibilitem o processamento pela máquina e agentes de forma a compreender a semântica das informações apresentadas. Assim, a Web Semântica trata-se de prover formatos para integração de dados de diferentes fontes ([W3C, 2001](#)), onde a Web tradicional mantém-se como o meio de publicação e interconexão de documentos, e na contraparte semântica, armazena-se dados que se relacionam com objetos e coisas do mundo real. Um agente pode se deparar com uma página de clínica na Web e não apenas compreenderá que possui palavras como “tratamento, terapia, remédios, médicos”, como tipicamente é encontrado na Web tradicional, mas também saber que o “Dr João” trabalha nessa clínica nas segundas e quartas com horários no formato *dd/mm/YYYY*.

3.1 Arquitetura e formato de dados

O funcionamento da Web Semântica depende da capacidade de máquinas acessar coleções estruturadas de informações, dados e regras de inferência para executar raciocínio automatizado ([Berners-Lee et al., 2001](#)). O desafio é de como representar conhecimento. Inicialmente o desenvolvimento desses sistemas utilizaram uma abordagem centralizadora, requerendo que as partes envolvidas compartilhem exatamente as mesmas definições de conceitos comuns ou hierárquicos. Entretanto, com a quantidade de conteúdo existente hoje em diferentes línguas, controle centralizado é desafiador. Contrastando essa visão inicial, na Web Semântica cria-se linguagens para regras as quais são tão expressivas quanto o necessário para que a Web seja ampla como desejado ([Berners-Lee et al., 2001](#)). Com um sistema que não seja centralizado é possível que não se responda todas as perguntas, ou seja, encontrado todas as informações, mas permite que regras sejam usadas para criar inferências e escolher o curso de ações para poder ou tentar responder tais perguntas.

Com esses fundamentos os pesquisadores da Web Semântica, em especial o *World*

3.1. ARQUITETURA E FORMATO DE DADOS

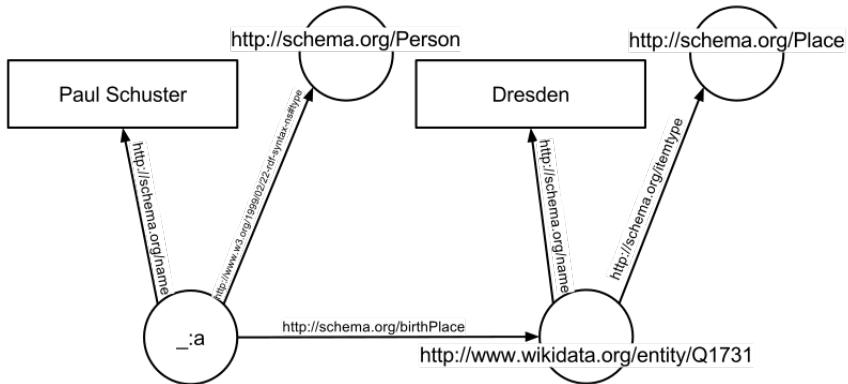


Figura 3.1: Exemplo do grafo RDF ([RDF, 2017](#))

Wide Web Consortium, desenvolveram uma série de padrões e formatos de dados para o uso na Web. O intuito é possibilitar máquinas compreenderem documentos com dados semânticos e não discursos e textos criados pelo homem. Uma tecnologia muito importante para o desenvolvimento da representação do conhecimento e protocolo de comunicação entre máquinas, foi a eXtensible Markup Language ([XML](#)). Com a XML é possível que qualquer um seja capaz de criar suas próprias *tags* e estruturas de um documento com definição de cada termo presente de forma arbitrária. Desse ponto de vista a XML é fundamental como um padrão de comunicação entre máquinas. Anos seguintes, a W3C introduziu outras três importantes tecnologias presentes no cenário atual da Web Semântica: Resource Description Framework ([RDF](#)), SPARQL Protocol and RDF Query Language ([SPARQL](#)), Ontology Web Language ([OWL](#)).

3.1.1 RDF

Resource Description Framework é um modelo de dado para a Web que facilita a junção de dados mesmo que seu *schema* difira, além de permitir a sua evolução sem requerer que seus consumidores tenham que se adaptar ([W3C, 2014](#)). No RDF a estrutura da *web* de links é estendida para usar os Universal Resource Identifier ([URI](#)) para nomear a relação entre qualquer coisa, com ambas as pontas, formando o que é conhecido como a tripla.

O uso da URI é especialmente notável para a Web, uma vez que não é possível apenas se basear em valores literais, mesmo para representar um atributo de algo, já que é desejado ter a definição e estrutura podendo considerar um domínio em específico. Como exemplo, com uma URI é possível identificar de forma única o predicado “título” que se refere ao título da função em uma empresa, e não um título de filme. Então, a tripla forma um grupo de três entidades que expressam uma declaração sobre o dado semântico,

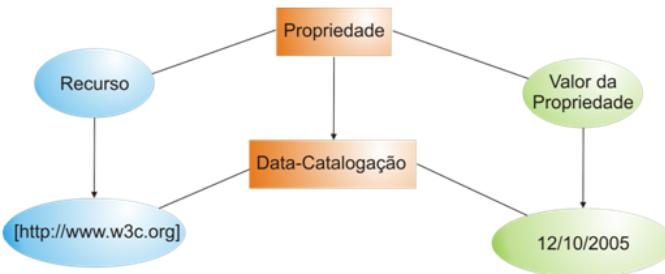


Figura 3.2: Exemplo do grafo da tripla sujeito predicado objeto (Web, 2009)

na forma de “sujeito, predicado, objeto”. Com essa estrutura de links é formado um grafo direcionado, com *labels*, aonde suas arestas representam o link nomeado entre dois recursos representados pelos seus nós (W3C, 2014).

3.1.2 SPARQL

O SPARQL é uma linguagem de consulta para o grafo do RDF (W3C, 2013). Dessa forma, pode-se criar *queries* através de diversos conjuntos de dados de triplas, podendo ser aplicado uma série de filtros para limitar e ordenar os resultados retornados. Diferentemente das linguagens de consulta de banco de dados relacionais, o objeto da coluna não é homogêneo, ou seja, o tipo dado da célula da tabela de resultados é implicado ou definido pelo predicado informado através da URI. O sujeito do RDF pode ser classificado com um análogo a uma entidade nos bancos de Structured Query Language (SQL), diferindo onde os campos (ou atributos) são representados como predicados e/ou objetos separados.

O exemplo do Código Fonte 3.1 demonstra a consulta de dados de uma ontologia *foaf*⁴, conhecida como "friend of a friend".

3.1.3 OWL

Ontology Web Language é uma linguagem para definir e instanciar ontologias na Web (W3C, 2009). Um programa que deseja comparar ou combinar informações entre dois bancos de dados com URIs distintas, deve saber se termos podem ser usados para descrever o significado da mesma coisa (Berners-Lee *et al.*, 2001). O objetivo é que um programa descubra o significado comum seja para o que for encontrado entre os conjuntos de dados. A solução proposta na Web Semântica para esse problema é a utilização de uma coleção de informações denominadas de ontologias. Na filosofia uma ontologia tem por objeto o estudo das propriedades do ser, tratando da natureza da existência. Entretanto,

⁴<http://xmlns.com/foaf/spec/>

3.1. ARQUITETURA E FORMATO DE DADOS

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 SELECT ?name
4     ?email
5 WHERE
6 {
7     ?person a foaf:Person .
8     ?person foaf:name ?name .
9     ?person foaf:mbox ?email .
10 }
11 }
```

Código Fonte 3.1: Exemplo de consulta na linguagem SPARQL

no campo da inteligência artificial e na Web, é definido como os termos básicos e relações que compreendem um vocabulário de um domínio, bem como regras para combiná-los junto com relações para definir extensões desse vocabulário (Patil *et al.*, 1992).

Em essência a ontologia é um documento que define formalmente as relações entre termos. As ontologias podem ser vistas de forma semelhante à hierarquia de classes na programação orientada a objetos. Tipicamente uma ontologia para a Web possui uma taxonomia e um conjunto de regras de inferência. A taxonomia define classes (ou conceitos) de objetos e suas relações, sendo assim, um endereço pode ser definido como um tipo de localidade e o código de uma cidade pode ser definido para ser aplicado apenas a localizações, entre outros exemplos.

A linguagem OWL provê três sublinguagens, OWL Lite, OWL DL, OWL Full como apresentado pela [W3C \(2009\)](#).

- **OWL Lite:** Para a criação hierárquica e simples de limitações de *features*. Como exemplo, é possível oferecer suporte a limitações de cardinalidade que só permitam valores de 0 ou 1. É mais simples de prover suporte.
- **OWL DL (descrição lógica):** Oferece suporte a uma expressividade máxima sem perder a completude computacional (todas as implicações são garantidas para serem computadas), decidibilidade (todos os cálculos finalizaram em um tempo finito). Inclui todas as construções com restrições e separação de tipos (uma classe também não pode ser indivíduo ou propriedade, uma propriedade também não pode ser um indivíduo ou uma classe).
- **OWL Full:** Oferece o máximo de expressividade e é sintaticamente livre do RDF sem garantias computacionais. Nessa linguagem uma classe pode ser tratada

CAPÍTULO 3. WEB SEMÂNTICA

```
1 <rdf:RDF
2   xmlns ="http://www.w3.org/TR/2004/REC-owl-guide
3     ↪ -20040210/wine#"
4   xmlns:vin ="http://www.w3.org/TR/2004/REC-owl-guide
5     ↪ -20040210/wine#"
6   xml:base ="http://www.w3.org/TR/2004/REC-owl-guide
7     ↪ -20040210/wine#"
8   xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide
9     ↪ -20040210/food#"
10  xmlns:owl ="http://www.w3.org/2002/07/owl#"
11  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#
12    ↪ "
13  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
14  xmlns:xsd ="http://www.w3.org/2001/XMLSchema#">
```

Código Fonte 3.2: Exemplo do topo de um documento OWL

simultaneamente como uma coleção de indivíduos ou indivíduo como todo. Então, a OWL Full permite uma ontologia ter seu significado ampliado ao pré-definido (RDF ou OWL) vocabulário.

Todas as sub-linguagens são extensões de sua predecessora, sendo assim cada ontologia válida em OWL Lite é uma ontologia válida em OWL DL que por sua vez é uma ontologia válida em OWL Full ([W3C, 2009](#)). É notável destacar que o inverso das relações não é verdadeiro. Completando, todo documento OWL é um documento em XML construído com o RDF.

Estrutura de um documento

Com a OWL é possível descrever de forma natural classes e relacionamentos entre documentos e aplicações na Web ([Júnio César de Lima, 2005](#)). Os termos descritos devem estar dispostos de tal maneira que não cause ambiguidade, assim é necessário que seja informado quais vocabulários serão empregados. Para o uso de vocabulários a [W3C \(2009\)](#) informa que deve-se definir no topo do documento os *xml namespaces*⁵, conforme mostrado no código fonte 3.2.

Acrescentando, a World Wide Web Consortium ([W3C](#)) recomenda incluir no documento um cabeçalho XML que preceda as definições das ontologias como apresentado no código fonte 3.3

⁵No XML, os namespaces são nomes únicos para elementos e atributos no documento. Para resolver as ambiguidades e facilitar as referências antes dos nomes são utilizados prefixos

3.1. ARQUITETURA E FORMATO DE DADOS

```
1 <!DOCTYPE rdf:RDF [  
2   <!ENTITY vin "http://www.w3.org/TR/2004/REC-owl-guide  
3     ↪ -20040210/wine#">  
4   <!ENTITY food "http://www.w3.org/TR/2004/REC-owl-guide  
5     ↪ -20040210/food#"> ]>
```

Código Fonte 3.3: Exemplo do cabeçalho XML de um documento OWL

```
1 <owl:ObjectProperty rdf:ID="subordinate">  
2   <rdf:type rdf:resource="&owl;TransitiveProperty"/>  
3   <rdfs:domain rdf:resource="#Agent"/>  
4   <rdfs:range rdf:resource="#Agent"/>  
5 </owl:ObjectProperty>  
6  
7 <Agent rdf:ID="Joao">  
8   <subordinate rdf:resource="#Pedro"/>  
9 </Agent>  
10  
11 <Agent rdf:ID="Pedro">  
12   <subordinate rdf:resource="#Maria"/>  
13 </Agent>
```

Código Fonte 3.4: Exemplo de propriedades transitivas no OWL

