UNIVERSIDADE FEDERAL DE OURO PRETO

UMA ABORDAGEM PARA ESTIMAR A SIMILARIDADE ITEM-ITEM BASEADA NOS RELACIONAMENTOS SEMÂNTICOS DA LINKED OPEN DATA

Italo Magno Pereira

Orientador: Anderson Almeida Ferreira

Ouro Preto
Novembro de 2019

UNIVERSIDADE FEDERAL DE OURO PRETO

UMA ABORDAGEM PARA ESTIMAR A SIMILARIDADE ITEM-ITEM BASEADA NOS RELACIONAMENTOS SEMÂNTICOS DA LINKED OPEN DATA

Italo Magno Pereira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, da Universidade Federal de Ouro Preto, como parte dos requisitos necessários à obtenção do título de Mestre em Ciência da Computação.

Orientador: Anderson Almeida Ferreira

Ouro Preto
Novembro de 2019

P436a Pereira, Ítalo Magno.
Uma abordagem para estimar a similaridade item-item baseada nos relacionamentos semânticos da Linked Open Data [manuscrito] / Ítalo Magno Pereira. - 2019.
60f.: il.: color; grafs; tabs.

Orientador: Prof. Dr. Anderson Almeida Ferreira.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Dados vinculados. 2. Ontologia. 3. Correlação. I. Ferreira, Anderson Almeida. II. Universidade Federal de Ouro Preto. III. Titulo.

CDU: 004.62

Catalogação: www.sisbin.ufop.br

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ATA DE DEFESA DE DISSERTAÇÃO

Aos 24 dias do mês de outubro do ano de 2019, às 10:00 horas, nas dependências do Departamento de Computação (Decom), foi instalada a sessão pública para a defesa de dissertação do mestrando **Ítalo Magno Pereira**, sendo a banca examinadora composta pelo Prof. Dr. Anderson Almeida Ferreira (Presidente - UFOP), pelo Prof. Dr. Alvaro Rodrigues Pereira Junior (Membro - UFOP), pela Profa. Dra. Lívia Couto Ruback Rodrigues (Membro - Externo). Dando início aos trabalhos, o presidente, com base no regulamento do curso e nas normas que regem as sessões de defesa de dissertação, concedeu ao mestrando 60 minutos para apresentação do seu trabalho intitulado "Uma Abordagem para Estimar a Similaridade Item-Item Baseada nos Relacionamentos Semânticos da Linked Open Data". Terminada a exposição, o presidente da banca examinadora concedeu, a cada membro, um tempo máximo de 60 minutos para perguntas e respostas ao candidato sobre o conteúdo da dissertação, na seguinte ordem: Primeiro, Profa. Lívia Couto Ruback Rodrigues; segundo, Prof. Alvaro Rodrigues Pereira Junior; terceiro, Prof. Anderson Almeida Ferreira. Dando continuidade, ainda de acordo com as normas que regem a sessão, o presidente solicitou aos presentes que se retirassem do recinto para que a banca examinadora procedesse à análise e decisão, anunciando, a seguir, publicamente, que o mestrando foi aprovado por unanimidade, sob a condição de que a versão definitiva da dissertação deva incorporar todas as exigências da banca, devendo o exemplar final ser entregue no prazo máximo de 60 (sessenta) dias à Coordenação do Programa. Para constar, foi lavrada a presente ata que, após aprovada, vai assinada pelos membros da banca examinadora e pelo mestrando. Ouro Preto, 24 de outubro de 2019.

Prof. Dr. Anderson Almeida Ferreira
Presidente

Prof. Dr. Alvaro Rodrigues Pereira Junior

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Profa. Dra. Lívia Couto Ruback Rodrigues
(Participação por Videoconferência)

Mestrando

Certifico que a defesa realizou-se com a participação a distância do(s) membros(s) Profa. Dra. Lívia Couto Ruback Rodrigues e que, depois das arguições e deliberações realizadas, cada participante a distância afirmou estar de acordo com o conteúdo do parecer da banca examinadora, redigido nesta ata.

Prof. Dr. Anderson Almeida Ferreira
Presidente

Resumo da Dissertação apresentada à UFOP como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciência da Computação (M.Sc.)

UMA ABORDAGEM PARA ESTIMAR A SIMILARIDADE ITEM-ITEM BASEADA NOS RELACIONAMENTOS SEMÂNTICOS DA LINKED OPEN DATA

Italo Magno Pereira

Novembro/2019

Orientador: Anderson Almeida Ferreira

Programa: Ciência da Computação

A época atual está sendo vista como uma era de sobrecarga de informação, uma vez que mais dados são produzidos do que humanos podem processar. Este fato implica na melhoria constante de sistemas de recuperação e tratamento de informação. Inserido neste contexto, os sistemas de recomendação são ferramentas importantes aos usuários, por sugerir itens que possam ser interessantes. No entanto, os sistemas de recomendação baseados em filtragem colaborativa sofrem com o problema conhecido como *cold start* ou falta de dados iniciais. A opção para contornar esse problema é explorar outras fontes de dados, como a *Linked Open Data* (LOD), para enriquecer os dados. Contudo, muitas soluções baseadas na LOD não fazem uso dos relacionamentos semânticos e, quando o fazem, não ponderam corretamente seus relacionamentos e, assim, não exploram o seu potencial. Este trabalho visa apresentar uma abordagem para explorar os relacionamentos semânticos da *Linked Open Data*, por meio da descoberta de características relevantes e ponderação de tais características sem intervenção de um especialista de domínio de aplicação. Para avaliar a proposta, foram realizados experimentos em dois domínios de aplicação, domínio de filmes e museus. Os resultados mostraram-se competitivos comparados a outras abordagens, onde a seleção de propriedades relevantes é feita manualmente.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Devido à grande popularidade da Web, das redes sociais e à rápida propagação dos dados publicados na Web 2.0, estamos vivendo em uma era de *sobrecarga de informação*. As pessoas publicam mais dados do que podem processar e consumir. Este fato tem resultado no aumento de esforços para desenvolver sistemas capazes de lidar com esta situação. Tais sistemas são categorizados como Sistemas de Recuperação e Tratamento da Informação (Colucci et al., 2016, Di Noia et al., 2012, Di Noia and Ostuni, 2015, Leng et al., 2018, Mirizzi et al., 2012, Srinivasan and Mani, 2018).

Dentre os sistemas de recuperação e tratamento da informação podemos destacar os Sistemas de Recomendação (SR), cujo propósito é sugerir itens que sejam de interesse do usuário (Di Noia et al., 2012, Ricci et al., 2015, Srinivasan and Mani, 2018). Uma característica esperada em SR, é ativamente alertar os usuários sobre informações que sejam de seu interesse, auxiliando no processo de tomada de decisão (Alfarhood, 2017).

Os SR podem ser divididos em duas categorias, conforme suas características: SR que realizam a predição de itens similares a um item, que um usuário tenha gostado, comuns em serviços de *e-commerce* ou *Video-on-demand*; ou SR que realizam a predição de itens com base no perfil dos usuários, usando as preferências de um usuário ou as preferências de usuários com perfis similares (Colucci et al., 2016, Leng et al., 2018).

O interesse em SR cresceu de forma significativa, tendo como fator motivador, o fato de ter um papel importante em grandes empresas de serviços *online* como Amazon, YouTube, Netflix, Spotify, LinkedIn, Facebook, Tripadvisor, Last.fm, IMDb, entre outros (Ricci et al., 2015). Esses sistemas têm sido adotados e oferecidos aos usuários como parte dos serviços dessas empresas, estão disponíveis em diversas plataformas *online*, seja comércio eletrônico, notícias ou entretenimento (Alfarhood, 2017).

Muitos SR usam abordagens estatísticas, como, por exemplo, filtragem colabo-

rativa baseada nas escolhas de usuários para realizar as recomendações. Contudo, essas abordagens ainda encontram problemas em realizar a predição acurada de itens, devido ao problema conhecido como *Cold Start*. Este problema é caracterizado pela falta de informação para realizar a recomendação em seus estágios iniciais, seja pela recência de itens ou de usuários no sistema (Leal et al., 2012, Srinivasan and Mani, 2018). Para exemplificar a situação, pode-se tomar como base uma loja de livros. Para recomendar um determinado livro a um usuário é utilizada como base a avaliação de outros usuários que compraram o livro e tenham um perfil similar.

Para superar esta limitação, muitos dos SR baseados em conteúdo, considerados estado da arte, fazem uso de coleta de dados em diferentes fontes de dados, com o objetivo de identificar padrões para o desenvolvimento de modelos e realização de recomendações (Di Noia et al., 2012, Srinivasan and Mani, 2018). Baseado nesta hipótese a exploração de dados na *Linked Open Data* (LOD) pode ser considerada uma proposta bastante promissora.

A LOD é fruto do projeto *The Linking Open Data*, suportado pela W3C *Semantic Web Education*, que tem como objetivo a adoção e aplicação dos princípios da *Linked Data*. Tem como propósito inicializar e manter a Web de Dados, por meio da identificação de conjuntos de dados existentes, disponíveis sob licenças abertas, publicando-os na Web (Berners-Lee, 2006, Bizer et al., 2009). Os dados da LOD são representados por meio do padrão *Resource Description Framework* (RDF), sendo descritos por um conjunto de triplas compostas por sujeito, predicado (ou propriedade) e objeto. Esse padrão pode ser representado em forma de grafos, chamados grafos RDF (Berners-Lee, 2006, Leal et al., 2012).

Algumas propostas de SR fazem uso da LOD, mas muitas destas não usam os relacionamentos semânticos disponíveis entre as instâncias. E, quando usam, os relacionamentos (representados por meio de propriedades) mais relevantes e as respectivas ponderações são definidas de forma manual, em função da análise feita pelo usuário sobre cada domínio da informação.