```
1 <owl:Ontology rdf:about="">  
2   <rdfs:comment>An example OWL ontology</rdfs:comment>  
3   <owl:priorVersion rdf:resource="http://www.w3.org/TR  
4     ↪ /2003/PR-owl-guide-20031215/wine"/>  
5   <owl:imports rdf:resource="http://www.w3.org/TR/2004/REC-  
6     ↪ owl-guide-20040210/food"/>  
7   <rdfs:label>Wine Ontology</rdfs:label>  
8   ...
```

Código Fonte 3.5: Exemplo do cabeçalho de uma ontologia

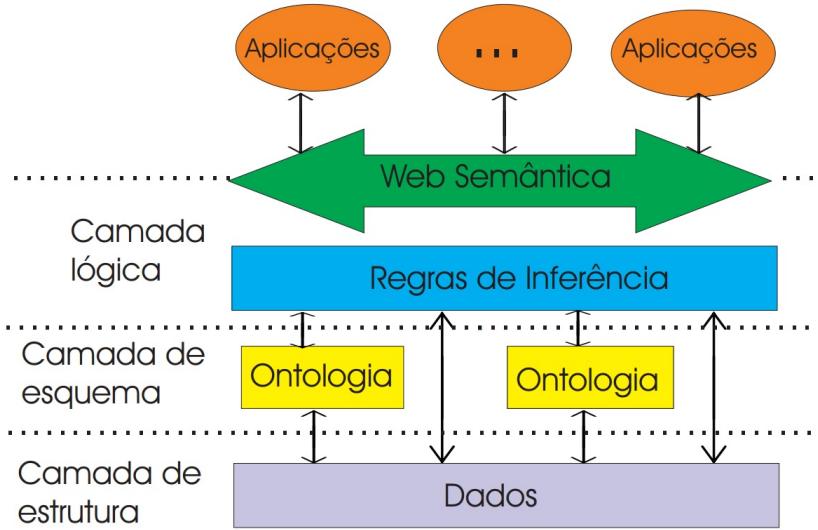


Figura 3.3: Camadas na rede semântica. ([Júnio César de Lima, 2005](#))

Por último será informado o cabeçalho da ontologia junto a suas propriedades. Nesse cabeçalho é importante fornecer informações sobre ela própria. Para descrevê-las utilize-se as propriedades do OWL, uma vez que a ontologia é um recurso, assim demonstrado no código fonte [3.5](#)

Dentro da definição da ontologia poderão ser informados as classes e indivíduos relacionados como as propriedades e suas relações. As propriedades podem ser descritas como transitivas, simétricas, funcionais ou inversamente funcional. Como exemplo, numa propriedade transitiva de subordinado, se é dito que João é subordinado de Pedro e Pedro é subordinado de Maria, portanto João é subordinado de Maria. No código fonte [3.4](#) é demonstrado a declaração desse tipo de propriedade.

3.1.4 Estrutura na rede semântica

A introdução das tecnologias para alcançar os princípios idealizados na Web Semântica são implantados em camadas. De acordo com [Berners-Lee et al. \(2001\)](#) é possível dividir esses serviços em três grandes camadas, como demonstrado na Figura 3.3. Na camada de estrutura os dados são organizados e definidos seus significados, na qual utiliza-se as triplas do RDF. A camada com os esquemas estão as ontologias, utilizando-se o OWL para a representação de conceitos, inferências através das taxonomias e conjunto de regras. Por último na camada lógica é definida para fazer inferência sobre os dados. Dessa

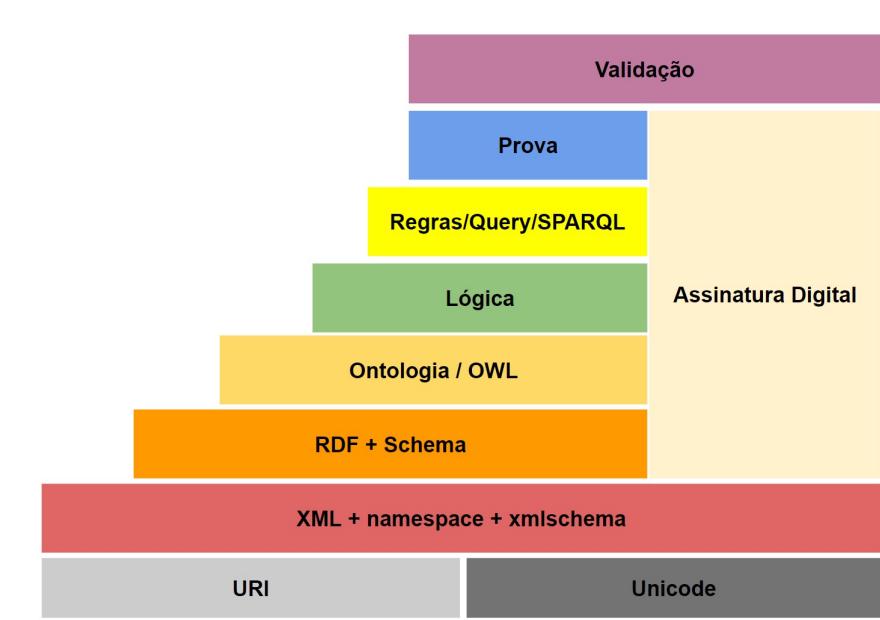


Figura 3.4: Camadas na rede semântica. Figura elaborado pelo autor de acordo com a publicação de [Berners-Lee \(2008\)](#)

forma, o desenvolvimento dessas tecnologias (ainda em andamento) e padronização dos formatos foi formulado pela W3C como uma pilha das camadas ([Berners-Lee, 2008](#)) da Web Semântica confiável, conforme mostrado na figura 3.4.

3.2 Dados ligados

A evolução da WWW tornou cada vez mais acessível a publicação e acesso a documentos pela navegação no espaço global, através links dos hipertextos ([Bizer et al., 2009](#)). Com os navegadores da Web pode-se passear pelos links nesse espaço e em especial com o uso dos buscadores, que indexam páginas para facilitar a recuperação. Tais mecanismos já estão amplamente difundidos na publicação de documentos, mas quando comparados aos dados⁶ em si, esses princípios ainda foram timidamente aplicados. Assim, com o crescimento da Web Semântica trouxe-se a ênfase em criar uma Web para os dados, capaz descrever entidades individuais presentes nos documentos, conectando-se por links categorizados para relacionar tais entidades. O objetivo não é somente colocar dados na Web, mas utilizar links que ambas máquinas (principalmente) e humanos possam navegar.

⁶Note que embora os termos "dados" e "documentos" possam ser análogos, no contexto da Web, documentos tratam-se das páginas dos sites e dados, de fato a informação em si. Assim, na Web os documentos apenas objetivam o aspecto da apresentação não contento a semântica dos dados presentes.

CAPÍTULO 3. WEB SEMÂNTICA

Suportando essa evolução da Web ([Berners-Lee, 2006](#)) introduziu um conjunto de melhores práticas para a publicação e conexão de dados estruturados na Web, denominado de *Linked Data* (dados ligados). A adoção dessas práticas permite a extensão da Web como um espaço de dados global conectado de diversos domínios, desde pessoas, livros, publicações até dados governamentais dos mais variados assuntos. Com essa Web de dados surge a oportunidade para novos tipos de aplicações ([Bizer et al., 2009](#)), como navegadores customizados para um determinado domínio podendo saltar entre diferentes fontes de dados.

Resumidamente, a [W3C \(2006\)](#) define que para a Web de dados ser uma realidade é necessário que os dados estejam disponíveis em padrões de formatos que sejam buscáveis e manipuláveis pelas ferramentas e tecnologias da Web Semântica. Complementando, é preciso também ter acesso ao relacionamento de dados. O conjunto de *datasets* inter-relacionados na Web, para criar links tipificados entre dados de diferentes fontes é o que se denomina de dados ligados.

Ao contrário dos documentos HyperText Markup Language ([HTML](#)) na Web dos hipertextos, os dados ligados se baseiam-se nos documentos contendo dados em RDF. Assim são construídos links que são tipificados para realizar declarações sobre coisas arbitrárias no mundo. [Berners-Lee \(2006\)](#) enumerou um conjunto das regras para a publicação e conexão dos dados, conhecidos como os princípios dos dados ligados:

1. Usar URIs para nomear coisas
2. Usar HyperText Transfer Protocol ([HTTP](#)) URIs para que pessoas possam procurar seus nomes.
3. Quando alguém procura uma URI, forneça informação útil, utilizando os padrões como RDF e SPARQL.
4. Inclua links para outras URIs, para que assim eles possam descobrir mais coisas.

Um exemplo notável do uso das dados ligados, é o projeto da DBpedia⁷ que essencialmente torna o conteúdo da Wikipedia⁸ disponível em RDF.

3.2.1 Linked Open Data

Posteriormente em 2010, para incentivar o uso dados ligados no meio governamental, [Berners-Lee \(2006\)](#) desenvolveu um "sistema de avaliação" dos dados ligados. O objetivo

⁷<http://wiki.dbpedia.org>

⁸<https://www.wikipedia.org>

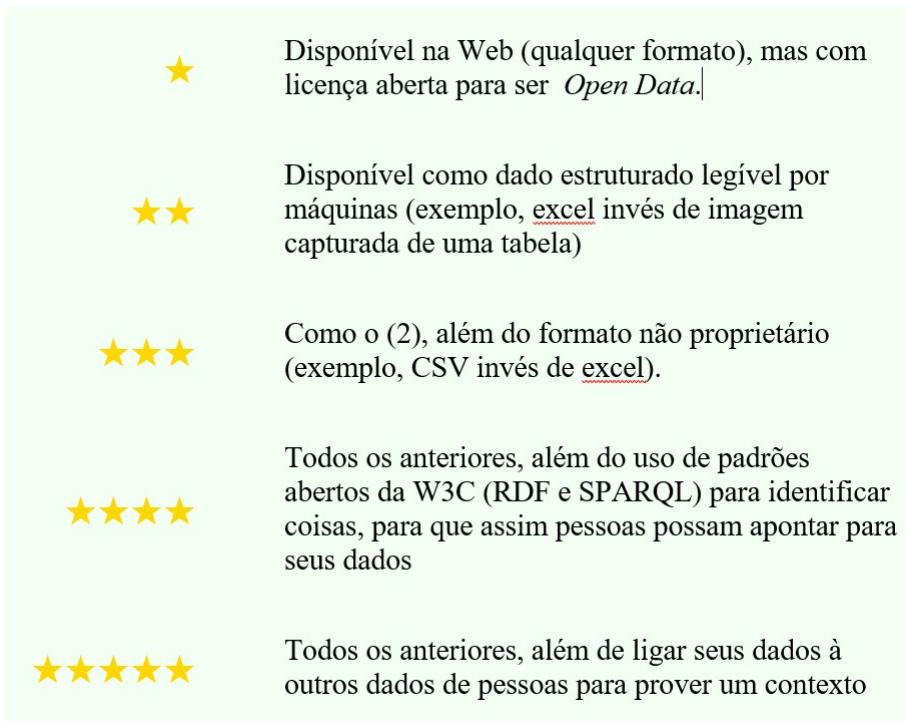


Figura 3.5: Sistema de avaliação do LOD (Berners-Lee, 2008)

era expandir o termo introduzindo os dados abertos, onde fossem publicados sob uma licença que não impede o livre reuso. No sistema de avaliação consta um esquema de pontuação em estrelas de 1 a 5, onde cada estrela a mais também acumula as definições das estrelas anteriores, conforme consta na figura 3.5.

O Linked Open Data (**LOD**) tornou-se o projeto de maior adoção dos princípios dos dados ligados (Bizer *et al.*, 2009), sendo um esforço colaborativo iniciado em 2007 para suportar as definições e tecnologias da Web Semântica introduzidas pela W3C. O motivo para o início da colaboração era de mapear os dados da Web identificando os conjuntos que já estavam disponíveis sob licença aberta. O projeto inclui dados de várias fontes, como a Wikipedia⁹, Geonames¹⁰, Wordnet¹¹ entre diversos outros de múltiplos domínios, alcançando um impressionante diagrama como mostrado na Figura 3.6.

⁹<https://www.wikipedia.org>

¹⁰<http://www.geonames.org>

¹¹<https://wordnet.princeton.edu>

CAPÍTULO 3. WEB SEMÂNTICA

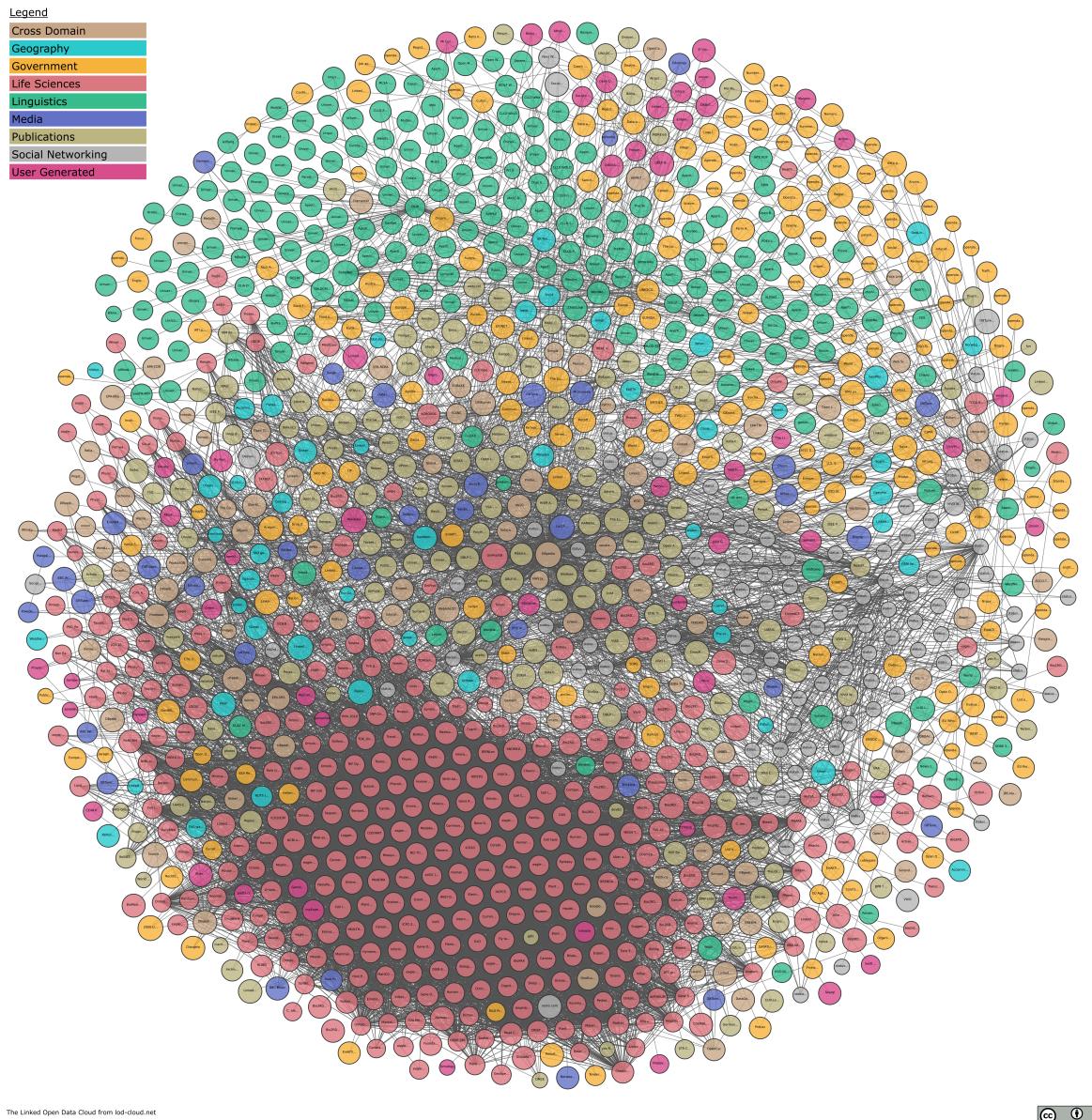


Figura 3.6: Diagrama da nuvem dos dados ligados ([Andrejs Abele, 2020](#))

3.3 Similaridade Semântica

A similaridade semântica entre dois termos, recursos, itens ou documentos é uma métrica para medir a distância de seus significados ou semântica, dado suas ontologias (Slimani, 2013). O objetivo é estabelecer características em comum entre dois conceitos. A distância entre dois conceitos para humanos pode não ter uma definição formal, já que se pode criar um juízo de valor diferente no relacionamento entre eles. Como exemplo, para uma pessoa a maçã e a banana podem estar mais relacionadas do que a maçã e a pera para outra. A similaridade e relação semântica podem por vezes serem determinadas como a mesma coisa, ambas como métricas de distâncias entre termos, contudo a similaridade semântica é mais específica (Slimani, 2013). A relação semântica é calculada usando um modelo de espaço vetorial e uma métrica de similaridade, como a similaridade do cosseno 3.2 que dado dois vetores A e B como uma representação de dois documentos e A_i e B_i seus componentes, seja calculado o produto vetorial euclidiano. (Singhal, 2001).

$$A \cdot B = \|A\|_2 \|B\|_2 \cos(\theta) \quad (3.1)$$

$$\text{similaridade} = \cos(\theta) = \frac{A \cdot B}{\|A\|_2 \cdot \|B\|_2} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.2)$$

Entretanto, para a similaridade semântica é levado em consideração relações léxicas de sinonímia e hiperonímia onde o significado é abrangido pelo outro termo mais geral (como carro e veículo) (Gracia and Mena, 2008). Na prática, a similaridade semântica pode ser medida pelo menor caminho entre dois termos utilizando suas ontologias associadas. Para calcular a similaridade podem ser usadas diversos tipos de ontologias. Slimani (2013) descreve dois principais tipos de ontologias usadas para medir similaridade.

- **Propósito genérico:** *Wordnet*¹² é um banco de dados que modela o conhecimento léxico da língua inglesa. Nomes, verbos, adjetivos e advérbios são agrupados em conjuntos sinônimos, onde cada um expressa um conceito distinto. Essa ontologia pode ser utilizada para criar um *score* de similaridade. Pode ser considerada um ontologia para termos de linguagem natural.
- **De domínio específico:** *ULMS*¹³ é um sistema de linguagem médica com uma rede semântica de ontologias de multiuso, multilíngue para biomedicina, conceitos e

¹²<https://wordnet.princeton.edu>

¹³<https://www.nlm.nih.gov/research/umls>

assuntos relacionados à saúde. O banco de dados do sistema possui uma coleção de vocabulários de conceitos e termos e seus relacionamentos que são denominados de *Metathesaurus*. Cada Metathesaurus é classificado como pelo menos uma categoria semântica.

3.3.1 Medidas de Similaridade Semântica

Na literatura já foram apresentadas algumas medidas de similaridade semântica, mas comumente existem três fatores principais (Slimani, 2013) que podem ser associados na topologia (i.e. nós do grafo direcionado) das ontologias: *path length*, *depth*, *density*. Todos esses fatores afetam a medida da distância semântica, assim como as características entre dois termos, que podem aumentar ou diminuir as medidas de acordo com suas semelhanças. Quanto a densidade entre dois termos trata-se do número de filhos dos quais pertencem ao menor caminho (*path*) da raiz ao mais específico conceito entre esses termos. Os fatores que influenciam nas medidas levam a definição de uma classificação que podem ser divididas em quatro principais (Slimani, 2013): baseadas em estrutura, conteúdo, recursos ou características e as híbridas que combinam as características estruturais (*path length*, *depth*, *density*) e alguma outra abordagem.

Baseadas em estrutura

As medidas baseadas em estrutura (*Structured-based ou Path-based*, utilizam funções que computam a similaridade baseada na hierarquia e estrutura da ontologia, ou seja, onde um conceito é definido como “é parte de”, “é um” etc. A função calcula o tamanho do caminho que liga os termos e seus posicionamentos no grafo direcionado da ontologia. Quanto mais dois conceitos são similares, mais *links* existem entre eles. Dentre as medidas baseadas em estrutura se destacam:

- **Shortest Path** (Rada et al., 1989): A medida do menor caminho é um tipo de medida de distância que é primariamente voltada para lidar com hierarquias em redes semânticas. A função da similaridade entre conceitos C_1 e C_2 é definida como:

$$Sim(C_1, C_2) = 2 * Max(C_1, C_2) - SP \quad (3.3)$$

A função *Max* é o maior tamanho do caminho entre C_1 e C_2 , quanto a *SP* é menor caminho relacionando os dois conceitos.

- **Weighted Links:** Similar a medida do menor caminho, contudo é introduzido um conceito de pesos para os links entre os conceitos a serem comparados.
- **Wu and Palmer (1994):** Para essa medida sejam dois conceitos C_1 e C_2 , é levado em consideração a noção intuitiva de que quanto maior a profundidade, mais similares os conceitos são. Na função tem-se que N_1 e N_2 são a quantidade de links da forma "é um" de C_1 e C_2 , onde o conceito mais específico é o mais próximo ancestral C entre eles.

$$Sim_{W\&P}(C_1, C_2) = \frac{2H}{N_1 + N_2 + 2H} \quad (3.4)$$

Baseadas em conteúdo

As medidas baseadas no conteúdo, são aquelas que utilizam a informação do conteúdo para medir similaridade. O conteúdo de um conceito é definido pela frequência de termos dado uma coleção de documentos. Grande parte das medidas deste tipo utilizam a informação compartilhada de dois conceitos pais C_1 e C_2 , dos qual $S(C_1; C_2)$ é o conjunto de conceitos que os engloba, conforme a Equação 3.5. O menor $p(C)$ é utilizado quando há mais de um pai em comum que C é o Most Informative Subsume (MIS), ou seja, o conceito mais informacional que os engloba.

$$P_{mis}(C_1, C_2) = \min_{C \in S(C_1; C_2)} \{p(C)\} \quad (3.5)$$

Algumas das medidas deste tipo são:

- **Resnik (1999):** O princípio desta medida define que dois conceitos são mais similares se eles possuem mais informações compartilhadas. A informação compartilhada entre C_1 e C_2 é o conteúdo de conceitos que os engloba no grafo. A definição de Resnik define a medida como a seguinte equação:

$$Sim_{Resnik}(C_1, C_2) = -\ln(p_{mis}(C_1, C_2)) \quad (3.6)$$

- **Lin (1993):** A proposta é incorporar o vetor semântico e a ordem das palavras para calcular a similaridade. A medida combina o menor caminho SP entre dois conceitos e a profundidade N da taxonomia em relação ao conceito C mais em comum. A definição da equação segue conforme abaixo:

$$Sim_{Li}(C_1, C_2) = e^{-\alpha * SP} * \frac{e^{\beta * N} - e^{-\beta * N}}{e^{\beta * N} + e^{-\beta * N}} \quad (3.7)$$

Baseadas em características ou recursos

Baseia-se em características ou recursos (*Featured-based*), que partem do princípio de valorizar informações importantes em relação ao conhecimento sobre um termo. A medida assume que os conceitos são descritos por termos indicando suas propriedades ou *features*. A similaridade entre dois conceitos é definida por uma função (3.8) que relaciona suas propriedades ou relacionamentos a outros termos similares na hierarquia da ontologia. [Tversky \(1977\)](#) apresenta uma medida *Feature-based* de termos para calcular a similaridade entre diferentes conceitos, contudo o posicionamento desses termos na taxonomia e a informação do conteúdo não são levadas em consideração. A proposta é de que com termos descritos por um conjunto de palavras como propriedades do conceito, então as que são em comum tendem a aumentar a similaridade, enquanto as que não são em comum tendem a diminuí-la. Dessa forma, é definida uma equação onde C_1 e C_2 representam o conjunto de descrições dos termos e $\alpha \in [0, 1]$ é a relação de relevância das características que não são em comum. O valor de α aumenta o quanto mais em comum dois conceitos são, e decresce com suas diferenças, e não é necessariamente uma relação de simetria, mas mais baseada na similaridade ([Slimani, 2013](#)).

$$Sim_{Tversky}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha|C_1 - C_2| + (\alpha - 1)|C_2 - C_1|} \quad (3.8)$$

3.4 Projetos na Web Semântica

3.4.1 DBpedia

A DBpedia (DB para *database*) é um esforço colaborativo para a extração de dados do Wikipedia para publicação de dados essencialmente em RDF ([Auer et al., 2007](#)). Um dos objetivos é possibilitar que outros explorem a criar uma experiência da enciclopédia mais abrangente, utilizando serviços e aplicações na Web Semântica. O projeto é um dos mais famosos que aplica os conceitos de dados ligados, onde sua importância não somente é dada pela publicação dos dados da Wikipedia, mas também da incorporação de links de outros *datasets*. De fato, o DBpedia, por muitas vezes é considerado um núcleo dentro da iniciativa do LOD.

3.4. PROJETOS NA WEB SEMÂNTICA

Amostra de Dados do DBpedia						
	Línguas					
Instâncias	Inglês	Espanhol	Português	Francês	Alemão	Russo
Pessoas	1.445.104	99.147	60.056	134.749	179.421	86.269
Atores	6.501	13.831	7.546	14.019	0	0
Artistas	96.282	34.898	14.603	32.562	0	30.266
Políticos	40.343	7.460	4.110	11.461	0	0
Lugares	735.062	156.377	123.114	148.586	168.082	91.099
Instuições de ensino	49.172	1.709	514	2.943	2.600	1.418
Filmes	87.282	12.140	11.643	15.669	18.707	14.912
Livros	31.029	2.217	1.343	3.549	0	18.491
Software	31.401	6.284	4.245	8.980	5.286	0

Tabela 3.1: Recorte da tabela de dados de triplas de entidades mapeadas no DBpedia. ([DBpedia, 2014](#))

O projeto tem o foco em converter o conteúdo presente do Wikipedia em conhecimento estruturado utilizando as tecnologias da Web Semântica, para que outros agentes possam explorar realizando consultas e ligando a outros conjuntos de dados ([Auer et al., 2007](#)). Assim, o projeto cobre uma das limitações da Wikipedia que é a dependência de apenas ter a busca em texto livre para encontrar informação. Desse papel, o projeto promove três importantes contribuições:

- Desenvolvimento de um *framework* para extração de informação, o qual converte o conteúdo da Wikipedia em RDF.
- Prover o conteúdo da Wikipedia como um largo, multi-domínio *dataset* de RDF. Já são cerca de 900 milhões de triplas mapeadas (2021). A Tabela 3.1 mostra um recorte das entidades mapeadas do DBpedia.
- Interligar o DBpedia com outros conjuntos de dados abertos, o que expande a contagem das triplas RDF para mais de bilhão.
- Desenvolvimento de uma série de interfaces e módulos de acesso para que tal *dataset* possa ser acessado por serviços da Web ligado a outros sites.

Como a iniciativa compõe o movimento do LOD, os dados do DBpedia podem ser importados por aplicações *third party* utilizando a licença aberta. O projeto faz uso da plataforma do *Virtuoso Universal Server*¹⁴ para prover os dados de RDFs através de uma interface e um *endpoint* em SPARQL. Na Figura 3.7 é apresentado um panorama da arquitetura do projeto.

¹⁴<https://virtuoso.openlinksw.com>

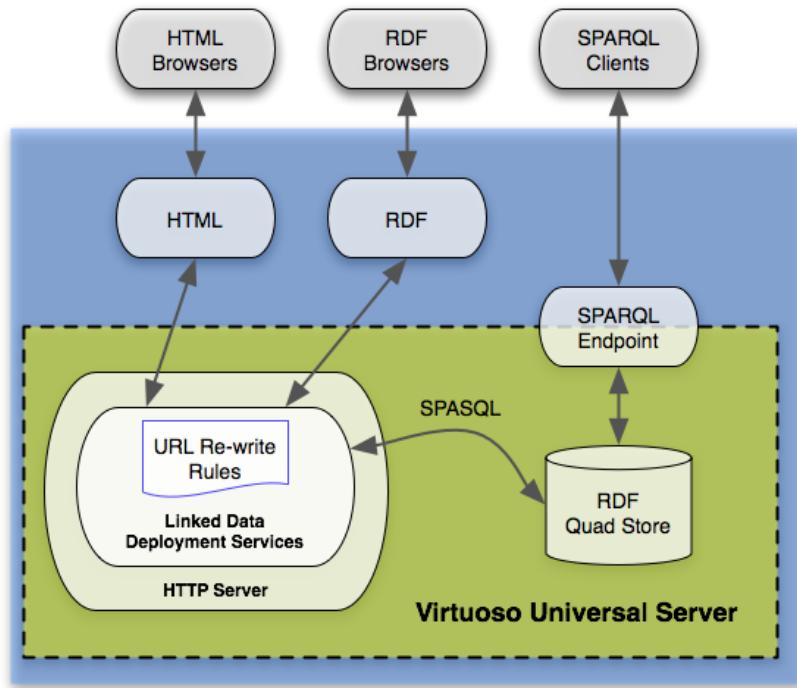


Figura 3.7: Ilustração da arquitetura do DBPedia ([DBPedia, 2017](#))

3.4.2 Google Knowledge Graph

O *Knowledge Graph* é uma base de conhecimento desenvolvida pelo Google¹⁵ para melhorar e ampliar seu mecanismo de busca ([Singhal, 2012](#)). Alinhado aos objetivos da Web Semântica, o projeto pretende expandir o buscador com um grafo de conhecimento, onde foi mapeado entidades de dados legíveis para máquinas com o intuito de recuperar a informação semântica nos termos buscados. De forma simples, o intuito é ter a informação de coisas e não de *strings*. Como exemplo, ao ser buscado a palavra "leão", não será apenas retornado uma lista de sites que possuem referências a palavra, mas também prover a semântica e taxonomias envolvidas com a ontologia e relacionados.

Com *Knowledge Graph* é possível pesquisar por pessoas, lugares, esportes, filmes e diversas informações que o Google mapeou no grafo do conhecimento. O serviço já conta com mais de 500 milhões de objetos e 3.5 bilhões de fatos sobre o relacionamento entre diferentes objetos. O objetivo do Google é ampliar o mecanismo de busca em três sentidos: Possibilitar encontrar o item certo, obter um melhor sumário, ser mais amplo e profundo. Um dos primeiros passos da companhia para atingir os objetivos do projeto é a construção do painel do sumário. Quando pesquisamos por *Leonardo*

¹⁵<https://www.google.com>

3.4. PROJETOS NA WEB SEMÂNTICA

The screenshot shows a summary card for Leonardo da Vinci. At the top, there is a large portrait of Leonardo da Vinci and a grid of smaller images related to him. Below the portrait, the name "Leonardo da Vinci" is displayed in a large, bold font, followed by the title "Cientista". A share icon is located to the right of the title. The main content area contains the following information:

Nascimento: 15 de abril de 1452, Anchiano, Itália
Falecimento: 2 de maio de 1519, Clos Lucé, Amboise, França
Em exibição: Museu do Louvre, Galeria dos Ofícios, MAIS
Períodos: Alta Renascença, Primeira Renascença, Renascimento, Renascença italiana, Escola florentina
Nome completo: Leonardo di ser Piero da Vinci
País: Caterina, Piero da Vinci
Irmãos: Bartolomeo da Vinci, Giovanni Ser Piero, MAIS

Obras de arte (with links to "Mona Lisa", "A Última Ceia", "A Anunciação", "Homem Vitruviano", and "São João Batista")

Pesquisas relacionadas (with links to "Michelan...", "Rafael", "Leonardo DiCaprio", "Vincent van Gogh", and "Sandro Botticelli")

On the right side of the card, there are "Ver mais" (View more) links for both the "Obras de arte" and "Pesquisas relacionadas" sections.

Figura 3.8: Ilustração do sumário de dados mapeados no Google Knowledge Graph.

Da Vinci, procurando seja pelas suas pinturas ou por pintores da renascença, o sistema montará um quadro de dados, conforme a Figura 3.8 com as informações, além trazer itens com relações próximas, como seus quadros e outros artistas relacionados. Com esse tratamento Singhal (2012) alega que será possível melhor compreender o que os usuários buscam, além de dar importantes passos para migrar de um motor informação para um de conhecimento, algo importante para o uso em seus assistentes virtuais.

3.5 Sumário

Neste capítulo, foi apresentado os conceitos que fundamentam o desenvolvimento de sistemas para representação de conhecimentos, assim como a abordagem e objetivos da Web Semântica sobre o tema. Em sequência foram introduzidas as principais tecnologias e princípios que são utilizados e o aprofundamento do significado das ontologias. Também foi abordado um panorama sobre a estrutura das tecnologias utilizadas na Web Semântica. Ainda foi introduzido o princípio e prática dos dados ligados e sua extensão com os dados abertos. Ainda foi exposto um panorama da similaridade semântica e tipos de medidas. Por fim, foram apresentados projetos proeminentes no cenário da Web Semântica. No capítulo 4 será discutido os conceitos da proposta do sistema de recomendação implementados neste trabalho.

4

Recomendação com similaridade semântica ponderada por links de recursos na DBpedia

You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future.

—STEVE JOBS

Desde tempos o homem busca construir ferramentas e máquinas que ampliem e sustem sua capacidade de trabalho e produção. Com o advento dos programas de máquina, o *software* tornou-se essencial para a demanda de problemas e desafios população. Como avaliado por ([Sommerville, 2010](#)), o software não se restringe a propriedades materiais das leis da física ou por processos de manufatura, o que por um lado simplifica a engenharia de software devido falta de restrições físicas, mas por outro, o torna complexo e custoso para mudanças. Assim, com a crescente quantidade e diversidade de computadores e dispositivos, é cada vez mais relevante a qualidade de software, visando a legibilidade, manutenção e evolução.

Neste capítulo serão apresentados os conceitos para a criação de sistema de recomendação com similaridade semântica ponderada por links de recursos na DBpedia¹, sendo baseado nas preferências do usuário. Serão discutidas as tecnologias, arquiteturas, modelos dos dados, recomendação e algoritmos.

¹<http://wiki.dbpedia.org>

4.1 Arquitetura

Na arquitetura do sistema proposto, existe uma camada que é responsável pela construção das recomendações que possui os seguintes principais serviços:

- **Geração de Tokens:** Um dos objetivos do sistema é recomendar itens baseando-se no conteúdo não estruturado, no caso a sinopse dos filmes. A primeira tarefa é extrair as palavras, os *tokens* relevantes dos textos, como nomes, adjetivos, lugares, entre outras, utilizando o processo de Natural Language Processing ([NLP](#)).
- **Cálculo da métrica de similaridade:** Após a geração das palavras importantes dos filmes, o sistema deve possuir um serviço para realizar o cálculo da similaridade entre dois tokens quaisquer, tirando proveito dos serviços da Web Semântica, no caso o DBpedia. Mais a frente será apresentada a equação da similaridade construída para este projeto, conforme consta na seção [4.4](#).
- **Geração das recomendações:** No último serviço, o sistema deverá construir um modelo dados através da geração de tokens para comparar as preferências do usuário com um filme, utilizando a métrica de similaridade proposta. Na seção [4.3](#) é apresentado como é construído esses modelos e na seção [4.4](#) o método utilizado para gerar a comparação e prover as recomendações.

4.2 Processo de Recomendação

A proposta é criar uma recomendação baseada em conteúdo, ou seja, nos interesses que o usuário demonstrou no passado. A *feature* analisada nos itens a serem avaliados, trata-se de um conteúdo não estruturado, no caso a descrição do item. Para este trabalho foi definido o domínio de filmes como exemplo de utilização, sendo assim, utilizando o texto da sinopse dos filmes como base para recomendação. A seguir será apresentado todas as etapas desde a captura dos dados até a apresentação das recomendações.

1. **Coleta dos filmes:** Serão coletados dados dos filmes utilizando o projeto MovieLens² (ver [4.3](#)).
2. **Pré-processamento dos filmes:** Nessa etapa após a coleta dos filmes, os dados serão previamente processados para a geração de *tokens* com [NLP](#), analisando

²<https://movielens.org>

4.2. PROCESSO DE RECOMENDAÇÃO

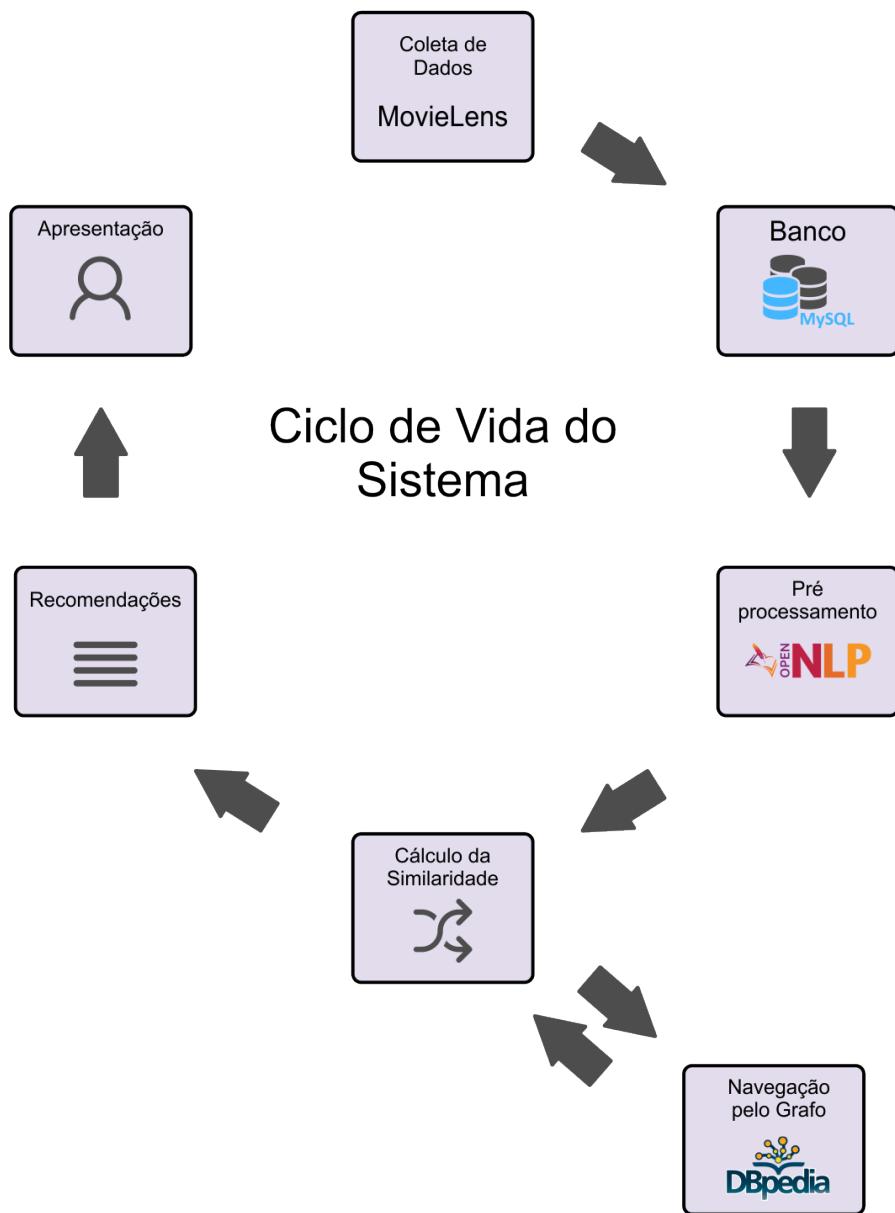


Figura 4.1: Fluxo das camadas do sistema de recomendação.

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

a descrição dos itens (sinopse dos filmes). Após processados os *tokens* serão persistidos no banco de dados.

3. **Coleta das preferências do usuário:** Serão coletadas as avaliações dos usuários pelo projeto MovieLens, podendo assim montar um perfil de preferências.
4. **Cálculo da Similaridade:** Após a etapa de pré-processamento dos filmes, será realizado o cálculo da similaridade entre os tokens do modelo do usuário e do filme, utilizando a métrica proposta [RLWS](#).
5. **Geração das recomendações:** De posse do cálculo da similaridade, será gerado um conjunto de tamanho qualquer com os melhores *scores* obtidos do cálculo desta similaridade. Posteriormente essa coleção filmes sugeridos será armazenada no banco, podendo ser atualizada conforme o perfil do usuário altera, ou novos filmes são cadastrados na base de dados. Na seção [4.4](#) é demonstrado e discutido o algoritmo central para a similaridade e recomendação.
6. **Apresentação dos resultados:** Por fim o sistema apresentará os resultados das recomendações para o usuário.

A Figura [4.1](#) mostra como esse fluxo de funcionalidades é operado por todo o sistema. A seguir será o modelo dados elaborado para este trabalho.

4.3 Modelo de dados

Para a estrutura do sistema de recomendação foi elaborado um modelo de dados para o usuário, levando em conta suas preferências, além do modelo para as informações dos filmes. Nesta seção serão apresentados como os dados estão estruturados no banco de dados, além de estabelecer um modelo formal para o usuário e o filme, prontos para serem executado pelas camadas de recomendação.

4.3.1 Banco de dados

Para trabalhar com as informações dos usuários e dos filmes, permitindo criar seus modelos para recomendação, os dados foram modelados e organizados de tal forma que possam ser facilmente persistidos e recuperados assim que necessário. A Figura [4.2](#) mostra como esses dados estão estruturados e interligados. É importante ressaltar algumas observações quanto a essa estrutura:

- A entidade *rating* persiste as avaliações dos usuários, medidas de 0 a 5. Desta entidade será construída as preferências do usuário, ou seja, extraídos aqueles filmes que possuem boas avaliações refletindo aquilo que o usuário tem interesse.
- A entidade *recommendation* persiste as sugestões de filmes calculadas pelo sistema, e o atributo “similarity” trata-se do algoritmo de similaridade utilizado, o que torna-se especialmente útil para realizar comparações com outros métodos (será discutido no 5).
- A entidade *movie* persiste os dados dos filmes retirados do projeto MovieLens³, assim como o processamento da sinopse dos filmes para geração dos *tokens*.
- A entidade *idf* persiste o cálculo do Inverse Document Frequency (IDF) da Equação 4.11, que servirá para o cálculo da similaridade cosseno, por motivos de comparação que serão discutidos no 5.
- As entidades *lod_cache* e *lod_cache_relation* tratam-se do serviço de cache para o cálculo da similaridade, assim poupano tempo para consultas do serviço do DBpedia. Na seção 4.5 a estrutura de cache dos sistemas é melhor abordada.

4.3.2 Modelo de filmes

Com a estrutura de dados utilizada para os filmes, é importante também formalizar como é seu modelo na recomendação, afinal este é o item a ser recomendado para o usuário. É definido um conjunto X como sendo conjunto de todos os filmes, onde cada filme $F \subset X$ é um conjunto de tokens únicos t_f extraídos pelo processo de NLP (ver seção 4.3.4) através da sinopse deste. Para simplificar a formalização dos modelos, nota-se que o conjunto de termos F do filme representa o filme junto com seus metadados, conforme definido em 4.3.1.

Desdessa forma formaliza-se o modelo de filmes conforme mostra as equações 4.1 e 4.2.

$$X = \bigcup_{i=1}^{\infty} F_i \quad (4.1)$$

$$F = \{t_{f_1}, \dots, t_{f_n} \mid 1 \leq n \leq \mathbb{N}^*\} \quad (4.2)$$

³<https://movielens.org>

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

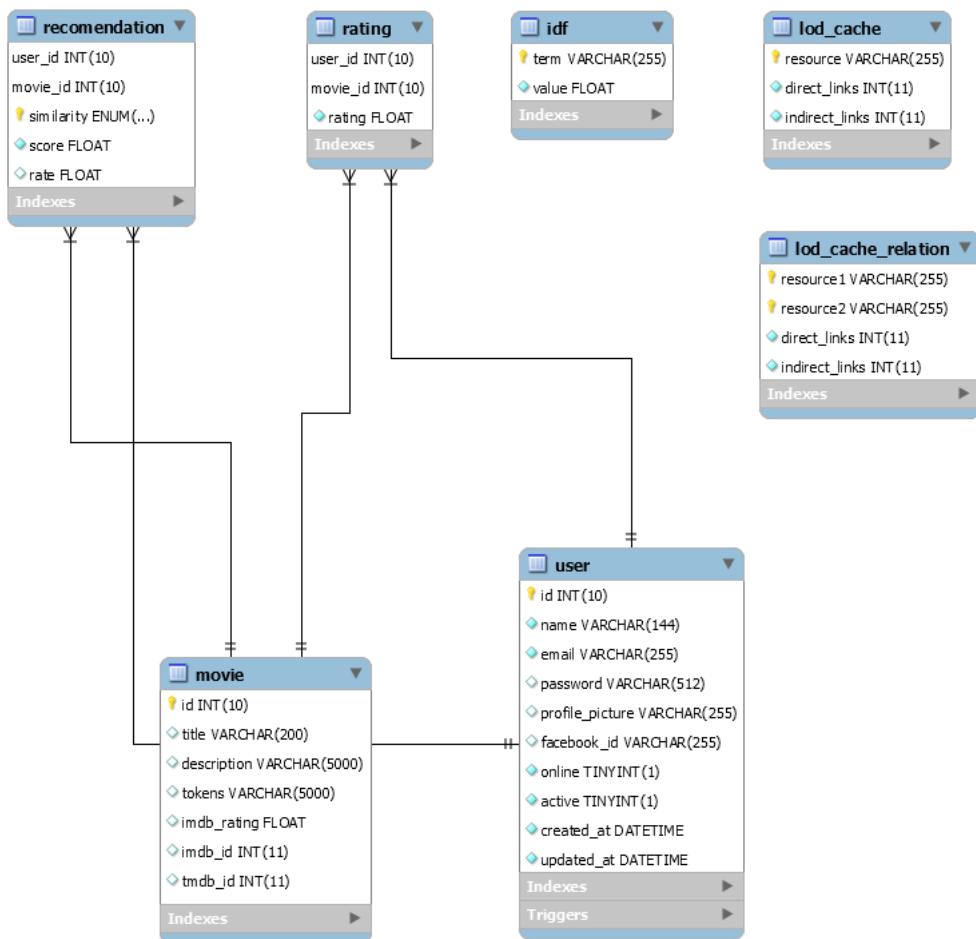


Figura 4.2: Diagrama da modelagem dos dados extraído do banco MySQL

4.3.3 Modelo de usuários

Os usuários inicialmente são estruturados como um conjunto de filmes, sendo extraídos aqueles bem avaliados por ele, à partir de uma nota de relevância r baseada num modelo de cinco estrelas, onde nenhuma estrela é totalmente irrelevante e cinco totalmente relevante. Os filmes bem avaliados serão aqueles com relevância $r \geq 3,5$, portanto sendo suas preferências. Contudo, desta forma ainda não é possível realizar a comparação com o modelo de filmes utilizando a métrica de similaridade proposta (ver 4.4), devido a comparação ser de termo a termo. Uma opção inicial seria simplesmente calcular todos os termos dos filmes de preferência do usuário, utilizado no processo de **NLP**, e uni-los num grande conjunto, porém isto tornaria o modelo de usuário muito custoso para ser utilizado, além de não escalar bem conforme as preferências do usuário aumentam. Sendo assim, optou-se por calcular os “melhores termos únicos” que representam o usuário, como sendo um conjunto de termos de um tamanho definido. Para calcular esses “melhores termos” é aplicado um modelo de frequência, o Term Frequency and Inverse Document Frequency (**TFIDF**) (ver Equação 4.11), que busca criar um ranking de termos determinando o quanto importante são numa coleção, no caso a união de todos os conjuntos filmes e seus termos. Com isso é definido Y como sendo o conjunto de todos os usuários U , e este por sua vez a união de todos os filmes F_u e P um subconjunto com aqueles de sua preferência, contendo todos os termos. Esses termos são denominados de termos do usuário, representado pelo elemento t_u . Assim como o modelo de filmes, os conjuntos U e P de termos do usuário também representam seus metadados. Posteriormente, define-se o conjunto M_u sendo o **modelo do usuário** dos melhores termos $t_u \in P$, ao passarem pela seleção da função M_{tfid} , que define um subconjunto de um tamanho definido pela constante z . Por fim, o conjunto Z é a união de todos os modelos de usuário M_u . O tamanho de M_u será explorado no capítulo 5. As equações 4.3 à 4.7 formalizam a construção do modelo do usuário.

$$Y = \bigcup_{i=1}^{\infty} U_i \quad (\text{todos os usuários}) \quad (4.3)$$

$$U = \bigcup_{i=1}^{\infty} F_{u_i} \quad (\text{todos os filmes do usuário}) \quad (4.4)$$

$$P = \bigcup_{i=1}^{\infty} F_{u_i} \quad \{F_{u_i} \mid 3,5 \leq r \leq 5 \wedge r \in \mathbb{R} \wedge \frac{r}{0,5} \in \mathbb{N}\} \quad (4.5)$$

(todos os filmes de preferência do usuário)

$$(\forall t_u \in F_u \wedge F_u \subset P) \quad M_u = \{t_u \mid M_{tfidf}(P, z), |M_u| = z \wedge z \in \mathbb{N}^*\} \quad (4.6)$$

(aplicação dos melhores termos do usuário)

$$Z = \bigcup_{i=1}^{\infty} M_{u_i} \quad (\text{todos os modelos de usuários}) \quad (4.7)$$

4.3.4 Preparação dos dados para recomendação

Antes da execução das recomendações é necessário realizar a etapa do pré-processamento dos dados. Para cada filme são gerados *tokens* que são palavras relevantes presentes na descrição. Essas palavras relevantes tratam-se do processo de exclusão daquelas que pouco agregam significado ao que se refere a temática do filme, como é o exemplo de preposições, conjunções e artigos.

Para extração dessas palavras, foi utilizado o processo denominado de **NLP** que envolve uma série de tarefas para o processamento de linguagens naturais ⁴, para compreensão e interação entre máquinas e humanos, conforme é apresentado por **Collobert et al. (2011)**. A Figura 4.3 ilustra as tarefas comuns no processamento de linguagem natural. O foco é extrair dos termos palavras com fortes significados, como adjetivos, verbos, além de também identificar substantivos, inclusive os compostos. Abaixo consta as tarefas que foram utilizadas para no processo de preparação dos dados:

- **Tokenization:** Esta tarefa consiste em segmentar um texto em partes da linguagem, sendo responsável por criar separar em tokens. Muitas vezes também é utilizado o processo de *chunking* que visa ir além de separar as palavras mas segmentar o texto em frases em partes maiores do texto, algo que não foi utilizado neste projeto.
- **Tagging, Part of Speech (POS):** O objetivo desta tarefa é reconhecer as palavras do processo de tokenization como "partes do discurso", criando *tags* para identificar o que elas representam na linguagem. Nesta parte é onde se classificam as palavras em verbos, adjetivos, substantivos, pontuações etc.
- **Name Entity Recognition (NER):** Esta tarefa objetiva reconhecer nas palavras e tokens aquelas que tratam-se de nomes próprios, além de associá-las à algum tipo, como pessoas, lugares, moedas etc.

⁴Linguagem desenvolvida naturalmente pelo ser humano de forma não premeditada, no caso a escrita

Common NLP Tasks

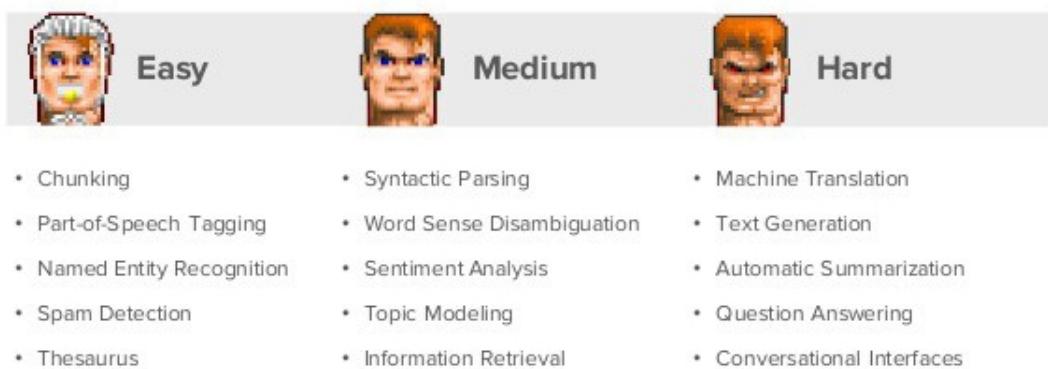


Figura 4.3: Segmentação de tarefas no NLP. ([Guts, 2016](#))

Para facilitar a execução dessas tarefas do **NLP**, optou-se por utilizar o *framework* OpenNLP⁵, conforme também abordado na seção 4.6.6, que através de um modelo de treinamento, possibilita facilmente realizar o processo de *tagging*, identificando as partes do discurso. Uma vez realizada a anotação do texto definiu-se apenas capturar as palavras marcadas com as *tags* conforme mostra a tabela 4.1. Vale ressaltar que essas definições foram concebidas para língua inglesa, uma vez que as sinopses importadas do projeto MovieLens⁶ estão nessa língua.

Em seguida é incluído ao conjunto de termos os tokens da tarefa **NER**, o que não é meramente o reconhecimento de nomes próprios do texto, mas também de nomes compostos. Esse processo é de grande valia para a etapa de similaridade uma vez que tendo um nome como "Buzz Lightyear", apenas utilizando o processo da marcação das partes do discurso, resultaria em dois termos "Buzz" e "Lightyear", sendo que o ideal seja também ter o nome composto por inteiro. Os substantivos próprios obtidos pelo processo **POS** também são mantidos no conjunto de termos, uma vez que o framework utilizado no processo de **NLP** apenas extrai, nomes, lugares e organizações. Para os nomes reconhecidos nesse processo também foi realizada uma formatação para adequação à consultas **SPARQL**, ou seja, nomes compostos como "Buzz Lightyear" são formatados para "Buzz_Lightyear", mantendo as iniciais maiúsculas e separando as palavras por "_". A tabela 4.2 demonstra alguns exemplos do tratamento da descrição de filmes para geração de *tokens*.

⁵<https://opennlp.apache.org>

⁶<https://movielens.org>

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA
POR LINKS DE RECURSOS NA DBPEDIA

Tabela 4.1: Relação das tags das partes do discurso

Tag	Descrição
NN	Substantivo comum, singular ou incontável
NNS	Substantivo comum, plural
NNP	Substantivo próprio, singular
NNPS	Substantivo próprio, plural
JJ	Adjetivo
JJR	Adjetivo comparativo
JJS	Adjetivo superlativo
FW	Palavra estrangeira
VB	Verbo, forma base

Tabela 4.2: Exemplos da geração de tokens

Filme	Sinopse	Tokens
Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	Woody, Andy, Toys, Room, Andy, Birthday, Scene, Place, Andy, Heart, Woody, Plots, Circumstances, Woody, Owner, Duo, Put, Differences, Buzz_Lightyear
GoldenEye	James Bond must unmask the mysterious head of the Janus Syndicate and prevent the leader from utilizing the GoldenEye weapons system to inflict devastating revenge on Britain.	Unmask, Mysterious, Head, Janus, Syndicate, Prevent, Leader, Goldeneye, Weapons, System, Inflict, Devastating, Revenge, Britain, James_Bond

4.4 Modelo de Recomendação

A criação de um modelo de recomendação envolve sugerir novos itens, sendo caracterizada como uma predição de o quanto provável o usuário terá interesse no conteúdo recomendado. Nesta seção serão apresentadas as etapas que existem para construção deste modelo, que envolve a definição da métrica de similaridade, através da equação da similaridade semântica (ver 4.4.1) e da construção da recomendação (ver 4.4.2).

Conforme abordado no capítulo 3, a similaridade semântica utiliza e retira proveito das estruturas de uma ontologia que neste caso trata-se das acessíveis através do DBpedia⁷. Cada termo de um filme será considerado como um potencial recurso na DBpedia, e os recursos tratam-se de entidades de ontologias mapeadas no grafo dos princípios do LOD. Diante disso, a métrica proposta é estabelecer uma equação de similaridade, utilizando-se de um modelo que considera a estrutura desses recursos, analisando a quantidade de relações e links diretos ou indiretos entre tais recursos.

Para estabelecer a equação proposta, considerou-se dois trabalhos relacionados com similaridade semântica, o de Passant (2010) que apresenta a equação Linked Data Semantic Distance (LDSD), e mais recentemente de Piao *et al.* (2016), que propõem o Resource Similarity (RESIM). O primeiro trabalho tenta estabelecer a importância que um recurso tem em relação ao total de relações da união de dois recursos, sem preferências entre links diretos e indiretos, enquanto que Piao *et al.* (2016) estende e modifica esse entendimento criando uma média ponderada entre as relações dos recursos e a similaridade entre suas propriedades.

Este trabalho propõe então um novo método denominado de RLWS, para realizar a medida da similaridade entre dois recursos presentes no DBpedia, criando uma média ponderada entre a relevância dos links diretos e indiretos, para gerar um valor na escala de 0 a 1, onde valores menores denotam menor similaridade. Além disso é apresentado o modelo de recomendação construindo uma similaridade entre um modelo que represente o usuário e suas preferências, com o modelo do filme.

Para esse modelo de recomendação foi concebida uma razão de similaridade inspirada em modelos mais complexos como o Word Mover's Distance (WMD), introduzido por Kusner *et al.* (2015), que define um método para calcular distância entre documentos utilizando um cálculo de distância entre palavras baseado no *word2vec* desenvolvido pelo Google⁸ (Mikolov *et al.*, 2013). O método WMD estabelece um peso de cada

⁷<http://wiki.dbpedia.org>

⁸<https://google.com>

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

palavra num documento e o "custo de transformação" dela em relação às outras palavras do documento a ser comparado, obtendo-se como a distância o melhor fluxo para essa "transformação", algo análogo ao problema do transporte (Oliveira, 2016). Na proposta deste trabalho não será utilizada uma distância entre palavras e sim uma similaridade com RLWS, além não ter uma distância entre documentos (no caso entre o modelo do filme e do usuário) que leve em consideração peso de palavras, realizando então um cálculo de média simplificado.

4.4.1 Equação para similaridade semântica

A Equação 4.8 demonstra o cálculo da similaridade semântica, que consiste em um conjunto de 5 funções $C_d, C_i, C_{di}, C_{do}, C_{io}$. Estas funções tratam-se de levar em consideração a quantidade links de um recurso ou entre recursos dentro de um conjunto seguindo os princípios do LOD, de acordo com a seguinte definição (Piao et al., 2016):

Definition 1. Um conjunto que segue os princípios LOD é um grafo G tal que $G = (R, L, I)$ aonde $R = \{r_1, r_2, \dots, r_n\}$ é um conjunto de recursos identificados por suas URI, $L = \{i_1, i_2, \dots, i_n\}$ é um conjunto de instâncias desses links entre recursos, como $i_i = < l_j, r_a, r_b >$.

$$RLWS(r_a, r_b) = \begin{cases} 1, & URI(r_a) = URI(r_b) \text{ ou } r_a \text{ dbo:wikiPageRedirects } r_b \\ \frac{S_d(r_a, r_b) * w_d + S_i(r_a, r_b) * w_i}{w_d + w_i}, & \text{caso contrário} \end{cases} \quad (4.8)$$

$$S_d(r_a, r_b) = 1 - \frac{1}{1 + \frac{\sum_i C_{di}(l_i, r_a, r_b) + \sum_i C_{do}(l_i, r_a, r_b)}{1 + \log(C_d(r_a) + C_d(r_b))}} \quad (4.9)$$

$$S_i(r_a, r_b) = 1 - \frac{1}{1 + \frac{\sum_i C_{ii}(l_i, r_a, r_b)}{1 + \log(C_i(r_a) + C_i(r_b))}} \quad (4.10)$$

A equação possui um condicional que implica o valor 1 quando dois recursos comparados tenham a mesma URI ou estejam relacionados pela propriedade dbo:wikiPageRedirects. Essa propriedade nada mais trata de redirecionamentos do próprio serviço do DBpedia, que quando consultando recursos como "Movie" e "Film", resultam na mesma página, pois são redirecionamentos. Para maior generalização o termo "link" será utilizado para se referir tanto a ligações como recursos ou propriedades relacionadas. As funções C_d e C_i tratam-se de computar os links distintos de um recurso qualquer, ou seja todas as



Figura 4.4: Imagem que retrata os links diretos saíentes e entrantes de um recurso.

ligações distintas a outros recursos, de forma direta e indireta respectivamente. No caso da função C_d , são computados todos os recursos distintos que sejam alcançados por uma propriedade qualquer através de um recurso analisado em questão, mais aqueles que partem de outro recurso e chegam nesse mesmo desejado. O exemplo da Figura 4.4 apresenta o recurso "Tom_Cruise" a ser calculado, que possui um link direto saínte para o recurso "American_film_producers" através da propriedade "sujeito de", e outro link direto de entrada pelo recurso "Jack_Reacher" através da propriedade "retratado por".

A contagem desses links é realizada através da consulta [SPARQL](#) conforme a Figura 4.4. Alguns filtros de propriedades são realizados, pois não são relevantes para a consulta. A propriedade `dbo:wikiPageRedirect` é realizada por outra consulta a parte, mas é adicionada no filtro, para não levá-la em consideração durante a contagem. Para a função C_i apenas são contabilizados os links indiretos de saída, por motivos de desempenho de consultas SPARQL no DBPedia. A imagem 4.5 retrata o cenário para os links indiretos, e o Código 4.2 a contagem dos links indiretos. Quanto para as funções C_{di} , C_{do} e C_{io} , referem-se a contagem dos links distintos compartilhados entre dois recursos, sendo os dois primeiros de forma direta e o último de forma indireta. Os Códigos 4.3 e 4.4 apresentam as consultas [SPARQL](#) para realizar a contagem dos links.

O objetivo da equação é obter o quanto relevante é o relacionamento entre dois recursos em relação a soma dos relacionamentos deles a quaisquer outros, obtendo S_d para uma "similaridade direta" entre r_a e r_b , e S_i para uma "similaridade indireta". Posteriormente essas similaridades são aplicadas na média ponderada com os pesos w_d e w_i , obtendo esta similaridade semântica média com o peso dos links. É importante ressaltar que para

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