Considerando grafos RDF, uma tripla contém um sujeito (nodo rotulado por uma URI, que identifica o sujeito), predicado (aresta rotulada por uma URI) e objeto (nodo que pode ter como rótulo uma URI ou um literal). Relacionamentos semânticos podem ser entendidos como os predicados (ou propriedades) que ligam os sujeitos aos objetos. Os sujeitos representam as instâncias, o conjunto de predicados representam os metadados das instâncias, e os objetos representam os valores dos dados propriamente ditos (Berners-Lee, 2006, Bizer et al., 2009, Yu, 2011).

Abordagens, cujas propriedades são definidas de forma manual, são dependentes de especialistas de domínio, sempre que for necessário estimar a similaridade entre itens em um novo domínio. Por exemplo, considerando o trabalho desenvolvido em (Ruback et al., 2017), cujo objetivo é verificar a similaridade de diferentes museus

e gerar uma lista de museus similares. Nesse trabalho, foi feita uma análise sobre o domínio da informação pelos autores, e constatado que as características mais importantes são "obras de arte" e "movimentos de arte". Contudo, essas características provavelmente não serão relevantes em outro domínio como, por exemplo, músicas, livros ou filmes. Por este motivo, pesquisar e propor formas de recomendar itens com base nas propriedades semânticas da LOD, pode responder a perguntas como:

- Como descobrir automaticamente propriedades relevantes a um domínio de aplicação, por meio da LOD, para verificar a similaridade entre itens?
- Como mensurar a relevância de cada propriedade, ponderando-as em relação às demais?
- Após a seleção das propriedades relevantes, como comparar as instâncias (itens) usando suas diversas propriedades?

1.1 Objetivos

Para responder a estas perguntas, este trabalho tem como objetivo geral, propor uma abordagem para explorar os relacionamentos semânticos entre as instâncias na LOD e estimar a similaridade entre elas. Ou seja, utilizar a LOD como fonte de dados, para auxiliar o processo de sugestão de itens em SR baseados em conteúdo. A abordagem seleciona e pondera automaticamente as propriedades relevantes das instâncias de um domínio de informação e estima a similaridade entre os itens (instâncias desse domínio) através de seus relacionamentos semânticos.

Objetivos Específicos

Este trabalho tem como objetivos específicos os seguintes itens:

- Pesquisar e propor formas de ponderar, de maneira automática, os predicados de um domínio de aplicação, de forma a indicar a relevância de cada um.
- Estimar a similaridade entre itens da LOD com base nas ponderações dos predicados, de forma eficiente e independente do domínio;
- Utilizar e avaliar métricas de similaridade, que melhor adéquem ao contexto da informação a ser comparada, com o intuito de gerar listas ordenadas de itens como recomendação;
- Pesquisar ou construir um *Ground Truth*, para verificar a eficácia da abordagem, com base em métricas adequadas e também comparar com trabalhos relacionados.

1.2 Justificativa

Os SRs são sistemas de filtragem e, por este motivo, tem tido grande destaque entre os Sistemas de Recuperação e Tratamento de Informação, devido a possibilidade de assistir usuários em suas escolhas, no atual cenário de sobrecarga de informações (Di Noia et al., 2012, Di Noia and Ostuni, 2015, Mirizzi et al., 2012, Srinivasan and Mani, 2018). E pelo seu grande impacto econômico tem tido também, grande influência em plataformas online de vários seguimentos, como comércio, notícias e entrenimento (Colucci et al., 2016). Existem diversos trabalhos neste campo que tentam melhorar diferentes aspectos desses sistemas, como, por exemplo, a acurácia (Alfarhood, 2017).

Um fator limitante a muitos sistemas de recomendação atuais é falta de dados em seu lançamento, *cold start*. Nesse contexto o uso da LOD surge como uma fonte valiosa de dados com potencial de mitigar tal limitação (Di Noia et al., 2012, Leng et al., 2018, Srinivasan and Mani, 2018). O projeto LOD surgiu em 2007 e nos últimos anos possibilitou a criação e publicação de vários conjuntos de dados, resultando em uma grande base de conhecimento descentralizada, sendo considerada além de um conjunto de documentos e dados interligados um subconjunto da *World Wide Web* ou Web de Dados (Di Noia et al., 2016, Ruback et al., 2017).

O potencial da LOD pode ser validado com base na sua principal fonte de dados, a Wikipedia[1], via DBPedia[2]. A Wikipedia possui mais de dois milhões de artigos e milhares de colaboradores e é considerada a maior enciclopédia existente (Milne and Witten, 2008). Além disso, a LOD emgloba atualmente mais de 1200 conjuntos de dados, nas mais diversas áreas do conhecimento[3], estimativa de 29 de março de 2019.

Outro importante aspecto a ser considerado em SR baseados em conteúdo, é a falta por parte de vários desses sistemas, de informações semânticas suficientes sobre os itens, e informações semânticas sobre os relacionamentos entre os itens, de modo que os itens relacionados possam ser identificados e recomendados com precisão (Alfarhood, 2017).

1.3 Contribuições do Trabalho

Esta Seção apresenta as principais contribuições com o desenvolvimento deste trabalho.

- Proposta de uma abordagem para explorar os relacionamentos semânticos en-

[1] http://www.wikipedia.org/
[2] https://wiki.dbpedia.org/
[3] https://lod-cloud.net/

tre as instâncias da LOD, através da descoberta de características relevantes (predicados), e estimar a similaridade entre elas.

- Realização de experimentos para validação da abordagem proposta.
- Publicação de um artigo intitulado "*An Item-Item Similarity Approach based on Linked Open Data Semantic Relationship*" no Simpósio Brasileiro de Sistemas Multimídia e Web em 2019.

1.4 Organização da dissertação

O restante desta dissertação está organizado da seguinte maneira.

No capítulo 2, são apresentados conceitos essenciais que servem como base para o entendimento do trabalho além de apresentar trabalhos significativos na literatura relacionados com a proposta deste documento.

No capítulo 3, é descrita a proposta implementada para tratar o problema de estimar a similaridade de itens com base em conjuntos de dados da LOD.

No capítulo 4, são apresentados os resultados experimentais para validação da proposta com base nas *baselines* selecionadas.

No capítulo 5, são apresentadas as conclusões do trabalho, limitações e trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Este capítulo tem como objetivo apresentar assuntos relevantes relacionados ao trabalho, com intuito de contextualizar o leitor e também expor trabalhos relacionados. A Seção 2.1 apresenta a fundamentação teórica com importantes tópicos, tais como: Sistemas de Recomendação, tópico relevante por estar relacionado ao objetivo geral desse trabalho; *Linked Open Data*, relevante pelo fato de ser a fonte de dados utilizada neste trabalho; Métricas de Similaridade, relevante pela necessidade de uso para avaliação de similaridade dos itens. Na Seção 2.2 são apresentados os principais trabalhos relacionados.

2.1 Fundamentação Teórica

A Seção 2.1.1 aborda as principais definições relacionadas a SR, definição do problema de recomendação, aplicabilidade e técnicas. Na Seção 2.1.2, são discutidas definições relacionadas a LOD, são expostos os princípios da *Linked Data* definidos por Tim Berners-Lee[1]. Além disso, apresenta como os dados da LOD são representados e também como realizar sua manipulação. Na Seção 2.1.3, são discutidas as principais métricas empregadas na avaliação da acurácia de recomendações top-N, foco deste trabalho.

2.1.1 Sistemas de Recomendação

Nesta Seção, serão explicitadas as principais definições relacionadas a SR.

SRs são definidos como ferramentas de software e técnicas que sugerem itens potencialmente úteis a usuários (Di Noia and Ostuni, 2015, Figueroa et al., 2015). Tais sistemas tem como objetivo realizar a sugestão de itens, cuja probabilidade de interesse seja maior para um determinado tipo de usuário, consequentemente

[1] https://en.wikipedia.org/wiki/Tim_Berners-Lee

melhorando sua experiência. Para tal, são utilizadas técnicas que possibilitem a seleção e apresentação desses itens (Ricci et al., 2015).

Usuários, no contexto de SR, são definidos como atores que recebem as recomendações. Um importante aspecto a ser considerado são suas preferências, estas devem ser mantidas, pois serão utilizadas como insumo para que o SR possa selecionar e fornecer recomendações personalizadas. A representação dos usuários pode ser realizada de diferentes maneiras em função da técnica de recomendação empregada (Di Noia and Ostuni, 2015).

Item em SR é um termo genérico, que denota o domínio da recomendação, como, por exemplo, livros, filmes, entre outros (Ricci et al., 2015). Segundo Di Noia and Ostuni (2015), os itens podem ser caracterizados por sua complexidade e também por sua utilidade. Como exemplos de itens de baixa complexidade e baixa utilidade podem ser considerados: livros, páginas da web e filmes. Em contraposição há itens de maior complexidade e maior utilidade ou valor agregado. Como exemplos podem ser considerados: equipamentos como telefones, computadores, entre outros, ou ainda, serviços de oferta de empregos, viagens e serviços financeiros, entre outros.

*Avaliação (**Rating**)*, no contexto de SR, é definida como informações atualizadas sobre as preferências ou *feedback* dos usuários, e é utilizado pelo SR para geração das recomendações. A coleta dessa informação pode ser classificada como explícita ou implícita.

Na coleta explícita, as informações são fornecidas pelos usuários através da classificação de suas experiências sobre os itens, em função de uma escala. Essa escala pode ser binária (gosto / não gosto), numérica (por exemplo, de 1 a 5 estrelas) ou ordinal (concordo totalmente, concordo, neutro, discordo, discordo totalmente). A coleta implícita de informação dos usuários é mais utilizada na prática pelos SR, apesar das informações coletadas de forma explícita serem mais comuns na literatura. Na coleta implícita de informação, os sistemas fazem inferência sobre as preferências dos usuários com base em seu comportamento sem causar perturbação (Di Noia and Ostuni, 2015).

A Figura 2.1 apresenta um exemplo de uma matriz de *Ratings* de usuários em função dos itens (filmes), na qual os usuários demonstram suas preferências através de uma nota.