```

1 PREFIX : <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT (count (distinct ?p1) as ?x)
4 WHERE {
5     {values (?r1) {(<http://dbpedia.org/resource/r1>) } ?
6      ↪ r1 ?p1 ?r2 . FILTER (?r1 != ?r2)}
7     UNION
8     {values (?r1) {(<http://dbpedia.org/resource/r1>) } ?
9      ↪ r2 ?p1 ?r1 . FILTER (?r1 != ?r2)}
10    FILTER ( ?p1 != dbo:wikiPageID )
11    FILTER ( ?p1 != dbo:wikiPageRevisionID )
12    FILTER ( ?p1 != dbo:wikiPageRedirects )
13    FILTER ( ?p1 != dbo:wikiPageExternalLink )
14    FILTER ( ! isLiteral(?r2) )
15 }
```

Código Fonte 4.1: Consulta SPARQL para contagem de links diretos

manter a equação correta é necessário que a soma dos pesos seja igual 1. A introdução das funções de log tem o objetivo de transformar os dados, suavizando o enviesamento da proporção entre os valores do total da soma de links diretos em relação aos links da comparação de r_a e r_b . De caráter ilustrativo, quando comparamos termos como "United_States" e "Group" em relação aos links indiretos, tem uma soma de 429.116 links que em comparação entre o que relaciona um ao outro é de 2.010.

Por último é destacável notar que a equação exibe os axiomas abaixo que são importantes para a consistência da similaridade:

- **Similaridade reflexiva:** $RLWS(r_a, r_a) = RLWS(r_b, r_b)$, para todo r_a e r_b .
- **Simetria:** $RLWS(r_a, r_b) = RLWS(r_b, r_a)$, para todo r_a e r_b .

4.4.2 Algoritmo da recomendação

Com equação de similaridade semântica entre dois recursos é possível comparar os termos dos filmes que por sua vez habilita a construção de um *ranking* de filmes mais similares em relação as preferências do usuário. Para montar esse perfil de preferências, conforme foi apresentado na seção 4.3.3, o usuário se tornará um conjunto de termos, podendo ser interpretado como uma *query*, no mesmo formato do modelo de filmes, onde objetivo é uma comparação entre um conjunto e outro, utilizando a similaridade proposta. Apesar de ser possível utilizar todos os termos de todos os filmes que o usuário gostou, optou-se



Figura 4.5: Imagem que retrata os links indiretos saintes de um recurso.

```

1 PREFIX : <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT (count (distinct ?r2) as ?x)
4 WHERE {
5   values (?r1) { (<http://dbpedia.org/resource/r1>) } ?
6     ↪ r2 ?p1 ?r1 . ?r2 ?p2 ?r3 . FILTER (?r1 != ?r3
7     ↪ && ?r2 != ?r1 && ?r2 != ?r3)
8   FILTER ( ?p1 != dbo:wikiPageID )
9   FILTER ( ?p1 != dbo:wikiPageRevisionID )
10  FILTER ( ?p1 != dbo:wikiPageRedirects )
11  FILTER ( ?p1 != dbo:wikiPageExternalLink )
12  FILTER ( ! isLiteral(?r2) )
13  FILTER ( ?p2 != dbo:wikiPageID )
14  FILTER ( ?p2 != dbo:wikiPageRevisionID )
15  FILTER ( ?p2 != dbo:wikiPageRedirects )
16  FILTER ( ?p2 != dbo:wikiPageExternalLink )
17 }
```

Código Fonte 4.2: Consulta SPARQL para contagem de links indiretos.

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

```
1 PREFIX : <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT (count(distinct ?p1) as ?x)
4 WHERE {
5     {values (?r1 ?r2) {(<http://dbpedia.org/resource/r1>
6         ↳ <http://dbpedia.org/resource/France>) } ?r1 ?p1
7         ↳ ?r2 . FILTER (?r1 != ?r2) }
8     UNION
9     {values (?r1 ?r2) {(<http://dbpedia.org/resource/r2>
10        ↳ <http://dbpedia.org/resource/Paris>) } ?r1 ?p1
11        ↳ ?r2 . FILTER (?r1 != ?r2) }
12         FILTER ( ?p1 != dbo:wikiPageID )
13         FILTER ( ?p1 != dbo:wikiPageRevisionID )
14         FILTER ( ?p1 != dbo:wikiPageRedirects )
15         FILTER ( ?p1 != dbo:wikiPageExternalLink )
16         FILTER ( ! isLiteral(?r2) )
17 }
```

Código Fonte 4.3: Consulta SPARQL para contagem de links diretos (saíntes e entrantes) entre dois recursos.

```
1 PREFIX : <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT (count (distinct ?r2) as ?x)
4 WHERE {
5     {values (?r1 ?r3) {(<http://dbpedia.org/resource/r1>
6         ↳ <http://dbpedia.org/resource/r2>) } ?r2 ?p1 ?r1
7         ↳ . ?r2 ?p2 ?r3 . FILTER (?r1 != ?r3 && ?r2 != ?
8             ↳ r1 && ?r2 != ?r3) }
9         FILTER ( ?p1 != dbo:wikiPageID )
10        FILTER ( ?p1 != dbo:wikiPageRevisionID )
11        FILTER ( ?p1 != dbo:wikiPageRedirects )
12        FILTER ( ?p1 != dbo:wikiPageExternalLink )
13        FILTER ( ! isLiteral(?r2) )
14        FILTER ( ?p2 != dbo:wikiPageID )
15        FILTER ( ?p2 != dbo:wikiPageRevisionID )
16        FILTER ( ?p2 != dbo:wikiPageRedirects )
17        FILTER ( ?p2 != dbo:wikiPageExternalLink )
18 }
```

Código Fonte 4.4: Consulta SPARQL para contagem de links indiretos (saíntes) entre dois recursos.

4.4. MODELO DE RECOMENDAÇÃO

por escolher uma quantidade determinada de "melhores termos únicos". Esses melhores termos são calculados através de um modelo construído pela frequência, o **TFIDF**. Esse cálculo trata-se de uma estatística que tem por objetivo de refletir o quanto importante um termo é para o documento numa coleção ([Rajaraman and Ullman, 2011](#)).

A Equação [4.12](#), trata-se da frequência do termo em relação ao documento, o que neste caso refere-se a cada termo de um filme do usuário e sua frequência em relação aos termos desse filme. Quanto a segunda, [4.13](#), refere-se ao inverso da frequência do documento, que busca balancear os termos muito frequentes em relação aos pouco frequentes, uma vez que não necessariamente todos os termos têm importância igual. Dessa forma é construída um conjunto de termos únicos de todos os filmes do usuário que para cada um deles seja contabilizado a presença na coleção dos filmes. Por fim, cada termo recebe um *score*, conforme a Equação [4.11](#).

$$TFIDF(t) = TF(t) * IDF(t) \quad (4.11)$$

$$TF(t) = \frac{f_{t,d}}{f_{t',d} : t' \in d} \quad (4.12)$$

$$IDF(t) = \log\left(\frac{N}{d \in D : t \in d}\right) \quad (4.13)$$

Com a equação pronta basta estabelecer um *ranking* de termos, escolhendo aqueles com os melhores *scores*, para montar o perfil do usuário. Nota-se que com esta equação obtém-se um score por termo de cada filme, e eventualmente este termo pode reaparecer em outro conjunto de termos de outro filme, possuindo um score diferente. Entretanto, como é estabelecido um ranking de melhores termos únicos, não há interesse na presença de dois termos iguais com scores diferentes, o interesse é montar um perfil com termos distintos e variados com melhor pontuação. Vale ressaltar que conforme estabelecido em [4.3.3](#) é definida uma constante z como sendo o tamanho do conjunto desses "melhores termos" extraídos desse ranking criado. No capítulo [5](#) o impacto do tamanho do perfil do usuário será melhor explorado. Sendo assim, é definida a Equação [4.14](#), como extensão para obtenção dos melhores termos, que avalia cada termo único t do modelo de usuário M_u , escolhendo aqueles com melhor score.

$$M_u(t) = \max TFIDF(t) \quad (4.14)$$

Definido o conjunto de termos do usuário, este será comparado com o conjunto de termos de todos os filmes que o usuário não tenha informado preferência, ou seja, que

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

ele desconheça, denominado de D_u , pois o objetivo é recomendar novos itens. Para melhor compreensão o conjunto dos termos do filme F e do usuário U , representam respectivamente seus metadados, conforme estabelecido em 4.3.1. Sendo assim, almeja-se obter os filmes com os melhores *scores* de recomendação, definido por $R_{max,U}$, que é o resultado da maximização das melhores recomendações R obtida pela equação função $S(M_u, F)$, cuja fornece a similaridade entre o modelo de usuário e o filme. As equações abaixo demonstram a formalização do cálculo da recomendação:

$$D_u = X - U \quad (\text{filmes que o usuário desconhece}) \quad (4.15)$$

$$\begin{aligned} & (\forall t_u \in M_u \wedge \forall t_f \in F) \quad recModel(U, F) = M_u \times F \\ & = \{\{t_{u_1}, t_{f_1}\}, \{t_{u_1}, t_{f_2}\}, \dots, \{t_{u_n}, t_{f_m}\}\}, n = |M_u| \wedge m = |F| \end{aligned} \quad (4.16)$$

(recModel é o produto cartesiano do modelo de usuário e do filme)

$$S(M_u, F) = \{\{t_u, f_f\} \subset M_u \times F \mid avg[RLWS_{max,t_u}(t_u, t_f)]\} \quad (4.17)$$

(a similaridade com o recModel do filme e o usuário)

$$(\forall U \subset Y \wedge F \subset D_u) \quad R_{max,U} = max_{F \subset D_u}[S(M_u, F)] \quad (4.18)$$

(Rank de recomendações maximizando S com filmes desconhecidos)

Para que a similaridade entre o modelo de usuário e o filme seja concebida, é definido o modelo de recomendação $recModel(U, F)$, como sendo a função que gera o produto cartesiano entre cada termo $t_f \in M_u$ e $t_f \in F$. Em seguida o cálculo de S trata-se da média da maximização do *RLWS* de cada termo t_u do modelo do usuário, com os termos t_f do filme, para que então seja criado o *ranking* de filmes com maior *score*. Nota-se que não haverão termos repetidos tanto no modelo do usuário como no modelo do filme, portanto cada termo "tem a mesma relevância no conjunto". Esse processo de recomendação possui ainda algumas observações, conforme descrito abaixo:

- Caso a quantidade de termos do usuário seja menor que a do filme, será feita uma comparação partindo-se dos termos do filme, assim a similaridade dos filmes mantém-se simétrica.
- Caso um dos termos a ser comparado não se trate de um recurso no DBpedia, então esta comparação é descartada e não impacta na similaridade.

4.5. ESTRUTURA DE CACHE PARA RECOMENDAÇÃO

- São escolhidos os n melhores scores das comparações entre filmes e o usuário. Caso um score seja igual, a escolha entre um e outro será aleatória.

O pseudo-código 1 exemplifica os passos para o cálculo da similaridade entre os termos do usuário e os termos do filme.

A Figura 4.6 ilustra como é o fluxo da recomendação e seu cálculo. No exemplo define-se que M_u como o conjunto de termos extraídos do texto "Lula fala com a mídia no Paraná", e por sua vez F como os termos do texto "Presidente saúda imprensa em Curitiba". O objetivo é de que dois textos **possuem palavras diferentes, mas são similares**.

4.5 Estrutura de Cache para Recomendação

É importante ressaltar que para este projeto foi desenvolvido um sistema de *cache* para o cálculo da similaridade de recomendação dos filmes, dividido-se em duas camadas:

- **Cache remoto com banco de dados:** Para calcular a similaridade entre dois tokens quaisquer, o sistema utiliza uma Equação (4.8) de similaridade que tira proveito do serviço da DBpedia. Na equação, conforme abordado em 4.4 realiza-se a contagem de links diretos e indiretos dos recursos. Essa contagem posteriormente é armazenada no banco de dados para uma consulta mais ágil.
- **Cache local, em memória:** A cada comparação realizada é verificado inicialmente se a mesma está presente no cache remoto, evitando consultas desnecessárias ao DBpedia. Após a verificação, a comparação é armazenada num cache local em memória de execução (Random Access Memory (**RAM**))). O objetivo é que durante o cálculo da recomendação de um usuário não seja necessário realizar a todo instante consultas ao banco, visto o volume de comparações, já que cada palavra dos termos do usuário é comparada com todas as palavras dos filmes. O processo de realizar tais consultas ao banco no momento da comparação prejudica o desempenho da aplicação. Assim, optou-se por utilizar a estrutura de *cache* em memória fornecida pelo framework Spring⁹. Após um determinado número de comparações o sistema verifica quais delas precisam ser persistidas, salvando-as em sequência e posteriormente liberando o cache em memória.

⁹<https://spring.io>

Algorithm 1 Pseudocódigo da geração dos filmes recomendados.