Definição do Problema de Recomendação

Uma formulação formal do problema de recomendação, com base nos *ratings* de usuários, é descrita em (Di Noia and Ostuni, 2015). Seja U um conjunto de usuários e I um conjunto de itens, ambos os conjuntos podem ser muito grandes. Seja $f : U \times I \rightarrow R$, onde R é um conjunto totalmente ordenado, seja uma função de utilidade que mede a utilidade do item $i \in I$ para o usuário $u \in U$. Então,

Figura 2.1: Exemplo de Matriz de *rating* usuário-item com filmes.
Fonte: Di Noia and Ostuni (2015).

o problema de recomendação consiste em encontrar para cada usuário esse item $i^{max,u} \in I$ maximizando a função de utilidade f. Mais formalmente, isso corresponde ao seguinte:

$$\forall u \in U, \;\; i^{max,u} = arg \;\; max_{i \in I} \;\; f(u,i)$$

Normalmente, a utilidade de um item é representada por uma classificação, que indica como um usuário em particular gostou de um item específico. O problema central dos sistemas de recomendação é que a utilidade não é definida em todo o espaço $U \times I$, pois apenas um subconjunto dele está realmente disponível. Para cada usuário, apenas uma parte de seus valores é conhecida. Portanto, a principal tarefa do sistema diz respeito à estimativa da função de utilidade a partir dos dados disponíveis (Di Noia and Ostuni, 2015).

Aplicabilidade de Sistemas de Recomendação

A definição de SR pode ser refinada em função dos possíveis papéis desempenhados por este, a percepção dos benefícios de um SR é diferente, conforme o tipo de usuário, seja do ponto de vista do prestador de serviço ou do ponto de vista do tomador do serviço. Um prestador de serviço, como Expedia.com, faz uso de um SR visando aumentar o volume de seus negócios, através do aumento das vendas de vagas em quartos em hotéis. Já um tomador de serviço, acessa um sistema dessa natureza motivado em encontrar um hotel adequado ao seu gosto, e eventos oferecidos no seu destino. É possível notar uma clara distinção entre os diferentes papéis desempenhados pelo SR (Ricci et al., 2015).

Dentre as motivações para os prestadores de serviços, podem ser citadas: aumento na quantidade de itens vendidos; aumento na diversidade dos itens vendidos; aumento da satisfação dos usuários, o que consequentemente, pode levar ao au-

mento da fidelidade; melhor entendimento do que os usuários realmente querem (Ricci et al., 2015).

Dentre os benefícios dos SR aos tomadores de serviços, destacam-se: possibilidade de encontrar bons itens ou itens com maior potencial de satisfação; recomendação de sequências de itens, dado um item pode ser feita a seleção de itens considerados sequenciais; recomendação de pacotes, recomendação de itens afins ou itens que são ideais quando juntos; possibilidade de melhorar o próprio perfil perante o SR, através de informações de *feedback*, possibilitando uma recomendação mais precisa (Ricci et al., 2015).

Técnicas

Os SR podem ser classificados conforme as seguintes classes de abordagens (Burke, 2007, Ricci et al., 2015):

Filtragem colaborativa: Os SR baseados nesta abordagem realizam recomendações a usuários, com base em itens que outros usuários, com perfis similares, avaliaram de forma positiva anteriormente. Os perfis representam as preferências ou gostos. A similaridade entre dois usuários, é obtida por meio da verificação de similaridade do histórico de classificação de itens realizada previamente por eles (Ricci et al., 2015).

Esta técnica se beneficia do conhecimento adquirido por um conjunto de usuários, para realizar novas recomendações, ou seja, o sistema utiliza informação colaborativa do padrão de comportamento de um grupo de usuários, para recomendar um item a um outro usuário, conforme o perfil (Santos, 2017). Os SR colaborativos buscam usuários pares de um determinado usuário, com base no histórico de classificação similar para recomendar itens (Burke, 2007).

Os SR colaborativos podem realizar as recomendações com base nas características dos usuários, ou com base nas características de itens. A escolha da forma como será feita a recomendação, é decisiva no processo de classificação ou no desempenho computacional do sistema (Ricci et al., 2015).

A Figura 2.2 mostra um exemplo de uso da filtragem colaborativa correspondente a matriz de similaridade representada na Figura 2.1. Nesse exemplo, a usuária Alice tem um perfil similar aos usuários Tom e John, com isso as informações de classificações realizadas por eles podem ser usadas para recomendar novos itens a Alice.

Baseado em conteúdo: Os SR baseados em conteúdo realizam as recomendações de itens similares a itens avaliados pelo usuário no passado (Santos, 2017). O cálculo de similaridade é realizado sobre as características dos itens em comparação. Por exemplo, se um usuário gostou de um filme cujo gênero é comédia, esta característica será usada pelo SR para recomendar outros filmes com características similares (Ricci

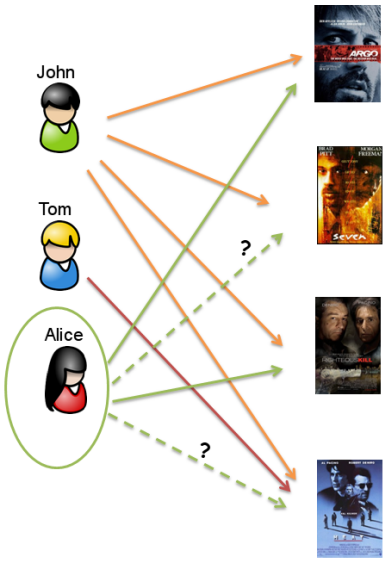


Figura 2.2: Exemplo de SR baseado em Filtragem Colaborativa.
Fonte: Di Noia and Ostuni (2015).

et al., 2015).

SR que seguem esta abordagem realizam a recomendação a partir de duas fontes, sendo a primeira as características associadas aos itens e a segunda a classificação dada ao item pelos usuários. Além disso, SR baseados em conteúdo tratam a recomendação como um problema de classificação, específico de um usuário, e aprendem um classificador para as preferências e rejeições do usuário com base nas características do item (Burke, 2007).

A Figura 2.3 mostra um exemplo de abordagem de recomendação baseada em conteúdo. Neste exemplo, os itens, filmes, são apresentados com atributos, como atores e gêneros. A intuição básica com essa abordagem é que, como Alice gosta do filme *Argo*, ela pode gostar também do filme *Heat*, porque ambos pertencem ao gênero Drama.

Figura 2.3: Exemplo de SR baseado em Conteúdo.
Fonte: Di Noia and Ostuni (2015).

Demográfica: Os SR baseados na abordagem demográfica, fazem uso do perfil demográfico do usuário para realizar as recomendações (Burke, 2007, Ricci et al., 2015). Esta abordagem é justificada pelo princípio de que, indivíduos que tenham determinados atributos pessoais em comum, também compartilham determinadas preferências, estes atributos podem ser, por exemplo: sexo, idade, idioma, região geográfica, grupo social entre outros (Santos, 2017).

Baseada em conhecimento: Os SRs baseados em conhecimento realizam as recomendações com base em inferências e necessidades dos usuários (Burke, 2007). SR baseados nesta abordagem, recomendam itens de domínios de conhecimento específicos, sobre como as características deste itens irão atender as preferências dos usuários, e como poderão ser úteis a ele. Nestes sistemas, a similaridade pode ser interpretada como a utilidade da recomendação para o usuário (Ricci et al., 2015) .

Um detalhe importante sobre SR baseados em conteúdo, é o fato de tenderem a ter melhor desempenho nas fases iniciais. Contudo caso não sejam dotados de sistemas de aprendizagem, podem ser ultrapassados por outros sistemas, como os baseados em filtragem colaborativa, que fazem uso de registros de interações entre homem e computador (Ricci et al., 2015).

Híbrida: São classificados como híbridos quaisquer SRs, que utilizem múltiplas técnicas, com objetivo de realizar recomendações (Burke, 2007). SRs híbridos tem como objetivo utilizar as vantagens de uma técnica para cobrir as deficiências de outra técnica, como por exemplo, um SR baseado na abordagem de filtragem colaborativa, cuja desvantagem é não poder realizar recomendações de itens a usuários sem histórico (Ricci et al., 2015, Santos, 2017).

2.1.2 *Linked Open Data*

Esta Seção tem como objetivo apresentar as definições relativas a *Linked Open Data* e tecnologias relacionadas.

A **Web** é um conjunto de documentos não estruturados e interligados, projetada para ser utilizada por humanos. Para tal deve ser possível sua leitura e entendimento por pessoas (*human-readable*) (Leal et al., 2012). Contudo, o volume de informação produzido dentro do contexto da Web está além do que as pessoas podem consumir e processar, o que é descrito como uma era de sobrecarga de informação. Desta maneira, os esforços da iniciativa da *Linked Data* para criação de uma Web de Dados, também referida como Web Semântica, que torne possível a leitura dos dados por máquinas, é justificável (Di Noia et al., 2012).

Linked Data pode ser entendida, de maneira simplificada, como a utilização da Web para criação de ligações tipadas entre dados de diferentes fontes. De forma mais técnica, pode ser entendida, como dados da Web publicados de uma maneira

que seja possível a leitura por máquinas (*machine-readable*) (Bizer et al., 2009). A *Linked Data* pode ser referida também como o conjunto das melhores práticas, para publicação e interconexão de dados estruturados da Web. Estas melhores práticas foram introduzidas por Berners-Lee (2006) e se tornaram conhecidas como Princípios da *Linked Data* (Heath and Bizer, 2011).

Linked Open Data (LOD) é fruto do projeto *The Linking Open Data*, suportado pela W3C *Semantic Web Education*. A LOD visa a adoção e aplicação dos princípios da *Linked Data*, com o objetivo de inicializar e manter a Web de Dados. São identificados conjuntos de dados disponíveis sob licenças abertas e publicada na Web (Bizer et al., 2009).

A LOD, bem como seus padrões e formatos, é o resultado natural proveniente da necessidade de padronização, que possibilita a distribuição e consumo de dados, de forma estruturada, oriundos de diferentes fontes (Alfarhood, 2017).