```

1: function GERA RECOMENDACOES( $U, z, n, X, w_i, w_d$ )
2:    $D_u \leftarrow X - U$                                  $\triangleright$  outros filmes
3:    $M_u \leftarrow M_{tfid}(U, z)$                    $\triangleright$  calcula os  $z$  melhores termos únicos com TFIDF
4:    $R \leftarrow \emptyset$                              $\triangleright$  conjunto vazio das recomendações
5:   for all filme  $F \subset D_u$  do
6:     if  $|M_u| > |F|$  then                       $\triangleright$  verifica quem é maior para manter simetria
7:        $R.inclui \leftarrow \text{CALCULARLWSENTRETERMOS}(M_u, F, w_i, w_d)$ 
8:     else
9:        $R.inclui \leftarrow \text{CALCULARLWSENTRETERMOS}(F, M_u, w_i, w_d)$ 
10:    end if
11:   end for
12:   return  $\max(R, n)$                           $\triangleright$  retorna o conjunto de  $n$  filmes com maior score
13: end function
14:
15: function CALCULARLWSENTRETERMOS( $M_u, F, w_i, w_d$ )  $\triangleright w_d$  - é o peso para os
   links diretos,  $w_i$  - é o peso para os links indiretos
16:    $comparacoes \leftarrow 0$                        $\triangleright$  total de comparações de termos válidos
17:    $similaridade \leftarrow \emptyset$                  $\triangleright$  conjunto vazio das recomendações
18:   for all termo  $t_u \in M_u$  do
19:      $melhorScore \leftarrow -1$ 
20:     for all termo  $t_f \in F$  do
21:        $s \leftarrow RLWS(t_u, t_f, w_i, w_d)$ 
22:       if  $s$  valido AND  $s > melhorScore$  then
23:          $melhorScore \leftarrow s$ 
24:       end if
25:     end for
26:     if  $melhorScore > -1$  then           $\triangleright$  descarta os não encontrados no DBpedia
27:        $similaridade \leftarrow similaridade + melhorScore$ 
28:        $comparacoes \leftarrow comparacoes + 1$ 
29:     end if
30:   end for
31:   if  $comparacoes > 0$  then
32:     return  $similaridade/comparacoes$ 
33:   else
34:     return 0
35:   end if
36: end function

```

4.5. ESTRUTURA DE CACHE PARA RECOMENDAÇÃO

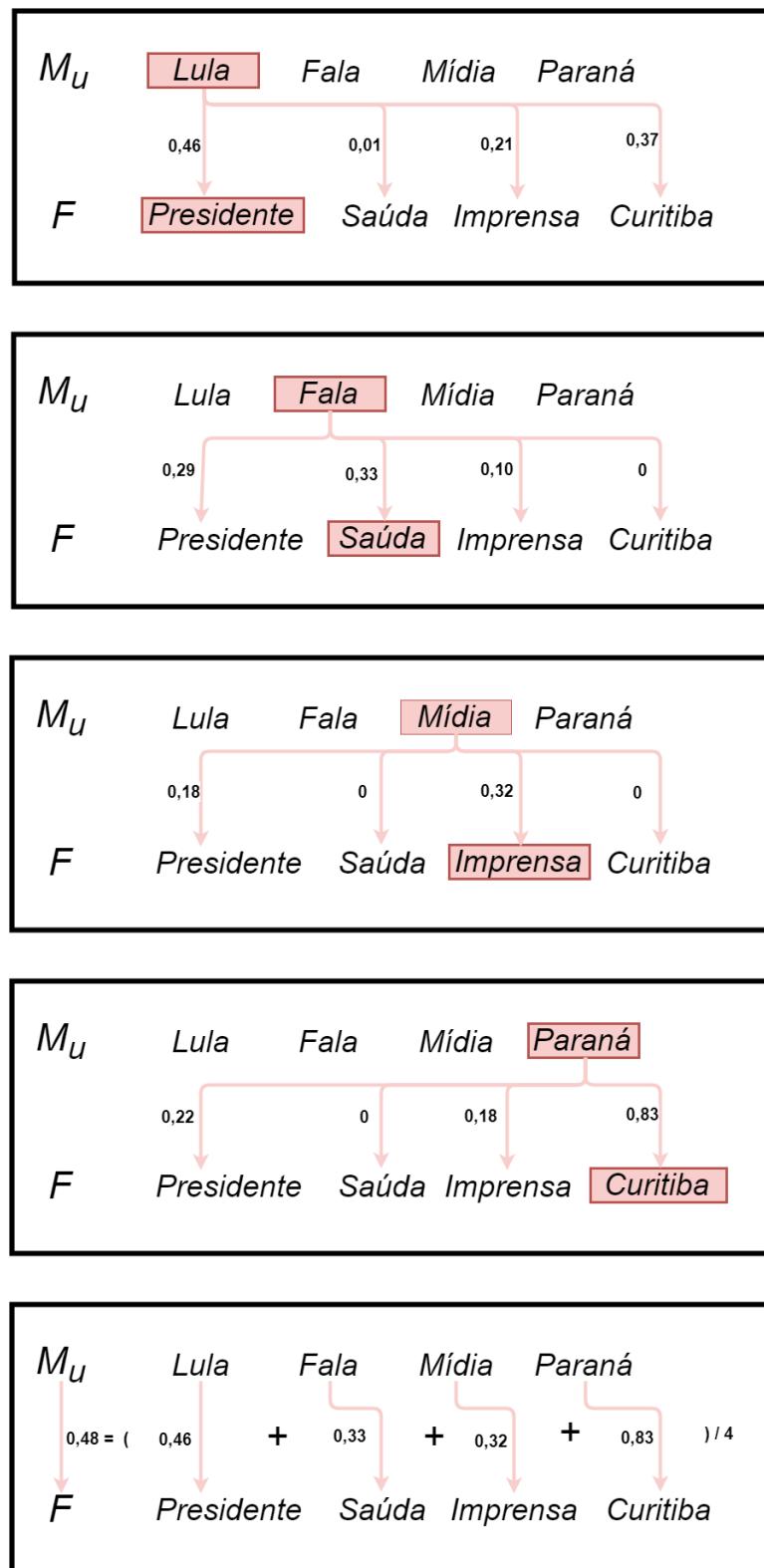


Figura 4.6: Imagem que exemplifica o fluxo da recomendação.

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

Com esse sistema minimiza-se o tráfego de dados entre a aplicação, o DBPedia e o banco de dados durante a comparação de filmes, acelerando o processo, além de ter um bom balanço de desempenho entre o cache remoto e cache local.

4.6 Tecnologias

Para o desenvolvimento do sistema foram escolhidas algumas tecnologias para arquitetura software, como linguagens de programação, *framework* Model View Controller ([MVC](#)), processamento e banco dados, entre outras. A seguir serão apresentadas as tecnologias utilizadas.

4.6.1 JAVA

JAVA¹⁰ é uma linguagem de programação de propósito genérico, desenvolvida originalmente por James Gosling na Sun Microsystems¹¹ em 1995. Atualmente a linguagem foi comprada pela Oracle Corporation¹². As características em destaque da linguagem estão no fato de ser baseada em classes e orientada a objetos. A Object Oriented Programming ([OOP](#)) é um paradigma de programação que abstrai conceitos em objetos, que podem conter dados, campos e comportamentos nomeados de *methods* ([Lewis and Loftus, 2000](#)).

Outra característica importante da linguagem trata-se da filosofia apresentada pelos desenvolvedores de “escreva uma vez, rode em qualquer lugar”. A filosofia trata-se da linguagem ser compilada por uma Virtual Machine ([VM](#)) possibilitando escrever um mesmo pedaço de código que possa ser portado para outra plataforma sem necessidade de alterá-lo, uma vez que cada [VM](#) implementa as especificidades da nova plataforma abstraindo o acesso ao Sistema Operacional ([SO](#)).

A linguagem JAVA é usada em diversos sistemas e plataformas, com inúmeros propósitos, desde aplicações *desktop*, pesquisa científica, desenvolvimento Web entre outros propósitos.

¹⁰<https://www.java.com>

¹¹<https://www.oracle.com/br/sun/index.html>

¹²<https://www.oracle.com>

4.6.2 Spring Boot

Spring Boot¹³ é um projeto da Pivotal Software¹⁴ para facilitar o processo de configuração e publicação de aplicações e serviços providos pelo Spring¹⁵, com baixo esforço e configuração. O *Spring* é um framework *open source*¹⁶ que provê um comprehensivo conjunto de modelos de configuração para aplicações JAVA. O elemento principal do *Spring* é prover infraestrutura para aplicações oferecendo os seguintes principais recursos:

- **Inversão de Controle:** Inversion of Control (**IOC**), também conhecido como *dependency injection* é um princípio em que as “dependências” devem ser supridas, injetadas por outro objeto. As dependências são objetos que serão usados como “serviços” para acessar suas funcionalidades, dentro dos *containers* de **IOC**. A injeção é a passagem da dependência para um objeto (o cliente) (**LLC, 2006**). O termo “inversão de controle” origina-se do fato que a criação de valores de classes externas ao objeto não deve ser realizada pelo próprio objeto mas, sim pelos *containers* de **IOC**.
- **Acesso a dados:** O framework possui diversas bibliotecas para o acesso a dados, tanto para bancos relacionais como não relacionais. Também é oferecido um sistema Object Relational Mapping (**ORM**) que trata-se de uma técnica para traduzir o formato de dados de um banco relacional para **OOP**, facilitando sua manipulação.
- **Arquitetura MVC:** Fornece todo suporte para customizar e criar uma arquitetura **MVC**.

4.6.3 HTML, CSS, Javascript

O HTML¹⁷, Cascading Style Sheets (**CSS**)¹⁸ e JavaScript forma a principal pilha de tecnologias utilizadas na Web. O HTML é uma linguagem de marcação mantida pela **W3C** para criação de páginas, originalmente desenvolvida por Tim-Berners-Lee (**Raggett et al., 1998**). O objetivo é a fácil construção e publicação de conteúdo no ambiente Web e consequentemente na **WWW**. No *Spring Boot* as páginas HTML podem ser escritas

¹³<https://projects.spring.io/spring-boot/>

¹⁴<https://pivotal.io>

¹⁵<https://spring.io>

¹⁶Modelo de desenvolvimento que promove um licenciamento livre para o design ou esquematização de um produto

¹⁷<https://www.w3.org/html>

¹⁸<https://www.w3.org/Style/CSS/>

CAPÍTULO 4. RECOMENDAÇÃO COM SIMILARIDADE SEMÂNTICA PONDERADA POR LINKS DE RECURSOS NA DBPEDIA

utilizando algum dos mecanismos de *templates*, como o *thymeleaf*. Uma das vantagens da utilização desses mecanismos é a herança de visualizações, assim como facilidade de interligar em manipular os dados passados pela camada do *controller* no [MVC](#).

O [CSS](#) é uma linguagem para criar regras de estilização das páginas [HTML](#). O CSS cria ou altera um formato de apresentação (tamanho, cores, margens etc) de algum elemento do HTML, como blocos, parágrafos, imagens entre outros. Quanto ao JavaScript é uma linguagem de programação originalmente criada por Brendan Eich na Netscape Communications¹⁹. A linguagem é utilizada para controlar o comportamento de páginas HTML, oferecendo dinamicidade, podendo alterar elementos da página em tempo real.

4.6.4 MySQL

O MySQL²⁰ trata-se de um Sistema de Gerenciamento de Banco de Dados ([SGBD](#)) que utiliza a linguagem [SQL](#) para manipulação de dados guardados em um sistema de arquivos ([Ventavoli, 2014](#)). Originalmente desenvolvido por Michael Widenius em 1994, o seu foco é para o desenvolvimento de aplicações Web, embora tenha se popularizado para a maioria das plataformas existentes ([DuBois, 2013](#)). Foi o banco de dados escolhido para a persistência de dados da aplicação, além de ser de fácil integração com o *framework* *Spring Boot*.

4.6.5 Apache Jena

Apache Jena²¹ é um *framework open source* para Web Semântica, escrito na linguagem Java. A biblioteca provê uma Application Programming Interface ([API](#)) que facilita a extração e criação de dados nos grafos do [RDF](#), além de oferecer suporte para a linguagem de consulta [SPARQL](#). O objetivo da escolha dessa tecnologia para o projeto, é para facilitar a busca e navegação pelo grafo de entidades (*resources*) no sistema da DBpedia²² utilizando [SPARQL](#). Após o [SR](#) extrair entidades das descrições do filme, essas serão buscadas no serviço da Web Semântica estendendo o conhecimento do recurso.

¹⁹<http://isp.netscape.com>

²⁰<https://www.mysql.com>

²¹<https://jena.apache.org>

²²<http://wiki.dbpedia.org>

4.6.6 Apache OpenNLP

Apache OpenNLP²³ é um *framework open source* de aprendizado de máquina que é usado para processamento de **NLP**. A biblioteca provê uma **API** com serviços para geração de *tokens*, sentenças, segmentação, reconhecimento de partes da fala, extração de entidade de nome, geração de *chunks* (pedaços), entre outras tarefas do **NLP**.

No projeto essa tecnologia será utilizada para o **NER** e extração de partes gramaticais presentes na descrição do filme, assim como a geração dos *tokens*. O objetivo é que com essa biblioteca seja possível gerar *tokens* com entidades encontradas, de nomes localizações, como também partes do texto de nomes próprios, substantivos e adjetivos.

4.6.7 Apache Lucene

Apache Lucene²⁴ é um *framework open source* para sistemas de recuperação de informação e recomendação. O projeto oferece dois principais recursos: indexação e pesquisa de texto. Lucene é muito reconhecido por sua utilidade na implementação em mecanismos de buscas na Internet ([McCandless et al., 2010](#)). O projeto também é muito utilizado em sistemas de recomendação com implementação de diversos algoritmos para calcular a similaridade de documentos. No projeto essa tecnologia será utilizada para tirar proveito dos algoritmos de similaridade, como o *cossine similarity* (ver [3.2](#)), possibilitando realizar comparações com as métricas propostas no trabalho.

4.7 Sumário

Neste capítulo foram apresentadas as funcionalidades para a proposta do sistema de recomendação, assim como as tecnologias empregadas. Também foram abordadas as etapas do ciclo de vida da aplicação, demonstrando o modelo de dados, assim como a etapa de preparação para recomendação. Por fim foi elaborada a proposta de um novo método de similaridade semântica, mostrando suas características e equações, além dos algoritmos para geração das recomendações. No próximo capítulo serão apresentados os resultados obtidos com novo método elaborado, junto técnicas para sua obtenção, assim como a comparação com outros modelos.

²³<https://opennlp.apache.org>

²⁴<https://lucene.apache.org>

5

Avaliação

Neste capítulo serão apresentadas as avaliações da solução proposta, metodologias utilizadas, conjunto de dados estudados, métricas e discussões sobre o significado dos resultados em relação aos objetivos inicialmente traçados. Espera-se que com o desenvolvimento de uma métrica de similaridade semântica, explorando as relações de recursos no DBpedia¹, seja possível tirar vantagem para sugerir itens, invés da análise mais sintática do conteúdo utilizado em métodos como TFIDF. Também é desejado verificar o impacto do uso da sinopse do filme, um dado não estruturado, invés de itens mais comuns como gênero, diretor, atores, com o objetivo de “fugir” das recomendações que prendam mais o usuário no mesmo tipo de filmes, mas ainda assim ser capaz de ser relevante aos seus interesses.

Inicialmente serão apresentados os dados utilizados e resultados iniciais do uso da métrica de similaridade utilizada, analisando os efeitos desejados. Posteriormente o método de recomendação que utiliza a similaridade semântica apresentado na seção 4.4.1 será comparado com o método da similaridade do cosseno, utilizando-se métricas que serão definidas e apresentadas. O resultado esperado é de que utilizando um método que leve em consideração relações semânticas tenha melhores resultados daqueles que apenas possuem análises sintáticas. Por fim, serão abordadas discussões sobre resultados alcançados além de pontos de melhoria.

5.1 Metodologia

Na avaliação de um sistema de recomendação é importante entender sua eficácia e seus algoritmos envolvidos, uma vez que uma análise incorreta pode levar subestimação ou superestimação da sua real precisão, como aponta Aggarwal (2016c). Sendo assim,

¹<http://wiki.dbpedia.org>

CAPÍTULO 5. AVALIAÇÃO

recomendadores podem ser avaliados tanto usando métodos denominados como *online* ou *offline*. Num sistema *online* as opiniões e reações dos usuários são consideradas e medidas de acordo com as recomendações apresentadas, tendo sua participação de fato, como algo crucial para a compreensão dos resultados. Contudo, como a avaliação desse método requer a participação do usuário, o que nem sempre é viável, também existe o método *offline*, onde um conjunto de diferentes tipos de dados históricos dos usuários são utilizados (Herlocker *et al.*, 1999). Para os experimentos descritos nesta seção, serão utilizados em sua maioria um conjunto de dados offline retirados do projeto MovieLens², conforme apresentado na seção 5.2. Utilizando-se da mesma metodologia também serão avaliados resultados de testes online, mas para um grupo bem pequeno, apenas para realizar uma comparação da diferença de resultados encontrados.

O objetivo dos experimentos que serão apresentados, é avaliar se a utilização da similaridade semântica junto ao método de recomendação proposto, é capaz de trazer resultados melhores nas métricas de avaliação em relação a similaridade do cosseno utilizando **TFIDF**. Os resultados tratam-se das análises das métricas extraídas das avaliações realizadas por usuários em relação as recomendações geradas por esses métodos.

Para realizar os testes entre os dois métodos de recomendação é necessário construir um perfil do usuário, formado através dos filmes que avaliou. Cada usuário deve ter pelo menos 10 avaliações. Com intuito de captar os interesses do usuário, definiu-se que para montar o modelo do usuário com seus melhores termos, serão utilizados apenas os filmes com avaliação igual ou superior a 3,5, sendo 5 a avaliação máxima. No capítulo 4 foi definido que o perfil do usuário seria um conjunto M_u dos melhores termos, possuindo um tamanho z constante. Nos experimentos a seguir foi estipulado z como 15 termos. A quantidade de termos definida possui um impacto grande na performance do sistema, uma vez que a complexidade do algoritmo da comparação entre termos é de $O(zm)$ (considerando z como constante tem-se $O(m)$), sendo m a quantidade de termos do filme. A Figura 5.1 demonstra um gráfico da quantidade de termos do modelo do usuário em relação ao tempo de processamento, considerando que todos os dados estão no *cache local*, conforme abordado no capítulo 4, ou seja, o melhor caso.

Definida a metodologia para a construção do perfil do usuário, na análise dos resultados serão realizados 3 experimentos para construir as recomendações. O total de recomendações possíveis trata-se de todos os outros filmes que o usuário não avaliou, o que torna extremamente trabalhosa a sua avaliação, portanto indo contrário aos propósitos de um **SR**, como filtrar e classificar resultados personalizados, poupando-o tempo

²<https://movielens.org>

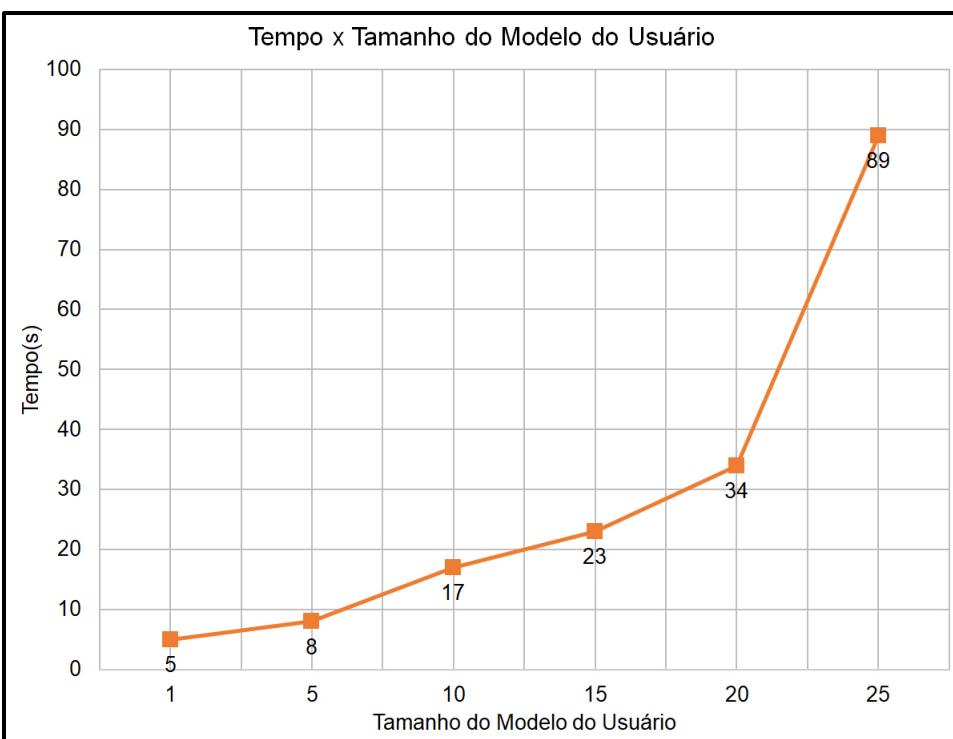


Figura 5.1: Gráfico da relação do tamanho do modelo do usuário (quantidade de termos usados) com o tempo de processamento para recomendação de um usuário. Todas as comparações de termos estão em *cache*. Execução numa máquina com processador *i7 6700K*, 16GB RAM, *Windows 10*.

CAPÍTULO 5. AVALIAÇÃO

na busca por informações. Sendo assim, apenas uma quantidade pequena de filmes serão recomendados, sendo um total de 20 recomendações com melhores scores, por experimento, uma vez que o importante é avaliar os bons primeiros resultados, ou aqueles exibidos primeiramente, pois conforme cada vez o usuário tem que continuar procurando por resultados, pior pode ser a percepção de relevância, conforme argumentado por [Manning et al. \(2008\)](#).

Os três experimentos consistem nas seguintes abordagens:

1. **Experimento 1:** Serão construídas recomendações utilizando o método apresentado na seção [4.4.1](#), definindo os pesos como 0,8 para links diretos e 0,2 para indiretos. A seguir este experimento será referido como **RLWS DIRECT**, indicando que a prioridade nos relacionamentos diretos entre recursos.
2. **Experimento 2:** Semelhante ao experimento 1, mas com pesos 0,2 para links diretos e 0,8 para indiretos. Este experimento será referido como **RLWS INDIRECT** para indicar a prioridade dos relacionamentos indiretos entre recursos.
3. **Experimento 3:** Recomendações construídas utilizando o método da similaridade do cosseno, referenciada por **COSINE**.

O objetivo da variação dos pesos para a similaridade **RLWS** é analisar o comportamento privilegiando o relacionamento direto ou indireto de recursos na DBpedia.

É importante ressaltar que para os testes *online*, os usuários serão entrevistados, fornecendo um conjunto de 10 filmes de sua preferência, além de que não será detalhado qual método foi utilizado para as recomendações geradas por cada experimento. Para cada item recomendado o usuário terá de avaliar com uma nota de 0 a 5 estrelas, onde serão considerados os itens relevantes como aqueles que possuírem 3,5 estrelas ou mais. Já para os testes offline, como não há como saber a avaliação do usuário, será utilizada uma média das avaliações de outros usuários, onde também serão consideradas relevantes aquelas avaliações com média superior ou igual a 3,5 estrelas.

Para medir os resultados das recomendações dos experimentos serão utilizadas métricas como *Precision* e *Recall*, que dependem de um modelo de classificação binária ([Powers, 2008](#)), sendo assim avaliações maiores ou iguais a 3,5 estrelas serão consideradas relevantes ou positivas, e inferiores como irrelevantes ou negativas. Na seção [5.3](#) serão melhor discutidas as métricas empregadas para avaliar os experimentos, assim como seus resultados.

Na seção seguinte será apresentado os dados trabalhados e gerados durante a execução dos experimentos.

5.2 Conjunto de dados

Os dados usados durante os experimentos tratam-se de filmes , usuários com seus termos extraídos pelo processo do **NLP**, além de avaliações. A Tabela 5.1 demonstra a quantidade de dados utilizados durante os testes. Note que os “URIs válidas” tratam-se termos extraídos das sinopses que possuem uma **URI** associada a um recurso no DBPedia³. Os “usuários, teste offline” e “online” tratam-se da quantidade de usuários utilizados do total, em que foram geradas as recomendações dos experimentos.