A Figura 2.4 mostra os conjuntos de dados presentes na LOD em seu surgimento em maio de 2007. No início eram apenas doze conjuntos de dados interligados. O mapa dos dados, com a situação em Abril de 2018, pode ser visualizado na Figura 2.5. Este mapa conta com 1184 conjuntos de dados, de diversos domínios. Podendo ser domínios específicos como livros, músicas ou filmes, ou domínios genéricos, com as mais diversas informações, como, por exemplo, o conjundo de dados da DBpedia.

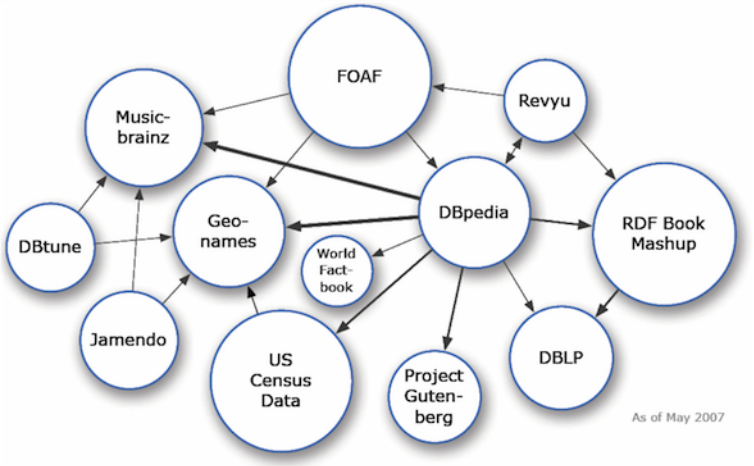


Figura 2.4: Linked Open Data *Cloud* em Maio de 2007
Fonte: McCrae (2018)

O mapa da LOD é mantido por McCrae (2018) e pode ser encontrado no site http://lod-cloud.net/. Nesta página podem ser encontradas diversas informações referentes a LOD. A Tabela 2.1 contém informações referentes a evolução dos conjuntos de dados presentes na LOD ao longo dos anos. Com a evolução na quantidade de conjuntos também cresceu a quantidade de dados.

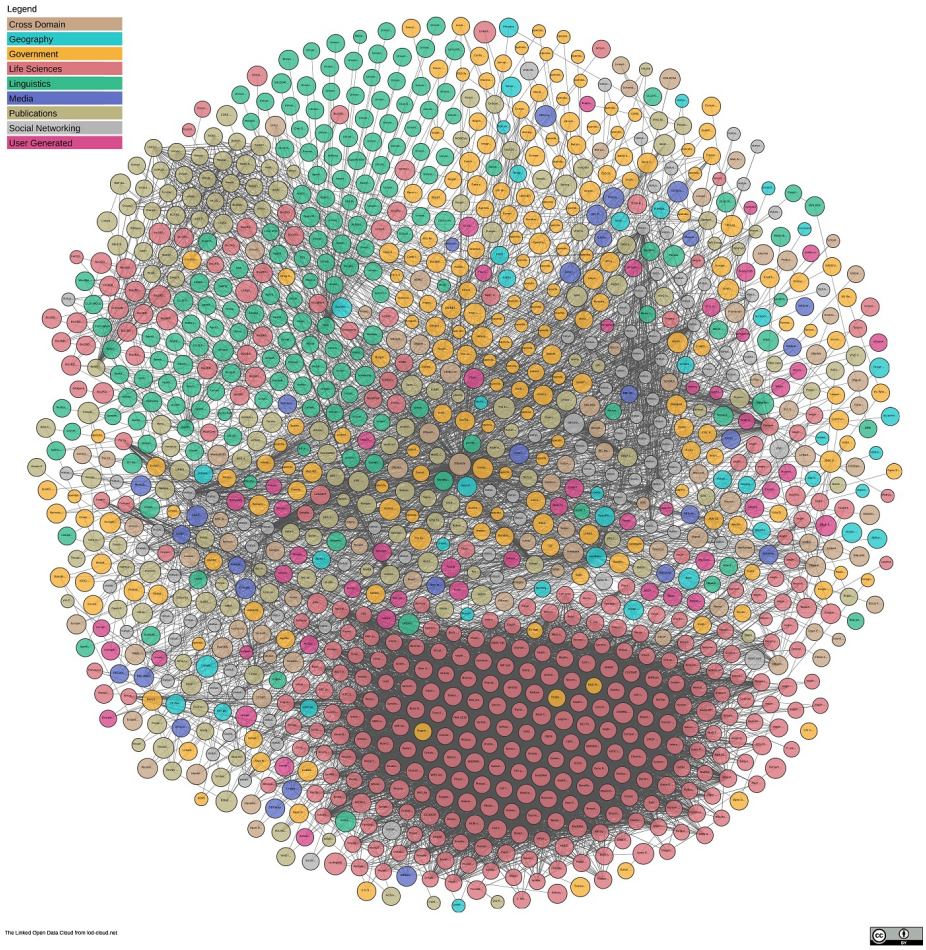


Figura 2.5: Linked Open Data *Cloud* em Abril de 2018
Fonte: McCrae (2018)

Princípios da *Linked Data*

De acordo com Berners-Lee (2006), a *Linked Data* tem como princípios os seguintes itens:

1. Use URIs para identificar coisas
2. Use HTTP URIs para que as pessoas possam procurar esses nomes.
3. Quando alguém procura um URI, forneça informações úteis, por meio de padrões.
4. Inclua links para outros URIs, para que eles possam descobrir mais coisas.

Esses princípios fornecem uma forma de orientação no processo de publicação e conexão de dados, na forma da Web de dados. Diversas organizações, seguindo esta tendência, tem publicado seus dados seguindo os padrões da LOD, e interconectando seus dados a outros conjuntos relacionados (Alfarhood, 2017).

Tabela 2.1: Evolução da LOD

Data	Datasets
2018-04-30	1,184
2017-08-22	1,163
2017-02-20	1,139
2017-01-26	1,146
2014-08-30	570
2011-09-19	295
2010-09-22	203
2009-07-14	95
2009-03-27	93
2009-03-05	89
2008-09-18	45
2008-03-31	34
2008-02-28	32
2007-11-10	28
2007-11-07	28
2007-10-08	25
2007-05-01	12

Representação dos Dados da LOD

Resource Description Framework (RDF) é um modelo abstrato utilizado para representar os dados da LOD de acordo com os princípios da *Linked Data*. O formato RDF é uma especificação que inclui um modelo de dados, normalmente representado em XML, que armazena um conjunto de triplas compostas por sujeito, predicado e objeto, e que pode ser visto como um grafo direcionado rotulado (Leal et al., 2012).

A *Linked Data* através do RDF permite realizar a vinculação de coisas do mundo real por meio de de links, fazendo com que a Web de dados possa ser definida como uma rede de coisas do mundo, descritas por dados na Web (Bizer et al., 2009).

Uniform Resource Identifiers **(URI's)**, diferentemente de *Uniform Resouce Locators* (URL's), têm um significado mais genérico para identificar qualquer entidade do mundo real, enquanto URL's são utilizados como endereços para documentos e outras entidades, que podem ser localizadas na Web (Bizer et al., 2009).

O relacionamento explicitado na Figura 2.6 ocorre entre diferentes URI's. No exemplo são identificadas três pessoas:

- http://biglinx.co.uk/people/matt-briggs
- http://biglinx.co.uk/people/scott-miller
- http://biglinx.co.uk/people/linda-meyer

Cada URI neste exemplo define uma pessoa do mundo real. Para simplificação sejam estas pessoas referenciadas por A, B e C, respectivamente. A pessoa A conhece

a pessoa B. Neste caso a pessoa A é o sujeito e a pessoa B o objeto, a ligação entre as duas pessoas se dá pelo predicado ***knows*** (conhece). A pessoa B no relacionamento anterior é um objeto mas também é representada por uma URI, ou seja, um recurso, e, como tal, pode ser um sujeito em um outro relacionamento. Como exemplo, a pessoa B conhece a pessoa C. Neste relacionamento ela é um sujeito e a pessoa C um objeto. É importante ressaltar que os grafos RDF são direcionados, logo se pessoa A conhece B e B conhece A, necessariamente deve haver duas arestas entre os nodos. Neste caso, em uma das arestas, A será um sujeito e B um objeto, e na outra aresta, B será o sujeito e A o objeto.

O nodo sujeito sempre será um recurso e, por isso, sempre será determinado por uma URI, contundo objetos podem ser recursos ou podem ser literais. Como exemplo, o nome de uma pessoa é um literal, sendo "Matt Briggs"e o predicado ***name*** identifica a propriedade deste literal. Predicados também são identificados por URI's.

Figura 2.6: Uso de URIs para identificação de pessoas e relacionamento entre elas. Fonte: Heath and Bizer (2011).

Manipulação dos Dados da LOD

SPARQL, acrónimo para *SPARQL Protocol and RDF Query Language*, é uma linguagem de consulta e também um protocolo, utilizado para realizar consultas e manipulação de dados armazenados no formato RDF em um *dataset* (Alfarhood, 2017).

SPARQL provê quatro diferentes formas de consulta: *SELECT query, ASK query, DESCRIBE query, CONSTRUCT query.* O SELECT é a forma de consulta mais utilizada. Todas as formas de consulta são baseadas em dois conceitos SPARQL, Triple Pattern e Graph Pattern (Yu, 2011).

Triple Pattern: O modelo RDF é construído sobre o conceito de triplas. Constituídas por sujeito, predicado e objeto (RDF Triple), por sua vez o SPARQL é construído sobre conceito de *Tripple Pattern*, também descrito como sujeito, predicado e objeto (SPARQL *Triple Pattern*). A diferença entre RDF Triple e SPARQL *Tripple Pattern*, está no fato de que o segundo, pode incluir variáveis em sua construção, qualquer sujeito, predicado ou objeto pode ser substituído por variáveis.