Dado	Quantidades
Filmes	5.107
Usuários	100.004
Usuários, teste offline	30
Usuários, teste online	4
Total de avaliações	11.997.970
Total URIs	22.978
Total URIs válidas	18.630
Total de comparações de URIs	6.306.451

Tabela 5.1: Contagem dos dados utilizados durante os testes.

O total de URIs válidas é um ponto de contenção importante de ser analisado, uma vez que se o termo não se trata de um recurso na DBPedia, a comparação do **RLWS** é descartada. Sendo assim, algo importante para a viabilidade da similaridade é de que a maioria dos termos extraídos das descrições dos filmes, tenha uma **URI** associada ao recurso. A Tabela 5.2 demonstra algumas estatísticas em relação aos termos extraídos dos filmes. Abaixo é descrito os conceitos construídos para a análise dos dados da tabela.

- **Cobertura DBPedia:** Trata-se do percentual dos termos encontrados que possuem uma **URI** associada no DBPedia. É importante que este valor seja alto, pois caso não sejam encontrados os termos no serviço da web semântica a similaridade torna-se inválida.

³<http://wiki.dbpedia.org>

CAPÍTULO 5. AVALIAÇÃO

- **Cobertura Links Diretos:** É o percentual de links que possuem pelo menos um relacionamento direto com outro termo dentro do conjunto de dados do experimento. Este valor revela o quanto útil pode ser comparar dois termos quaisquer buscando por relacionamentos diretos no DBpedia, conforme abordado no capítulo 4.
- **Cobertura Links Indiretos:** Similar à cobertura de links indiretos, mas agora sendo o percentual de links que possuem pelo menos um relacionamento indireto com outro termo.
- **Cobertura de Filmes com Links Diretos:** É o percentual de filmes contendo pelo menos um termo com ao menos um relacionamento direto a outro termo. Este valor tem o intuito de demonstrar a viabilidade da comparação utilizando a métrica [RLWS](#) para comparar filmes utilizando-se os termos extraídos das suas descrições. A ideia é de que a maioria dos filmes possam utilizar a comparação de termos, tendo pelo menos um termo com relacionamento a outro encontrado.
- **Cobertura de Filmes com Links Indiretos:** O mesmo da cobertura de filmes com links diretos, mas agora sendo o percentual para os indiretos.

Dado	Valor
Cobertura DBpedia	81,08%
Cobertura Links Diretos	21,63%
Cobertura Links Indiretos	64,41%
Cobertura de Filmes com Links Diretos	99,96%
Cobertura de Filmes com Links Indiretos	100,00%

Tabela 5.2: Estatística da cobertura dos dados dos links de recursos na DBpedia

É importante ressaltar que os dados das coberturas apenas consideram os “links válidos”, ou seja, aqueles que possuem uma URI associada no DBpedia. Nota-se também que os dados mencionados na tabela em relação a cobertura de links diretos e indiretos trata-se das comparações realizadas nos experimentos, sendo assim, não se completam, pois ainda existem links com relacionamentos que não foram comparados.

Algo relevante para destacar quanto à relação entre recursos é de que a quantidade de relacionamentos entre recursos cujo a quantidade links diretos é maior que 0, é de aproximadamente 0,12% em relação ao total de relações entre recursos, conforme constatado na Tabela 5.3. Já era esperado que a maioria dos recursos não tivesse propriedades

5.3. MÉTRICAS DE AVALIAÇÃO

diretamente conectadas entre si, devido a variedade de comparações indiscriminada entre termos do usuário e termos dos filmes. Isso resulta numa tabela de dados altamente esparsa em relação ao cálculo a participação direta na equação [RLWS](#), o que por consequência leva a uma tabela de dados grande contendo diversas comparações zeradas. Quanto a proporção de relacionamentos indiretos maiores que 0 em relação ao total é de aproximadamente 8,05%. Embora esses valores sejam baixos, o foco é de que para cada termo existam pelo menos um relacionamento a outro termo, seja direta ou indiretamente, e de que esse relacionamento esteja distribuído pela maioria dos filmes, algo que é evidenciado pelos dados da Tabela 5.2. Na tabela abaixo ainda são apresentados os de recursos relacionados que possuem a propriedade *dbo:wikiPageRedirects*, ou seja, aqueles que o DBpedia resolve sua [URI](#) como sendo a mesma, portanto para a similaridade considerados iguais.

Dado	Quantidades
Relação entre recursos	6.306.451
Relação entre recursos, direto > 0	7.977
Relação entre recursos, indireto > 0	510.533
Relação entre recursos, redirecionados entre si	194
Relação entre recursos, direto e indireto = 0	5.794.914

Tabela 5.3: Contagem da relação entre recursos utilizados durante os experimentos.

5.3 Métricas de avaliação

Existem diversas métricas que são usadas tanto em avaliações *online* e *offline*, mas as mais comuns são as de *accuracy*, embora existam outras como *user coverage*, *novelty*, *trust* ([Jannach et al., 2010](#)). Para este trabalho foram utilizadas tanto avaliações *online* e *offline*, com métodos de precisão para avaliar as classificações da recomendações, como *Precision* e *Recall*. Abaixo serão abordados os conceitos das métricas de avaliação utilizadas nos experimentos.

5.3.1 Precision

Avaliando recomendações com métodos *offline* apenas utilizando dados históricos da preferência do usuário, somente pode informar aqueles itens que foram de conhecimento

Itens recomendados: 

		Sim	Não
Usuário gostou?:	Sim	Predicções Corretas	Falso Negativos
	Não	Falso Positivos	Omissões Corretas

Figura 5.2: Tabela de tipos de erros baseada na ilustração de Jannach *et al.* (2010).

do usuário, portanto todos os outros itens terão de avaliados de outras formas, que não seja diretamente pela sua opinião, o que pode levar à falso positivos e/ou negativos. Por outro lado, avaliando com usuários reais, esses podem julgar todos os itens recomendados, podendo de fato definir se a predição foi correta ou não. Com a avaliação do usuário é possível construir uma tabela de classificação conforme a Figura 5.2 (Jannach *et al.*, 2010), onde há o cruzamento entre o que o recomendador apresentou e o que usuário avaliou. Se um item foi apresentado na recomendação e o usuário tenha gostado, avaliado como relevante, tem-se um caso de predição correta, ou *true positive*. Outro resultado positivo, trata-se de quando o usuário não tenha gostado e o recomendador omitiu o resultado, ou seja uma omissão correta ou *true negative*. Assim os resultados positivos estão na diagonal da esquerda para direita da tabela, e os resultados não desejados e negativos estão na outra diagonal.

Considerando e classificando os resultados dessa forma binária, em positivos e negativos, defini-se *precision*, precisão ou confiança, como sendo a fração de resultados previstos e avaliados pelo usuário como positivos, ou seja, os *true positive*, em relação a quantidade de todos os itens recomendados (Powers, 2008). A Equação 5.1 demonstra o cálculo da precisão P , onde tp trata-se da quantidade de itens *true positive* e fp como *false positive*.

$$P = \frac{tp}{tp + fp} \quad (5.1)$$

Como a quantidade de resultados pode ser muito grande para calcular a precisão, e até para que o próprio usuário o faça, por extensão também defini-se como $P@k$, como sendo

5.3. MÉTRICAS DE AVALIAÇÃO

Usuário 1: AveP = $1/3 (1/1 + 2/3 + 3/6) = 0,72$

Usuário 2: AveP = $1/3 (1/1 + 2/2 + 3/4) = 0,917$

$$MAP = (0,72 + 0,917) / 2 = 0,8185$$

Figura 5.3: Exemplo do cálculo do MAP.

a precisão até k primeiros resultados retornados pelo recomendador (Aggarwal, 2016c). A Equação 5.2 demonstra a variação do cálculo de P onde r trata-se da quantidade de itens relevantes até o rank k . 5.2.

$$p@k = \frac{r}{k} \quad (5.2)$$

5.3.2 Mean Average Precision (MAP)

Outra métrica de precisão trata-se da Mean Average Precision (MAP), que busca estipular um único valor de precisão em relação ao conjunto de avaliações de múltiplos usuários (Manning *et al.*, 2008). A equação 5.4 demonstra o cálculo, onde $AveP(u)$ trata-se da média das precisões $p@k$ do usuário $u \in U$, onde $|R|$ é quantidade de itens relevantes até k -ésimo rank. Posteriormente obtém-se o MAP como sendo a média $AveP$ para todos os usuários avaliados. A Figura 5.3 exemplifica o cálculo da métrica, onde os quadrados “verdes” representam os itens *true positive* e os itens vermelhos os *false positive*.

$$AveP(u) = \frac{\sum_{k=1}^n p@k}{|R|}, u \in U \quad (5.3)$$

$$MAP = \frac{\sum_{u=1}^{|U|} AveP(u)}{|U|} \quad (5.4)$$

5.3.3 Mean Reciprocal Rank (MRR)

A métrica Mean Reciprocal Rank (MRR) trata-se da média da classificação recíproca (*reciprocal rank*) de cada usuário (Burges *et al.*, 2006), sendo esta o multiplicativo inverso a posição do primeiro item correto no rank de recomendações, ou top-N itens. O objetivo é obter um valor geral que informe o quanto longe o primeiro resultado positivo está do primeiro item. A Equação 5.5 demonstra o cálculo onde $\frac{1}{k}$ trata-se do *reciprocal rank* até o k -ésimo item. A Figura 5.4 exemplifica o cálculo da métrica, onde os quadrados

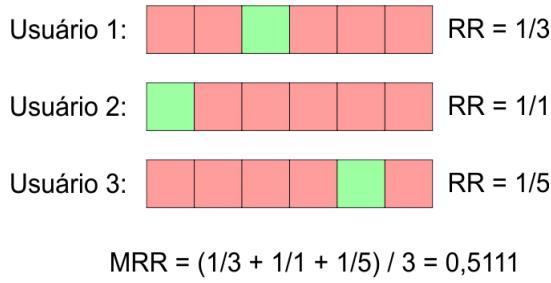


Figura 5.4: Exemplo do cálculo do MRR.

“verdes” representam os itens *true positive* e os itens vermelhos os *false positive*.

$$MRR = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{k}, \quad u \in U \quad (5.5)$$

5.4 Resultados

Antes de apresentar os resultados das recomendações com as avaliações dos usuários, é importante avaliar algumas premissas e comportamentos da própria equação de similaridade, a **RLWS**. Inicialmente o esperado é de que recursos que sejam intuitivamente próximos, ou provavelmente tenham diversas relações entre si, como *Earth* e *Moon*, possuam maior similaridade do que *Earth* e *Table*. E de fato, mesmo nos dois extremos de pesos, seja priorizando as links diretos ou indiretos, existe uma diferença considerável quando termos estão intuitivamente mais próximos do que aqueles que provavelmente não terão relacionamentos em comum, conforme mostra Figura 5.4. É importante ressaltar que “RLWS DIRECT” refere-se à configuração de pesos $w_d = 0,8, w_i = 0,2$, enquanto que “RLWS INDIRECT” é o inverso, assim priorizando os relacionamentos indiretos entre dos termos.

5.4. RESULTADOS

Termo 1	Termo 2	RLWS DIRECT	RLWS INDIRECT
France	Paris	0,434	0,858
France	Juice	0,111	0,443
France	Art	0,190	0,760
Brazil	Brasilia	0,193	0,770
Brazil	Box	0,050	0,200
Brazil	Paper	0,163	0,652
Brazil	Beach	0,282	0,726
Car	Automobile	1,0	1,0
United_States	Washington,_D,C,	0,372	0,842
China	Hong_Kong	0,377	0,842
Ariana_Grande	Selena_Gomez	0,320	0,800
Selena_Gomez	Elon_Musk	0,022	0,087
Coconut	Plant	0,393	0,683
Tom_Cruise	Lady_Gaga	0,162	0,646
Star	Galaxy	0,339	0,809
Earth	Moon	0,485	0,866
Earth	Table	0,033	0,132
Book	Movie	0,125	0,500
Book	Metal	0,096	0,386
Johnny_Cash	June_Carter_Cash	0,579	0,868
Johnny_Cash	Al_Green	0,176	0,705
Johnny_Cash	Elvis_Presley	0,316	0,816
Johnny_Cash	Kris_Kristofferson	0,317	0,804
Johnny_Cash	Carlene_Carter	0,457	0,743

Tabela 5.4: Tabela de amostra de comparações entre termos usando [RLWS](#).

É importante ressaltar que mesmo para termos que estejam aparentemente mais distantes, como *Selena_Gomez* e *Ariana_Grande*, por se tratarem de "coisas" que não são imediatamente próximas, ainda possuem uma alta similaridade, devido as conexões que ambas as pessoas possuem quanto ao domínio da música e aparições em temas de filmes, programas etc. Já quando compara-se *Selena_Gomez* com *Elon_Musk*, mesmo também sendo uma comparação entre pessoas, já possuem uma similaridade bem menor, o que também é intuitivamente esperado. Também se observa na comparação *Johnny_Cash* e

CAPÍTULO 5. AVALIAÇÃO

June_Carter_Cash, uma alta similaridade, pois os dois foram casados e cantores. Outra observação importante é quanto aos termos *Car* e *Automobile* que possuem similaridade 1. Isso é devido que os dois possuem a propriedade *dbo:wikiPageRedirect* conectando seus recursos, o que por regra na equação terá valor 1. Esses redirecionamentos também ocorrem nos termos *Future* e *Futuristic*, *Power* e *Powerful* entre outros. Para uma série de termos esses redirecionamentos contribuem para o desempenho da equação, devido a sua real proximidade, apesar de serem termos diferentes.

Nota-se que intuitivamente os resultados das comparações fazem sentido tanto usando pesos que privilegiam links diretos ou indiretos, o que é vital para coerência no momento da comparação termo a termo. Outro fato importante é a consideração de itens que possuem redirecionamentos, ainda que sejam raros, mas para palavras como "Carro" e "Automóvel" é sensato dizer que são similares.

Avaliadas as premissas e os resultados para a similaridade semântica proposta, em sequência serão calculados as métricas mencionadas na seção 5.3, medindo assim o impacto do modelo de recomendação assim como a própria similaridade RLWS. Os testes dos experimentos mencionados na seção 5.1 serão divididos em dois resultados, os testes com dados de “usuários offline” e “usuários online”.

5.4.1 MAP

Para medir a assertividade do SR foi utilizada a métrica MAP perante dois conjuntos de usuários, os dos testes *offline* (sendo no total de 30) e os *online* (com total de 4). As Figuras 5.5 e 5.6 apresentam os resultados obtidos nos três experimentos, tanto para os testes online quanto para os offline.

O mais notável dos valores obtidos é o fato de que os testes do experimento 2, com *RLWS INDIRECT*, teve um desempenho consideravelmente melhor que os outros nos dois testes, em especial a seu par que privilegia os links diretos, sendo 51% superior (0,532 contra 0,351) ao *COSINE* para um conjunto de cinco itens ($k = 5$) nos testes *offline*. Quanto aos resultados utilizando a métrica *RLWS DIRECT*, há uma divergência entre os testes das Figuras 5.5a e 5.5b, sendo que o primeiro teste obteve um desempenho próximo à métrica *COSINE*, na verdade até perdendo, enquanto que nos testes *offline* ganha em 27% (0,447 contra 0,351) para os itens até o *rank* 5. Nos testes offline ambas as variantes *RLWS* obtiveram um desempenho consideravelmente superior à similaridade do cosseno, já para os testes *online* o maior destaque vai para o experimento 2, com *RLWS INDIRECT*.

Mesmo com um desempenho inferior, o experimento *RLWS DIRECT* obteve um

5.4. RESULTADOS

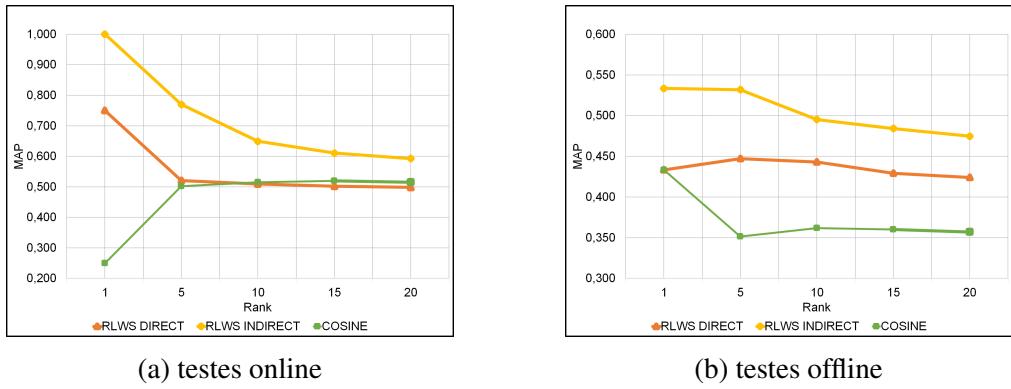


Figura 5.5: MAP - Gráfico com linhas dos três experimentos nos testes online e offline

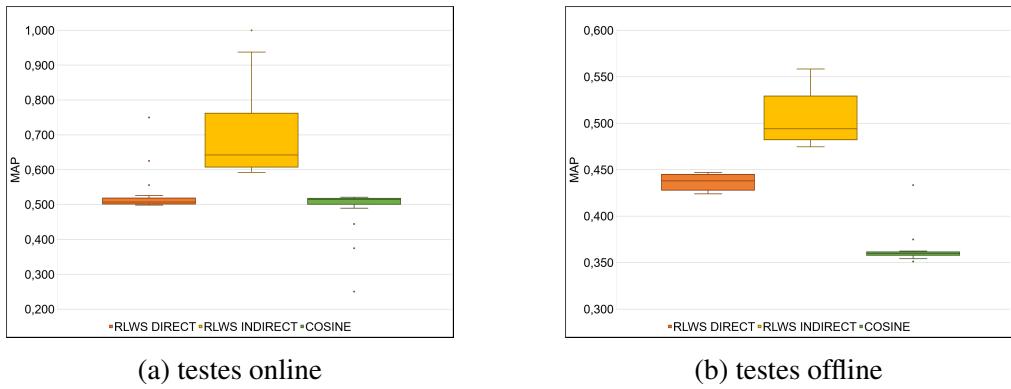


Figura 5.6: MAP - Gráfico de caixa dos três experimentos nos testes online e offline

resultado próximo da similaridade do cosseno nos testes online, o que indica que com um peso um pouco menor para os relacionamentos diretos como $w_d = 0,65, w_i = 0,35$, pode-se obter melhores resultados. A influencia dos relacionamentos diretos é esperada que tenha um impacto menor numa visão geral, devido a improbabilidade de um termo possuir tal conexão com outro, conforme é evidenciado pelas estatísticas apresentadas nas Tabelas 5.2 e 5.3. Outra observação importante é que para os 5 primeiros itens as duas métricas RLWS possuem uma maior precisão, o que pode ser notado tanto para os limites superiores e *outliers* nas Figuras 5.6a e 5.6b.

5.4.2 MRR

Para avaliar a objetividade do SR foi utilizada a métrica MRR perante dois conjuntos de usuários, os dos testes *offline* (sendo no total de 30) e os *online* (com total de 4). As Figuras 5.7 e 5.8 apresentam os resultados obtidos nos três experimentos para ambos os testes.

CAPÍTULO 5. AVALIAÇÃO

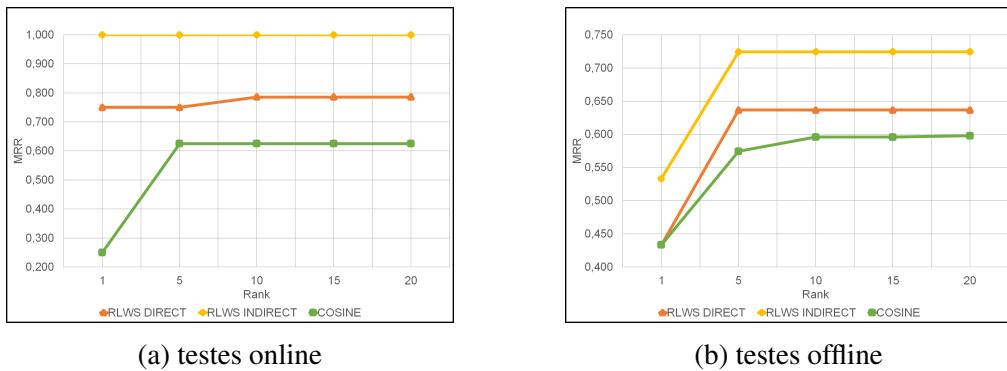


Figura 5.7: MRR - Gráfico com linhas dos três experimentos nos testes online e offline

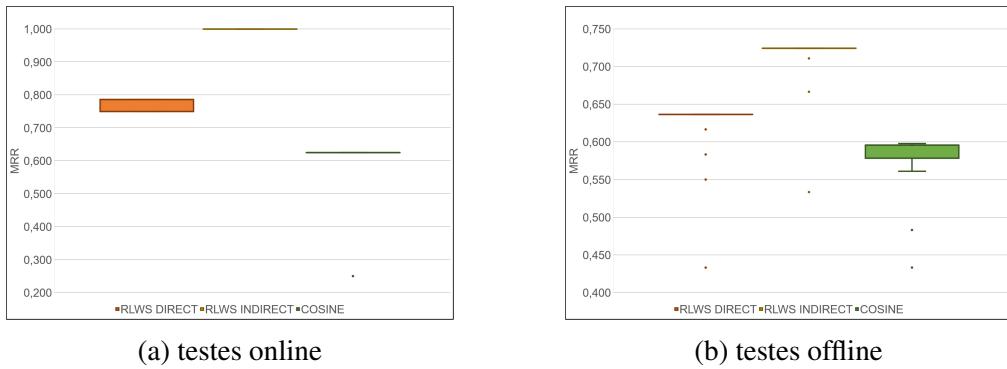


Figura 5.8: MRR - Gráfico de caixa dos três experimentos nos testes online e offline

Assim como na métrica MAP, também é possível observar um desempenho consideravelmente superior para o experimento 2 com a similaridade RLWS INDIRECT, sendo 26% (0,724 contra 0,574) superior a similaridade do cosseno nos testes offline, para os 5 primeiros itens. Quanto ao outro par RLWS DIRECT, obteve um resultado 11% (0,637 contra 0,574) superior nos testes offline. Nota-se que na Figura 5.7a tem-se uma linha reta com a maior nota possível, indicando que para todos os usuários nos testes online, um resultado positivo foi encontrado logo no primeiro item recomendado. Já para os testes offline é possível observar que para ambos os experimentos da métrica RLWS foram encontrados resultados positivos já no primeiro item para mais da metade dos usuários, com o experimento COSINE ficando um pouco abaixo.

Os gráficos de caixa aparentes nas Figuras 5.8a e 5.8b, possuem em sua maioria apenas linhas retas juntamente com os outliers devido ao fato de que a partir da 5º posição não há muito mais usuários que ainda não obtiveram uma recomendação positiva.

5.5 Discussão dos resultados

Utilizando as métricas apresentadas na metodologia foi possível observar que em especial ao uso de relacionamentos indiretos entre termos, obteve-se um resultado consideravelmente superior a métrica da similaridade do cosseno. Já quanto a variante **RLWS DIRECT**, obteve um resultado misto, sendo na maior parte das vezes superior em relação a métrica comparada. Ainda que com um grupo de testes limitado, os resultados indicam que é possível tirar proveito de uma análise semântica entre termos, podendo trazer resultados melhores de que uma análise sintática. Outra observação trata-se de que durante as entrevistas dos usuários nos testes *online*, foi notado que o uso da descrição dos itens como *feature* para construção do modelo de recomendação, aparentemente gerou recomendações que “fugiram à bolha tradicional” de resultados, algo que era um dos pontos do problema apresentado no capítulo 1. Contudo, para um diversa quantidade de usuários o uso de certas *features* dos itens, como ano de lançamento, gênero, autor ainda podem obter bons resultados, em especial de que é difícil avaliar se o modelo do usuário baseado na sinopse dos filmes, obtendo os termos mais relevantes, consegue captar corretamente suas preferências de narrativa, além de que o uso de outras características dos itens podem ser mais relevantes de usuário para usuário.