Exemplo extraído de (Yu, 2011):

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
<http://danbri.org/foaf.rdf#danbri> foaf:name ?name.
```

O RDF usa URIs em vez de palavras para nomear recursos e propriedades. Um conjunto destas URIs (geralmente criados para uma finalidade específica) são referidos como um vocabulário RDF. Além disso, todos os URIs desse vocabulário normalmente compartilham uma sequência principal comum, usada como prefixo comum. Esse prefixo geralmente se tornará o prefixo de espaço para nome para este vocabulário, e os URIs nesse vocabulário serão formados anexando nomes locais individuais ao final dessa cadeia principal comum (Yu, 2011). A primeira linha do exemplo acima apresenta um exemplo de prefixo, *alias*, de um *namespace* em um vocabulário. A segunda linha utiliza esse *alias* para compor uma URI, nesse caso foaf:name.

A segunda linha do texto acima é um SPARQL Triple Pattern. Nele é possível perceber que o sujeito é uma URI para Dan Brickley, o predicado é um foaf:name e o objeto é uma variável identificada pelo caracter ? seguida pelo seu nome.

Graph Pattern: O Graph Pattern é similar ao Triple Pattern e também é utilizado para selecionar triplas em um grafo RDF, contudo podem ser especificadas regras de seleção muito mais complexas, tomando como base um simples *Triple Pattern*.

Exemplo extraído de (Yu, 2011):

```
{
    ?who foaf:name ?name.
    ?who foaf:interest ?interest.
    ?who foaf:knows ?others.
}
```

Uma coleção de Triple Patterns é denominada um *Graph Pattern* e é delimitado pelos caracteres { e }. Para entender como um *Graph Pattern* pode ser utilizado para selecionar triplas em um grafo RDF é preciso saber que, se uma variável está

presente em múltiplos *Triple Patterns*, em um *Graph Pattern*, seu valor em todos os padrões será o mesmo (Yu, 2011).

O texto na caixa abaixo contém um exemplo de uma consulta básica para encontrar todas as pessoas conhecidas por Dan Brickley (Yu, 2011):