Outro ponto importante para ressaltar trata-se de que como o modelo de recomendação (ver seção 4.4.2) utiliza a descrição do item, um dado não estruturado, torna o sistema não muito dependente do domínio aplicado, sendo perfeitamente possível de ser utilizado numa recomendação para livros, por exemplo. Ainda que a similaridade semântica proposta seja intuitivamente coerente e de que é possível utilizá-la para modelar recomendações potencialmente mais precisas que métodos tradicionais, não seria tão adequada para comparações de textos grandes, sendo o objetivo para modelar um **SR**, devido ao alto custo e carga de consultas a serem realizadas no DBpedia. Para comparar recursos individualmente ou uma coleção de palavras em pequenos textos, seria o uso mais indicado para tal similaridade.

5.6 Pontos de melhorias

Alguns desafios foram encontrados durante a construção do sistema de recomendação, principalmente relacionados a métrica de similaridade. Notou-se que a ambiguidade é um grande problema para análise de recursos na DBpedia⁴, inclusive ambiguidade em

⁴<http://wiki.dbpedia.org>

CAPÍTULO 5. AVALIAÇÃO

dois sentidos. O primeiro deve-se à ambiguidade no próprio serviço da web semântica que é vista pela propriedade *dbo:wikiPageDisambiguates*, como no exemplo do termo “Paraná”, que pode ser resolvido para o estado brasileiro, rio ou clube de futebol. Quando uma pesquisa com termo ambíguo é realizada a contagem de links torna-se menos eficaz do que feita sabendo qual termo específico está se referindo.

O segundo sentido de ambiguidade é a da gerada pela propriedade *dbo:wikiPageRedirects*, que trata-se de como o termo é resolvido para seu nome final, comportando-se como *alias*. Quando se busca um termo como *Automobile*, o mesmo é redirecionado para *Car* como [URI](#) final, isso para todos os possíveis “sinônimos”. Ainda que no sistema sejam considerados termos iguais devido à análise de redirecionamentos entre si, quando se compara com outro termo, existe uma divergência para pior na contagem de links indiretos. Em futuros trabalhos poderá ser feito um ajuste nas consultas para que a contagem de links diretos e indiretos sejam mais precisas, diminuindo os problemas de ambiguidade.

O custo para se executar o modelo de recomendação proposto, é outro ponto de melhoria, pois ainda que o limite superior da complexidade de tempo seja $O(|Y| * |M_u| * |D_u|)$, dentro do esperado, o modelo do usuário e de filmes tendo um custo $|M_u| * |D_u|$, na prática torna-se um conjunto de termos grande para ser executado no DBpedia. É possível identificar este problema pela Equação 5.6 que demonstra o cálculo de todas as combinações de termos possíveis, sendo n o total de termos associados a [URI](#), que apenas utilizando os 5.107 filmes, foram encontrados 18630, resultado num total de 173.529.135 comparações possíveis. Ainda que este seja o limite teórico, e de que muitos termos não venham a ser comparados, é possível notar o quanto este número pode crescer rapidamente, sendo uma alta carga de comparações, mesmo utilizando *multithreading*⁵ e a estrutura de *cache* da contagem de links proposta na seção 4.5.

$$C(n, 2) = \frac{n!}{2!(n-2)!} \quad (5.6)$$

O objetivo não é alterar o custo dado pela equação das combinações, mas verificar maneiras de diminuir os modelos para recomendação, assim reduzindo a carga teórica de consultas no DBpedia. Inclusive este problema é um dos fatores que dificultaram a execução de testes com mais usuários, algo que precisa de mais tempo para ter um conjunto maior de comparações. Com essas melhorias realizadas nos modelos, será possível expandir mais facilmente os testes *online*, inclusive relacionado a este ponto

⁵Multithreading trata-se da possibilidade de trabalhar com múltiplas linhas execução concorrente, tornando um processamento de dados paralelo invés de linear.

também facilitaria a construção de uma plataforma online na Web para que o usuário pudesse escolher suas preferências, exibindo as recomendações para que ele mesmo avalie, dispensando o uso de entrevistas.

Outra observação importante para melhorar, trata-se do cálculo com o modelo de recomendação, onde realiza-se uma simples média do $RLWS_{max,t_u}$ de cada termo t_u . A diferença do tamanho entre os modelos não é levada em conta, podendo então ser aplicado algum peso, como a frequência. Seria relevante aprofundar nessa comparação, estudando variações e formas semelhantes como a [WMD](#), que realiza um cálculo do "custo de transformação" de um conjunto de termos no outro, utilizando conceitos do problema do transporte ([Oliveira, 2016](#)). Uma variação poderia ser invés de utilizar uma métrica de distância como a *word2vec* ([Mikolov et al., 2013](#)), aplicar uma que utilize similaridade (ou distância) semântica.

Um último ponto de melhoria trata-se do processo de [NLP](#), onde foram utilizados modelos pré-treinados oferecidos pela biblioteca *OpenNLP*⁶. Com o uso deste modelo ocorreram diversas falhas ao reconhecer nomes de entidades (processo [NER](#)). Para isso existem algumas possibilidades, como a de realizar um treinamento baseado nos textos dos filmes do próprio projeto, ou experimentar outras ferramentas como a *Stanford CoreNLP*⁷.

5.7 Sumário

Neste capítulo foram apresentados os resultados obtidos com o sistema de recomendação construído, assim como uma análise da métrica de similaridade semântica. Para isso foram discutidas as metodologias junto com os dados utilizadas, além de estudadas as métricas que foram usadas nos resultados.

⁶<https://opennlp.apache.org>

⁷<https://stanfordnlp.github.io/CoreNLP/>

6

Conclusão

São só dois lados da mesma viagem. O trem que chega é o mesmo trem da partida. A hora do encontro é também de despedida

—ENCONTROS E DESPEDIDAS, MARIA RITA

Neste trabalho de conclusão de curso foi apresentada uma proposta de recomendação com similaridade semântica ponderada por links de recursos na DBpedia¹. Inicialmente estabeleceu-se a motivação pela busca de novos resultados originados da similaridade semântica de elementos. Ainda foram discutidos os problemas que a solução proposta pretende confrontar, como a deficiência e dificuldade de sugerir quando apenas o conteúdo sintático é analisado, além da possibilidade de usuários expandirem sua busca para a maioria dos itens comumente recomendados utilizando *features* tradicionais na recomendação baseada em conteúdo.

No capítulo 2 foi introduzido referenciais sobre a construção de um sistema de recomendação, com suas variações, propósitos, além da apresentação de exemplos do estado da arte encontrada em diversas soluções utilizadas na literatura e no mercado.

Para o capítulo 3 foram apresentados os conceitos da Web Semântica, como sua formulação, tecnologias e princípios, além do aprofundamento do significado das ontologias. Também foi abordado um panorama sobre a estrutura das tecnologias utilizadas na Web Semântica, a prática dos dados ligados, além do panorama da similaridade semântica e tipos de medidas.

No capítulo 4 é detalhada a proposta para o sistema de recomendação, conceituando as etapas para construir o conjunto de itens recomendados. Também foram apresentados os modelos de dados, do usuário, dos itens e do modelo da recomendação, juntamente

¹<http://wiki.dbpedia.org>

CAPÍTULO 6. CONCLUSÃO

com a introdução dos algoritmos e a métrica de similaridade semântica da proposta deste trabalho, com suas equações.

Por último foram vistos os resultados da proposta deste trabalho, além da discussão sobre métricas de avaliação em sistemas de recomendação. Também foram discutidos os resultados obtidos pelas métricas avaliadas, além de apresentados pontos de melhoria.

6.1 Contribuições

Abaixo consta um resumo das principais contribuições oferecidas por este trabalho:

- **Métrica de similaridade semântica RLWS:** Foi proposta uma nova métrica de similaridade semântica ponderada por links de recursos na DBpedia. O foco é analisar o peso de relacionamentos (a quantidade de links) diretos e indiretos entre tais recursos, realizando uma média simples entre os dois aspectos.
- **Modelo de recomendação:** Foi apresentado um modelo de recomendação baseado em conteúdo utilizando um dado não estruturado, juntamente com um processamento de linguagem natural integrado com uma métrica de similaridade semântica.
- **Avaliação da recomendação:** Foram discutidos os benefícios de um **SR** com o uso além de *features* tradicionais que apenas realizam uma comparação sintática, introduzindo assim um método para similaridade semântica.

6.2 Trabalhos Futuros

Resumindo os pontos discutidos no capítulo 5, são levantados os seguintes pontos para trabalhos futuros:

- **Trabalhar na precisão das consultas SPARQL:** Conforme relatado no capítulo 5, a comparação de termos entre o modelo de usuário e do filme gera ambiguidades durante as consultas no DBpedia, principalmente aquelas relacionadas a propriedade *dbo:wikiPageRedirects*, o que afeta a contagem de links indiretos. Objetivo é trabalhar neste tipo de ambiguidade melhorando os resultados das comparações de termos.
- **Aprimorar a extração de termos com NLP:** Revisar o processo de extração de termos, uma vez que o mesmo possui falhas na identificação de entidades nomeadas.

- **Proposta de modelos menores:** O tamanho dos modelos do usuário, do filme juntamente com o modelo de recomendação implicam em uma carga muito grande para consultas no DBpedia, sendo assim o objetivo é estudar variações para diminuir a quantidade de termos comparados.
- **Realizar testes com mais usuários:** Devido ao custo inicial da execução da recomendação de itens, sem possuir informações em *cache* sobre as comparações de termos, foram realizados testes com poucos usuários.
- **Plataforma Web para testes:** Para facilitar a realização de mais testes *online* nos resultados, é de grande valia construir uma plataforma online na Web para que o usuário informe suas preferências além de avaliar as recomendações.
- **Melhorias na similaridade entre conjunto de termos:** Estudar diferentes formas de calcular a similaridade entre o modelo de usuário e o modelo de filmes (ver 4.4), buscando variações a medidas como o [WMD](#), ainda utilizando [RLWS](#).

6.3 Sumário

Este capítulo apresentou um resumo geral realizado neste trabalho, além de um panorama das contribuições e tarefas para serem realizadas em trabalhos futuros.

Referências Bibliográficas

- (2009). Skoob: Socializando o ato da leitura. Disponível em: <https://archive.is/TwnCc>. [Último acesso em 30 de Novembro de 2017].
- (2009). Web semântica: O futuro das aplicações - java magazine 85. Disponível em: <https://www.devmedia.com.br/web-semantica-o-futuro-das-aplicacoes-javascript-jquery-jquery-mvc-85/18493#>. [Último acesso em 02 de Novembro de 2017].
- (2017). Netflix officially kills star ratings, replacing them with thumbs up and down. Disponível em: <http://variety.com/2017/digital/news/netflix-kills-star-ratings-thumbs-up-thumbs-down-1202023257/>. [Último acesso em 30 de Novembro de 2017].
- (2017). Semantic web. Disponível em: https://en.wikipedia.org/wiki/Semantic_Web. [Último acesso em 02 de Novembro de 2017].
- (2017). Skoob: Quem somos. Disponível em: https://www.skoob.com.br/inicio/quem_somos. [Último acesso em 30 de Novembro de 2017].
- Aggarwal, C. C. (2016a). An introduction to recommender systems. In *Recommender Systems*, pages 1–28. Springer International Publishing.
- Aggarwal, C. C. (2016b). *Recommender Systems*, pages 225–255. Springer International Publishing.
- Aggarwal, C. C. (2016c). *Recommender Systems*, pages 139–166. Springer International Publishing.
- Alexandra, H. (2017). Activision patents matchmaking that encourages players to buy microtransactions. Disponível em: <https://kotaku.com/1819630937>. [Último acesso em 13 de Novembro de 2019].
- Andrejs Abele, J. M. (2020). The linking open data cloud diagram. Disponível em: <http://lod-cloud.net/>. [Último acesso em 21 de Abril de 2021].
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

REFERÊNCIAS BIBLIOGRÁFICAS

- Baubonienė and Gulevičiūtė (2015). E-commerce factors influencing consumers online shopping decision.
- Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.
- Berners-Lee, T. (2006). Linked data. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. [Último acesso em 03 de Novembro de 2017].
- Berners-Lee, T. (2008). Semantic web architecture. Disponível em: <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slides10-0.html>. [Último acesso em 03 de Novembro de 2017].
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, **284**(5), 34–43.
- Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, **5**(3), 1–22.
- Burges, C. J. C., Ragno, R., and Le, Q. V. (2006). Learning to rank with nonsmooth cost functions. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 193–200, Cambridge, MA, USA. MIT Press.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, **12**(4), 331–370.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, **12**, 2493–2537.
- DBpedia (2014). Dbpedia 2014 data set statistics. Disponível em: <http://wiki.dbpedia.org/services-resources/datasets/dataset-statistics>. [Último acesso em 03 de Novembro de 2017].
- DBpedia (2017). Learn about dbpedia. Disponível em: <http://wiki.dbpedia.org/about>. [Último acesso em 03 de Novembro de 2017].
- del Toro, G., Navarro, B., Cuarón, A., Torresblanco, F., and Augustin, A. (2006). El laberinto del fauno.

REFERÊNCIAS BIBLIOGRÁFICAS

- DuBois, P. (2013). *MySQL (Developer's Library)*. Addison-Wesley Professional.
- Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, **4**(2), 81–173.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, **35**(12), 61–70.
- Gracia, J. and Mena, E. (2008). *Web-Based Measure of Semantic Relatedness*, pages 136–150. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Guts, Y. (2016). Natural language processing (nlp). Disponível em: <https://www.slideshare.net/YuriyGuts/natural-language-processing-nlp>. [Último acesso em 21 de Fevereiro de 2018].
- Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’99, pages 230–237, New York, NY, USA. ACM.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, **22**(1), 5–53.
- Isinkaye, F., Folajimi, Y., and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, **16**(3), 261 – 273.
- Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition.
- Júnio César de Lima, C. L. d. C. (2005). Ontologias - owl (web ontology language). techreport, Instituto de Informática Universidade Federal de Goiás.
- Keating, G. (2012). *Netflixed: The Epic Battle for America’s Eyeballs*. Portfolio/Penguin.
- Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, **22**(1), 101–123.
- Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

REFERÊNCIAS BIBLIOGRÁFICAS

- Lewis, J. and Loftus, W. (2000). *Java Software Solutions: Foundations of Program Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.
- Lin, D. (1993). Principle-based parsing without overgeneration. In *ACL*.
- LLC, T. I. (2006). Dependency injection demystified. Disponível em: <http://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>. [Último acesso em 21 de Fevereiro de 2018].
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- McCandless, M., Hatcher, E., and Gospodnetic, O. (2010). *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA.
- Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space.
- Oliveira, A. R. L. d. O. (2016). O problema de transporte. Universidade Estadual de Campinas.
- Pariser, E. (2011). *The Filter Bubble: What the Internet is Hiding from You*.
- Passant, A. (2010). Measuring semantic distance on linking data and using it for resources recommendations.
- Patil, R. S., Fikes, R., Patel-Schneider, P. F., McKay, D. P., Finin, T. W., Gruber, T. R., and Neches, R. (1992). The darpa knowledge sharing effort: A progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, KR'92, pages 777–788, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Piao, G., Ara, S. s., and Breslin, J. G. (2016). Computing the semantic similarity of resources in dbpedia for recommendation purposes. In G. Qi, K. Kozaki, J. Z. Pan, and S. Yu, editors, *Semantic Technology*, pages 185–200, Cham. Springer International Publishing.
- Powers, D. (2008). Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Mach. Learn. Technol.*, 2.

REFERÊNCIAS BIBLIOGRÁFICAS

- Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 17–30.
- Raggett, D., Lam, J., Alexander, I., and Kmiec, M. (1998). *Raggett on HTML 4 (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Rajaraman, A. and Ullman, J. D. (2011). *Data Mining*, page 1–17. Cambridge University Press.
- Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, **40**(3), 56–58.
- Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Int. Res.*, **11**(1), 95–130.
- Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA.
- Singhal, A. (2001). Modern information retrieval: A brief overview. **24**, 35–43.
- Singhal, A. (2012). Introducing the knowledge graph: things, not strings. Disponível em: <https://googleblog.blogspot.com.br/2012/05/introducing-knowledge-graph-things-not.html>. [Último acesso em 03 de Novembro de 2017].
- Slimani, T. (2013). Description and evaluation of semantic similarity measures approaches. *International Journal of Computer Applications*, **80**(10), 25–33.
- Snyder, D. and Snyder, Z. (2011). Sucker punch.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition.
- Tversky, A. (1977). Features of similarity. **84**, 327–352.
- Ventavoli, F. (2014). *Sistema Gerenciador de Banco de Dados MySQL: Guia Prático (Portuguese Edition)*.
- VIANA NETO, J. Q. (2010). Skoob: ambiente virtual de socialização entre leitores e produtores de textos. resreport, Universidade Federal de Pernambuco.

REFERÊNCIAS BIBLIOGRÁFICAS

- W3C (2001). W3c semantic web activity. Disponível em: <https://www.w3.org/2001/sw/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2006). Linked data. Disponível em: <https://www.w3.org/standards/semanticweb/data>. [Último acesso em 03 de Novembro de 2017].
- W3C (2009). Owl web ontology language guide. Disponível em: <https://www.w3.org/TR/owl-guide/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2013). Sparql query language for rdf. Disponível em: <https://www.w3.org/TR/rdf-sparql-query/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2014). Rdf. Disponível em: <https://www.w3.org/RDF/>. [Último acesso em 02 de Novembro de 2017].
- Walker, S. J. (2014). Big data: A revolution that will transform how we live, work, and think. *International Journal of Advertising*, 33(1), 181–183.
- Wellman, D. (2013). What is big data? Disponível em: <https://www.slideshare.net/dwellman/what-is-big-data-24401517>. [Último acesso em 23 de Outubro de 2017].
- Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.