```
base <http://danbri.org/foaf.rdf>
prefix foaf: <http://xmlns.com/foaf/0.1/>
select *
from <http://danbri.org/foaf.rdf>
where {
    <#danbri> foaf:knows ?friend.
}
```

2.1.3 Métricas de similaridade

A verificação de similaridade é utilizada em diferentes campos de pesquisa, seja por pesquisadores em Processamento de Linguagem Natural, pesquisadores em Resolução de Entidades ou pesquisadores em Bancos de Dados. A verificação de similaridade também é utilizada em problemas de descoberta de entidades semanticamente similares e pode ser empregada em SR baseados em conteúdo (Ruback et al., 2017).

Um aspecto comum na avaliação da qualidade das recomendações em experimentos é a acurácia e de acordo com a literatura há duas maneiras de medi-la, quando o contexto são recomendações. A primeira está relacionada a previsão de *ratings* e a segunda está relacionada a recomendações top-N (Di Noia and Ostuni, 2015).

Quando um sistema é usado pra prover recomendações top-N, muitos dos métodos de avaliação desenvolvidos no campo de Recuperação de Informação, podem ser empregados, devido à natureza de tais recomendações. Contudo, isto pode não ser considerado em sua totalidade, quando a avaliação é feita com base em *ratings* de usuários, por dois motivos: natureza e a disponibilidade de informações relevantes sobre itens. Cada usuário tem suas relevâncias individuais sobre itens, determinadas por seus *ratings*, e os SRs têm apenas uma pequena porção de informação, sobre as avaliações destes usuários.

As métricas mais comuns para avaliação de qualidade de predições de *ratings*, são métricas baseadas em erro, como *Root Mean Squared Error* (RMSE) e *Mean Absolute Error* (MAE). Entretanto, não serão discutidas aqui, devido ao foco deste trabalho estar relacionado a tarefa de recomendação top-N. Medidas mais apropriadas para avaliação de acurácia de recomendações top-N, são métricas orientadas a precisão, por considerar listas ordenadas de itens (*ranked lists*) (Di Noia and Ostuni, 2015). Neste contexto, serão discutidas as métricas *Precision*, *Recall*, *Normalized*

Discounted Cumulative Gain (NDCG). Serão discutidas também, as métricas de similaridade de Cosseno e *Term Frequency-Inverse Document Frequency* TF-IDF.

Precisão* e *Revocação

Precision@N para o usuário u ($P_u@N$) é calculada como a fração dos itens recomendados no top-N, exibidos no conjunto de teste do usuário e que são relevantes para o usuário (Di Noia and Ostuni, 2015).

$$P_u@N = \frac{|L_u(N) \bigcap TS_u^+|}{N} \tag{2.1}$$

TS_u^+ é o conjunto de itens de teste relevantes para u e $L_u(N)$ é a lista de itens recomendados até a posição N (Di Noia and Ostuni, 2015).

O Recall@N ($R_u@N$) é calculado como a proporção dos itens recomendados no top-N que aparecem no conjunto de testes do usuário, que também são relevantes para o número de itens relevantes no conjunto de testes do usuário (Di Noia and Ostuni, 2015).

$$R_u@N = \frac{|L_u(N) \bigcap TS_u^+|}{|TS_u^+|} \tag{2.2}$$

***Term Frequency-Inverse Document Frequency* e Cosseno**

Term Frequency (TF) representa o número de vezes que um termo ocorre em um documento em relação a quantidade de termos no documento. Como o próprio nome sugere é a frequência do termo e determina o peso do termo dentro do documento.

Inverse Document Frequency (IDF) visa contornar um problema crítico da métrica TF, pois, esta considera todos os termos com o mesmo peso, ou seja, são considerados igualmente importantes. Como exemplo, considere uma coleção de documentos de uma indústria automotiva, o termo auto aparece diversas vezes e conforme a métrica TF, é considerado importante. O IDF é um mecanismo que visa minimizar este impacto, através da seguinte ideia: um termo que seja comum em diversos documentos, deve ter sua importância minimizada, desta maneira o IDF pode atenuar o TF (Manning et al., 2008).

O IDF de um termo é o *log* da relação entre a quantidade total de documentos em uma coleção(N) e a quantidade de documentos na coleção que contém este termo(df_t), calculado conforme a Equação 2.3:

$$IDF_t = log\frac{N}{df_t} \tag{2.3}$$

O TF-IDF é uma métrica de peso composta para cada termo, resultado da combinação das métricas TF e IDF.

$$TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t \tag{2.4}$$

O cálculo da similaridade de cosseno, está relacionado ao conceito de *Vector Space Model*, no qual um vetor é derivado de cada documento. O conjunto de documentos em uma coleção é visto como um conjunto de vetores em um espaço vetorial, como exemplificado pela Figura 2.7. Cada termo terá seu próprio eixo. Usando a fórmula mostrada na Equação 2.5, a semelhança entre quaisquer dois documentos pode ser calculada Manning et al. (2008).

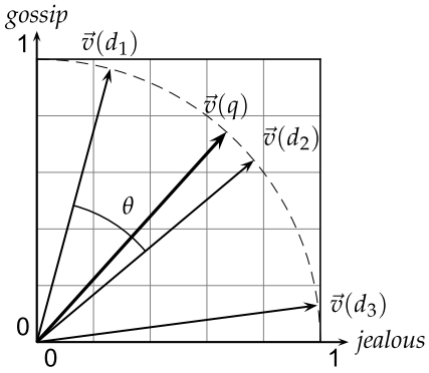


Figura 2.7: Similaridade do cosseno
Fonte: Manning et al. (2008)

$$sim(d_1, d_2) = \frac{\overrightarrow{V}(d_1).\overrightarrow{V}(d_2)}{|\overrightarrow{V}(d_1)||\overrightarrow{V}(d_2)|} \tag{2.5}$$

- $\overrightarrow{V}(d_1)$ é um vetor do documento d_1
- $\overrightarrow{V}(d_2)$ é um vetor do documento d_2
- $|\overrightarrow{V}(d_1)|$ é a norma de $\overrightarrow{V}(d_1)$
- $|\overrightarrow{V}(d_2)|$ é a norma de $\overrightarrow{V}(d_2)$

Normalized Discounted Cumulative Gain (NDCG)

Para entender melhor a métrica NDCG (Ganho Cumulativo com Desconto Normalizado), é necessário abordar previamente a métrica *Cumulated Gain* (CG) e a *Discounted Cumulated Gain* (DCG).

Na avaliação baseada em CG ou ganho de valor acumulado, a pontuação de relevância de cada documento é usado como uma medida de valor de ganho, de acordo com sua posição no *ranking* do resultado e o ganho é somado progressivamente da posição no *ranking* de 1 para N (Järvelin and Kekäläinen, 2002).

$$CG_p = \sum_{i=1}^{p} rel_i \tag{2.6}$$

Onde rel_i é a relevância do item na posição i.

Exemplo:

As listas abaixo representam, respectivamente, a lista com as pontuações dos itens, e a lista contendo os ganhos acumulados, em função da soma progressiva.

$$G_6 = [3, 2, 3, 0, 1, 2]$$
$$CG_6 = [3, 5, 8, 8, 9, 11]$$

O DCG está relacionado a ideia de que, quanto mais longe um item estiver, dentro do ranking, menor deverá ser o seu valor. Por este motivo, é aplicada uma função de desconto que, progressivamente reduz o valor do documento, com o aumento das posições no *ranking*. Uma maneira simples de realizar este decremento é dividir a pontuação de um item, pelo logaritmo de sua posição.

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.7}$$

Exemplo:

$$DCG_6 = [3, 4.262, 5.762, 5.762, 6.149, 6.861]$$

O processo de normalização ocorre através da seleção de todos os documentos mais relevantes, de maneira ordenada, dentro do *corpus*, produzindo o maior DCG possível até a posição p. Este é chamado de Ideal DCG (IDCG) e é calculado conforme a equação a seguir.

$$nDCG_p = \frac{DCG_p}{IDCG_p} = \frac{1}{IDCG_p} \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.8}$$

Exemplo:

Uma lista contendo as pontuações máximas com a seguinte representação $G_6 = [3, 3, 3, 2, 2, 2, 1, 0]$, teria um $IDCG_6 = 8.740$. Logo:

$$nDCG_6 = \frac{DCG_6}{IDCG_6} = \frac{6.861}{8.740} = 0.785$$

2.2 Trabalhos Relacionados

Esta Seção apresenta os trabalhos relacionados. Será apresentada uma breve descrição sobre estes trabalhos, explicitando suas contribuições e levantando questões

sobre pontos a melhorar. Todos os trabalhos a seguir fazem uso dos dados da LOD, com exceção dos trabalhos desenvolvidos em (Santos, 2017) e (Leng et al., 2018).

Em (**Ruback et al., 2017**), é proposto um framework denominado SELEcTor. Cujo objetivo é descobrir entidades similares na LOD a partir da exploração de listas ordenadas, *Ranked Lists* em inglês. Estas listas são extraídas diretamente da LOD e representam as entidades a serem comparadas. Segundo Ruback et al. (2017), foi realizada a adaptação do problema de comparação de similaridade entre duas entidades, para o problema de comparação de duas listas ordenadas. Como forma de resolver o problema desse trabalho foram elaboradas duas questões: Quais características das entidades são mais relevantes para calcular sua similaridade? Como medir a similaridade entre entidades com base em suas lista ordenadas? Uma das principais contribuições desse trabalho, é a proposta de um *framework* para extração de listas ordenadas de características de entidades, para posterior mapeamento ao problema de comparação de listas ordenadas. Como limitação, está a dependência de especialistas de domínio, para definir quais as características são mais relevantes sobre uma entidade em um domínio. Além disso, como deve ser feita sua extração para que seja possível a criação de uma lista ordenada.

Alfarhood (**2017**) explora propriedades relacionadas a forma de representação da LOD, feita através do uso de grafos, e propõe melhorias às abordagens atuais. Essas abordagens verificam o nível de similaridade, com base na distância semântica entre as entidades. Quanto maior o número de conexões entre dois recursos mais conectados eles serão. Este conceito é o núcleo de uma medida de relação de recursos, *Linked Data Semantic Distance* (LDSD), e também uma medida mais recente baseada nela, *Resource Similarity* (Resim). Estas abordagens analisam a conectividade entre dois recursos, se estão diretamente conectados ou indiretamente conectados, e calcula a distância semântica entre os dois recursos, ou seja, a medida de relação. Contudo, segundo Alfarhood (2017), estas abordagens têm como desvantagem o fato de tratar, todas as conexões entre recursos, da mesma forma. Então propõe melhorias a essas medidas que adotam ponderações entre as conexões. São propostas as medidas *Weighted Linked Data Semantic Distance* (WLDSD) e *Weighted Resource Similarity* (WResim). Além disso, propõe duas novas abordagens, que permitem o cálculo das ponderações entre as conexões *Resource-Specific Link Awareness Weights* (RSLAW) e *Information Theoretic Weights* (ITW).

Foi realizado também um estudo sobre o impacto do tipo das conexões, e foram propostas duas novas variações, que desconsideram estes tipos sobre as *baselines* *Typeless Linked Data Semantic Distance* (TLDSD) e *Typeless Resource Similarity* (TResim), além de uma análise sobre as variações ponderadas, *Typeless Weightet Linked Data Semantic Distance* (TLDSD) e *Typeless Weightet Resource Similarity* (WTResim). Esse trabalho tem como contribuição principal, a proposta de me-

lhorias em métricas existentes, poderando as conexões entre as entidades, além de propor meios de estimar essas ponderações.

Em (**Santos, 2017**), é proposta a aplicação do algoritmo de propagação de afinidades aos SRs. O algoritmo de propagação de afinidades ganhou destaque devido a sua aplicação em bioinformática, para problemas de agrupamento de sequências de DNA. Segundo Santos (2017), SR que fazem uso Filtragem Colaborativa baseada em agrupamento, têm maior escalabilidade e são utilizados em coleções muito esparsas. Normalmente adotam o algoritmo K-means, um algoritmo simples e eficiente. Contudo com algumas limitações conhecidas, como a necessidade de definição do número de grupos a priori, a sensibilidade à escolha inicial dos centróides na criação dos grupos, a possibilidade de gerar grupos vazios, entre outros. Tem como principal contribuição, a aplicação e avaliação do algoritmo de propagação de afinidades, como algoritmo principal em SRs baseados em filtragem colaborativa, no cenário de recomendação de filmes, e a análise dos resultados com diferentes tempos de término do algoritmo de agrupamento e número de grupos gerados. Uma limitação deste trabalho é a não avaliação de outras técnicas de agrupamento. Como, por exemplo, uso do algoritmo BIRCH, que permite a aglomeração de itens com realocação.

Leng et al. (2018) apresentam, em seu artigo, um estudo iniciado em (Colucci et al., 2016), de uma abordagem dirigida a dados para avaliar a similaridade entre filmes. Foi realizada a comparação entre um método supervisionado, no qual usuários realizaram a classificação de filmes, com base na sua percepção de similaridade, contra métodos não supervisionados. Em (Colucci et al., 2016), foram realizados testes com quatro métodos, dois usando filtragem colaborativa (um utilizando correlação de Pearson e outro a função cosseno) e dois baseados em conteúdo (um caixa-preta extraído do site TMDB e o outro que faz uso de metadados de filmes extraídos do site OMDB e comparação por meio de TF-IDF). Leng et al. (2018) em seu estudo, aperfeiçoa o trabalho iniciado através das seguintes modificações. No método baseado em conteúdo, que extrai metadados do site OMBD, são acrescentados novos metadados. Além disso, adota três métodos de comparação, tanto para o métodos baseados em filtragem colaborativa, quanto para os métodos baseados em conteúdo. Adota também, três formas de construção de similaridade em filtragem colaborativa, e três técnicas de aprendizado supervisionado. Como resultado, estão disponíveis 27 métodos para avaliação, gerados pela combinação dos métodos. Esse trabalho realiza um estudo sobre algoritmos, com objetivo de realizar a predição de filmes, tanto com filtragem colaborativa, quanto baseada em conteúdo. Na abordagens baseadas em conteúdo, são avaliadas a qualidade de algumas fontes de dados. Contudo, são trabalhos direcionados a um domínio. Na abordagem baseada em conteúdo, não faz uso da LOD e, em uma de suas fontes, são determinados de forma manual, quais são os metadados e qual a ponderação de cada um.

Em (**Song et al., 2017**), são propostos dois algoritmos de seleção de candidatos, com objetivo de melhorar os sistemas de correlação de entidades. Apesar destes algoritmos não terem sidos empregados em SRs, podem ser adaptados para este fim. Isto pois, o processo de seleção de candidatos, nada mais é, do que a seleção de entidades similares o suficiente, podendo ser empregadas como entidades correspondentes, em um SR baseado em conteúdo. O primeiro algoritmo, denominado HistSim, faz uso de histórias de correspondência para realizar a poda de instâncias que não são suficientemente similares. Desta forma, diversas comparações podem deixar de ser realizadas. Este método realiza a comparação das diversas entidades em um lista. Uma entidade é comparada com todas as entidades após ela e, a medida que são realizadas as comparações, é construído um histórico de correlação. A hipótese com esta técnica é comparar de forma mais refinada, apenas entidades que compartilharem uma quantidade suficiente de entidades em seus históricos. O segundo algoritmo proposto, denominado DisNGram, realiza um aprendizado não supervisionado de chaves de seleção de candidatos, e faz uso de técnicas de indexação, para selecionar, de forma escalável, pares de entidades candidatas. Este algoritmo realiza a descoberta de propriedades de forma iterativa, considerando a relação entre os fatores discriminalidade (percentual de entidades que dada uma propriedade tenham valores diferentes), e cobertura (percentual de entidades que usem uma propriedade). Esse trabalho tem, como limitação, a não avaliação em SR, para verificação de sua eficiência em tais sistemas. Foi utilizado no problema de deduplicação ou correlação de entidades da LOD.

Em (**Leal et al., 2012**), é apresentado o Shakti, uma ferramenta para extrair uma ontologia de um determinado domínio da DBPedia, e usá-la para calcular a relação semântica de termos definidos como rótulos, de conceitos nessa ontologia. Segundo o Leal et al. (2012), seu algoritmo, diferentemente de outros, tem o foco na noção de proximidade entre termos, ou seja, quão conectados dois termos estão. Como forma de exemplificar, tome um termo que esteja conectado a dois outros termos, tendo a mesma distância, contudo possui mais conexões com um destes. Logo, este deve ser considerado mais próximo. Termos com mais conexões, devem ser considerados mais próximos e portando tem alta proximidade. Umas das contribuições principais deste trabalho é o fato de desenvolver um novo algoritmo para realizar a verificação de similaridade semântica entre dois termos da LOD. Como ponto a melhorar, foi considerado o fato de que, para o algoritmo funcionar perfeitamente, devem ser identificadas classes ou conceitos relevantes sobre um domínio, bem como suas conexões, e também o fato de que as conexões entre dois termos, devem ser ponderadas em função de sua relevância. Caso seja necessário usar em outro domínio, será necessário reconfigurar estes parâmetros. A ponderação das conexões foi realizada com base na percepção dos autores, em uma faixa entre 0 e

10.

Em (**Di Noia et al., 2012**), é demonstrado como os conjuntos de dados da LOD são maduros o suficiente para serem utilizados como fonte principal de dados para SRs. O trabalho proposto, apresenta uma abordagem que realiza a adaptação dos dados disponibilizados em grafos RDF para *Vector Space Model* (VSM), através da conversão destes dados em uma matriz contendo recurso/propriedade/objeto. Cada fatia desta matriz representa uma entidade. Dada uma propriedade, a entidade é vista como um vetor, cujos valores são ponderados por TF-IDF, ou, como referido pelos autores, como *Resource Frequency-Inverse Movie (ou entidade) Frequency* (RF-IMF). Após a construção da matriz RF-IMF, a similaridade de cada propriedade é calculada pela correlação dos vetores, através do cosseno do ângulo entre eles. Este trabalho tem, como contribuição principal, o desenvolvimento de um SR baseado em conteúdo que explora com sucesso, somente conjuntos de dados da LOD e realiza a avaliação da proposta através do uso da DBpedia e do conjunto de dados MovieLens. Como ponto de melhoria, foi considerado o fato de não ter sido realizada uma ponderação entre as propriedades, antes de realizar a verificação de similaridade. Todas as propriedades são consideradas como tendo o mesmo peso e pode não representar a realidade com precisão.

O trabalho apresentado nesta dissertação, visa realizar a predição de itens com base em conteúdo. Seu diferencial frente aos trabalhos citados está no fato de realizar a exploração dos relacionamentos semânticos da LOD, através da descoberta de características relevantes, sem necessidade de intervenção de um especialista de domínio. Também é realizada a ponderação de tais características. Estas características são utilizadas durante o processo de comparação entre instâncias, para estimativa da similaridade entre elas.

Capítulo 3

Abordagem Proposta

Este capítulo descreve a abordagem proposta para explorar as relações semânticas da LOD, estimar a similaridade entre itens (instâncias) e gerar uma lista de itens semelhantes, que um SR baseado em conteúdo poderá usar posteriormente. A Seção 3.1 apresenta a visão geral da proposta. Na Seção 3.2, são apresentados cada um dos processos de coleta de dados. A Seção 3.3 apresenta métricas sobre os dados coletados, como índices de discriminalidade, frequência e TF-IDF. A Seção 3.4 apresenta as atividades realizadas como pré-processamento dos dados coletados. E a Seção 3.5 apresenta os métodos de estimativa de similaridade, utilizados para computar os *scores* de similaridade entre as instâncias.

3.1 Visão Geral

Uma visão geral da proposta é apresentada no fluxograma da Figura 3.1. Cada etapa do fluxograma corresponde a um processo e as etapas estão organizadas basicamente em quatro grupos: Coleta de Dados, Estimativa de Métricas, Pré-Processamento e Estimativa de Similaridade. Cada um dos grupos com suas respectivas etapas serão explanadas a seguir.

3.2 Coleta de Dados

Este grupo de processos compreende três etapas e é executado mediante definição de um *Endpoint* SPARQL, e qual o domínio dos dados a serem coletados. A coleta de dados ocorre nas etapas a seguir: coleta de instâncias, coleta de predicados e coleta de vizinhança.

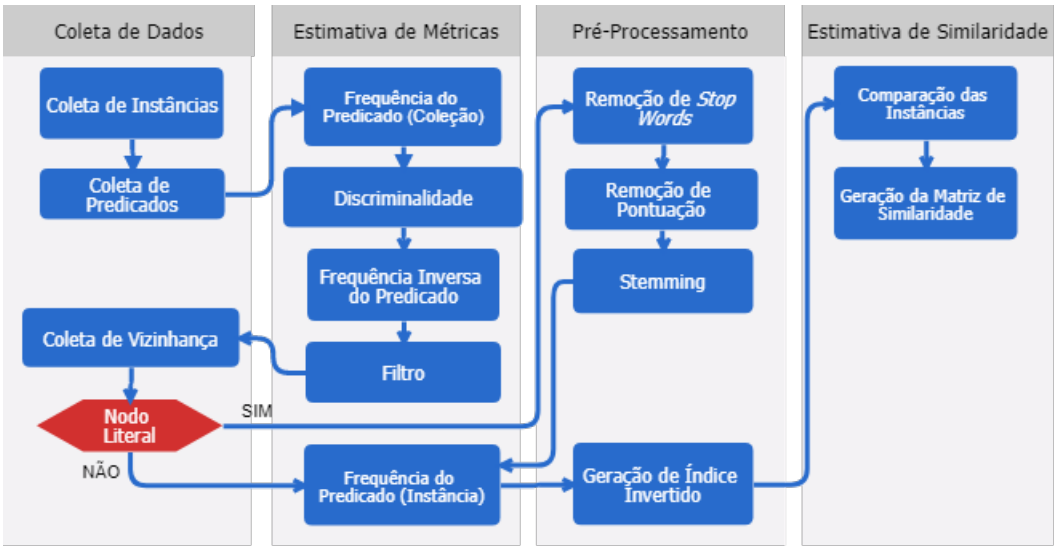


Figura 3.1: Visão geral do Método Proposto

.

3.2.1 Coleta de Instâncias

Nesta etapa, são coletadas as instâncias por meio do *endpoint* SPARQL definido, conforme o domínio de informação, previamente especificado. O objetivo dessa coleta é reunir todas as instâncias, conforme o domínio, e estimar a similaridade entre elas.

A coleta é realizada por meio de uma consulta SPARQL ao *endpoint*, e nela é necessário especificar qual o domínio da informação a ser retornado. Uma consulta desta natureza pode ser visualizada na Consulta 1. Neste exemplo, são coletadas todas as instâncias do domínio museus. Devido a limitação do número de registros retornados pela DBpedia, foi necessário informar os parâmetros *limit* e *offset*, para execução das consultas de forma paginada.

A linha número 1 da consulta, determina um prefixo para o domínio da informação a ser recuperada, usado na linha 4. A linha 2 especifica qual a informação será recuperada e as linhas 3 e 4 definem a condição, neste caso uma tripla RDF contendo sujeito, predicado e objeto. O sujeito (**?instance**) é a instância a ser recuperada, o uso do caracater interrogação mostra que esta é uma variável. O predicado define o tipo do objeto, neste caso é usada a palavra-chave SPARQL ***a***, que é um atalho para o predicado comum **rdf:type**. E o objeto (**dbo:Museum**) é o tipo da instância a ser retornada em ?instance.

A Tabela 3.1, contém exemplos de instâncias obtidas do *endpoint* DBpedia no domínio museus.

Consulta 1 Extração de Instâncias.

```
PREFIX dbo: <http://dbpedia.org/ontology/
SELECT  ?instance
WHERE
{  ?instance  a dbo:Museum ;  }
```

Tabela 3.1: Exemplo de Instâncias da DBpedia do domínio museus

Instância
:Underwater_archaeology
:Brody_Museum_of_History_and_District_Ethnography
:Museum_Computer_Network
:National_Archaeological_Museum,_Tirana
:National_Archaeological_Museum_(Bulgaria)
:National_Archaeological_Museum_(Florence)
:National_Archaeological_Museum_(France)
:New_Sweden_Farmstead_Museum
:The_Power_Plant
:Abbot_Hall_Art_Gallery

3.2.2 Coleta de Predicados

Nesta etapa, são coletados todos os predicados, relacionados às instâncias reunidas na etapa anterior. As arestas nos gráficos RDF representam as propriedades dos nodos sujeito (instâncias) ligados aos nodos objeto, que são as informações em si.

Para obter os predicados das instâncias, também é executada uma consulta SPARQL. A consulta é definida com base no número de níveis entre o nó sujeito (instância) e os nodos representando os objetos. No caso de um nível, a consulta recupera todos os predicados cujas arestas são diretamente incidentes sobre às instâncias (nodos sujeito), como arestas de entrada ou saída. Para dois níveis, a consulta recupera todos os predicados, incidentes sobre os nodos objeto incidentes sobre os nodos de instância, pelos predicados de primeiro nível, seja por arestas de entrada ou de saída. Como forma de exemplificar observe a Figura 3.2, na qual é apresentado um exemplo de grafo RDF sobre filmes. Considerando o nodo sujeito rotulado por *Pulp Fiction*, são predicados de primeiro nível as propriedades *director* e *starring*, como predicado de segundo nível há a propriedade *subject*.

Além disso, como na consulta para coletar as instâncias, é necessário fornecer como entrada o domínio da informação. A Consulta 2 mostra um exemplo de uma consulta SPARQL para coleta de predicados de primeiro nível. Nesta consulta é necessário usar o operador *UNION* para realizar a união entre os predicados que tem origem na instância e predicados que incidem sobre a instância. A Tabela 3.2 contém exemplos de predicados, todos são rotulados por URI's assim como os rótulos

de recursos.

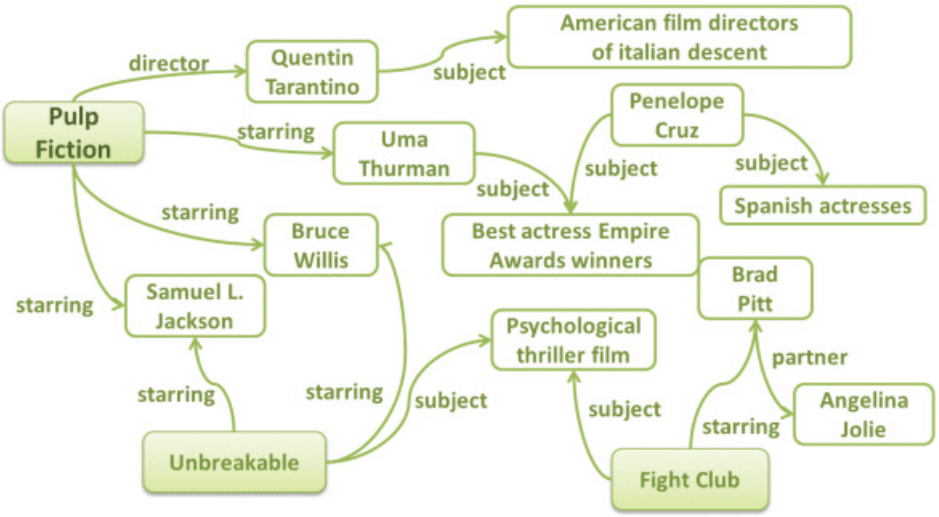


Figura 3.2: Fragmento de dados da DBpedia relativos a filmes.
Fonte: Di Noia et al. (2016).

Consulta 2 Coleta de Predicados.

```
1  SELECT DISTINCT ?predicate
2  WHERE
3  {
4    { ?instance  a "+domain+" .
5        ?instance ?predicate ?o1 .
6    }
7    UNION
8    { ?instance  a "+domain+" .
9        ?s1 ?predicate ?instance
10   }
11 }
12 OFFSET  0
13 LIMIT   10000
```

3.2.3 Coleta de Vizinhança

O processo de comparação entre as instâncias é realizado com base em seus dados, chamados de vizinhança da instância. Os dados das instâncias, considerando grafos RDF, estão presentes nos nodos objetos, em primeiro ou segundo níveis, conectados às instâncias através de predicados. A extração destes dados é realizada por meio da execução da Consulta 3. O objetivo desta consulta é realizar a extração dos nodos objetos, linhas 6 e 10 da consulta, ou seja a vizinhança da instância.

Tabela 3.2: Exemplos de predicados da DBpedia do domínio museus.

Predicados
rdf:type
owl:differentFrom
rdfs:seeAlso
owl:sameAs
dbpedia:ontology/architecturalStyle
dbpedia:ontology/curator
dbpedia:ontology/director
dbpedia:ontology/floorArea
dbpedia:ontology/floorCount
dbpedia:ontology/foundingDate

Devido a necessidade de recuperar a vinhança com um ou mais níveis de distância da instância, esta consulta pode ser realizada de forma recursiva, recuperando objetos dos objetos quando estes forem URIs, em vários passos de distância da instância. Quando este for o caso é necessário realizar a aplicação de filtros para que predicados, já consultados anteriormente não sejam considerados. Outro filtro aplicado, verifica se o tipo do dado é um literal e filtra dados em idiomas que não sejam o inglês.

Todos os nodos, cujos conteúdos sejam literais, passarão por um pré-processamento para realizar a remoção de *stop words*, pontuação e redução de palavras flexionadas a sua raiz (*stemming*).

A Figura 3.3 mostra um exemplo de vizinhança extraída da DBpedia, relativa à instância identificada pela URI abaixo. Todas as elipses são rotuladas por URI's dos recursos que representam, os retângulos representam objetos cujos tipo dos dados são literais e são rotulados por estes dados, a elipse ao centro representa a instância e as arestas rotuladas e direcionadadas representam os predicados.

http://dbpedia.org/resource/Museum_of_Ancient_Seafaring

3.3 Estimativa de Métricas

Pode-se obter várias métricas dos predicados nas instâncias, para estimar sua relevância na coleta de dados e permitir seu uso. A seguir, são descritas as métricas estimadas neste trabalho e usadas na abordagem proposta.

3.3.1 Cobertura dos Predicados na Coleção

A estimativa da cobertura de um predicado visa determinar sua frequência em uma coleção de instâncias e, é utilizada em um processo de filtragem com o intuito de

Consulta 3 Extração de Vizinhança.

```
SELECT  ?predicate ?object
WHERE
{
  {
  <http://dbpedia.org/resource/Museum_of_Ancient_Seafaring>
    ?predicate  ?object
  }
  UNION
  {
  ?object  ?predicate
  <http://dbpedia.org/resource/Museum_of_Ancient_Seafaring>
  }
  FILTER (
    str(?predicate) !=
      "http://dbpedia.org/ontology/wikiPageWikiLink"
  )
  FILTER (
        (( ! isLiteral(?object) ) || ( lang(?object) = "" ))
      || langMatches(lang(?object), "EN")
  )
}
OFFSET  0
LIMIT   10000
```

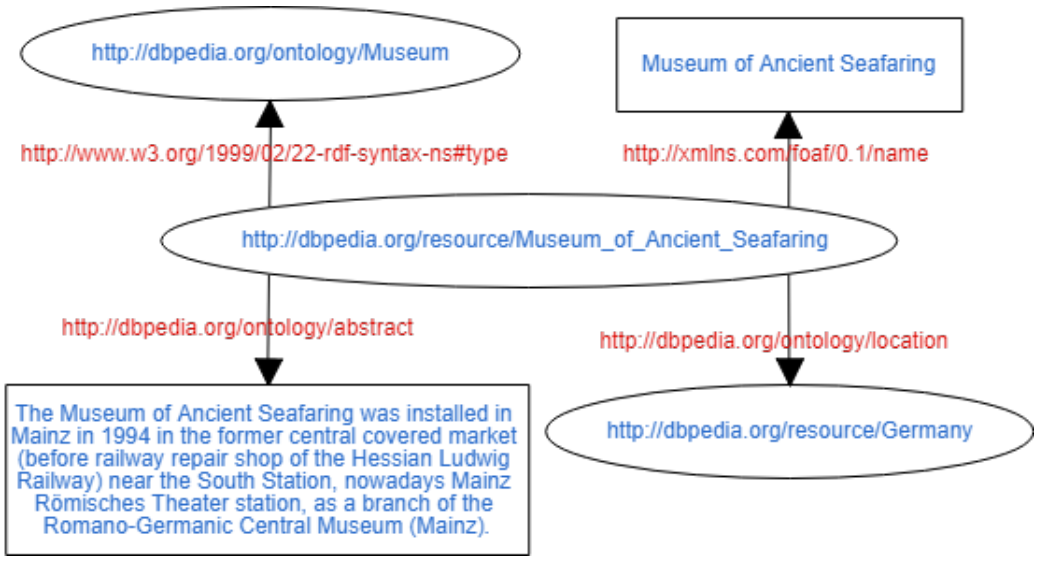


Figura 3.3: Exemplo de vizinhança

reduzir a quantidade de predicados, reduzindo a quantidade de dados. Basicamente, a cobertura é a relação entre a quantidade de instâncias com um determinado predicado diante da quantidade total de instâncias na coleção. Na Equação 3.1, o p_i representa o i-ésimo predicado. O cálculo deve ser realizado para cada um dos

predicados, desta maneira será obtida uma lista, com a frequência de cada um.

$$Cov_{p_i} = \frac{|Conjunto\ de\ instâncias\ distintas\ com\ p_i|}{|Conjunto\ total\ de\ instâncias|} \tag{3.1}$$

3.3.2 Discriminalidade

O conceito de discriminalidade, adotado neste trabalho, foi proposto por Song et al. (2017). A estimativa da discriminalidade visa realizar a poda de predicados que não sejam suficiente discriminantes, reduzindo o volume de dados a serem processados. A discriminalidade de uma tripla é obtida a partir do seu predicado e seu objeto. Dado um predicado, quanto mais diverso é o valor do objeto, mais discriminante ele é. Pode-se tomar como exemplo a discriminalidade do predicado *id* em uma coleção. Cada instância tem um identificador, logo pode-se concluir que a discriminalidade é de cem por cento, visto que não podem existir duas instâncias com mesmo identificador. A discriminalidade de um predicado é a relação entre a quantidade de objetos distintos, associados ao predicado, e a quantidade total de objetos com o predicado.

$$Dis_{p_i} = \frac{|Conjunto\ de\ objetos\ distintos\ com\ p_i|}{|Conjunto\ de\ objetos\ com\ p_i|} \tag{3.2}$$

3.3.3 *Predicate Frequency-Inverse Instance Frequency* (PF-IPF)

A hipótese deste cálculo é ponderar os predicados como se fossem termos em um documento, assim como ocorre no *Term Frequency - Inverse Document Frequency* (TF-IDF). No contexto deste trabalho, os documentos são análogos às instâncias, os termos análogos aos predicados e a coleção de documentos é análoga à coleção de instâncias. A teoria do TF-IDF é que quanto maior a frequência de um termo em um documento, maior será sua relevância, que é minimizada se o termo está presente em vários documentos.

Após o cálculo, o valor obtido será utilizado no processo de comparação das instâncias, ponderando o resultado da comparação.

O cálculo do IIF é realizado após a recuperação dos predicados das instâncias através da Equação 3.3.

$$IIF_{p_i} = log\frac{|Conjunto\ total\ de\ instâncias|}{|Conjunto\ de\ instâncias\ distintas\ com\ p_i|} \tag{3.3}$$

Já o cálculo da frequência do predicado sobre a instância, é realizado através da Equação 3.5.

$$PF_{p_i,I} = \frac{|\ Quantidade\ de\ predicados\ p_i\ na\ instância|}{|Quantidade\ total\ de\ predicados\ na\ instância\ |} \tag{3.4}$$

O PF-IIF, assim como seu análogo, é obtido a partir do produto do dois valores.

$$PF\text{-}IPF_{p_i,I} = PF_{p_i,I} \times IIF_{p_i} \tag{3.5}$$

3.4 Pré-processamento

O pré-processamento é realizado com o objetivo de realizar a limpeza dos dados. Desta forma, reduzindo ruídos durante o processo de comparação entre as instâncias. Nesta etapa, também é construído um índice invertido, cujo objetivo é tornar mais eficiente o processo de estimar a similaridade, através da comparação entre as instâncias.

3.4.1 Remoção de *stop words*, pontuação e *stemming*

O processo de remoção de *stop words*, remoção de pontuação e *stemming*, ocorre durante a fase de aquisição de dados da vizinhança das instâncias. Todos os nodos, cujo tipo de dado é literal, passam por este processo. Os nodos recursos, cujos valores são URI's, não passam por este processo, por potencialmente conectar o nodo a outro, com um passo a mais de distância da instância.

Nos literais, as *stops words* normalmente são os artigos, conjunções e preposições, e tem papel importante na formação das sentenças. A seguir são listados alguns exemplos de *stop words* da língua inglesa: *a, and, for, in, of, that, the, to* dentre vários outros. A remoção de *stop words* é realizada devido a baixa relevância dessas palavras, no processo de comparação, e pelo fato de afetar negativamente resultados, com a possibilidade de inserção de ruídos.

O *stemming* é o processo de reduzir palavras flexionadas por sufixos, a sua raiz. Desta maneira, palavras consideradas diferentes, devido a adição de sufixos, são consideradas iguais. Na língua inglesa há diversos exemplos, tais como: child, children, play, played, talk, talks, talked, entre outros.

3.4.2 Geração de Índice Invertido

A adoção do índice invertido, tem como objetivo, agilizar o processo de comparação realizado entre as diversas instâncias, por meio da redução do número de comparações.

No contexto de recuperação de informação em textos, os índices invertidos são construídos como um dicionário de termos, através da separação dos termos dos

documentos, *Tokenization*, e posterior limpeza, através da remoção de *stop words, stemming*, entre outros. Após construído, o índice invertido é composto de duas partes, o dicionário de termos e a lista de documentos, como pode ser observado na Figura 3.4 (Manning et al., 2008).

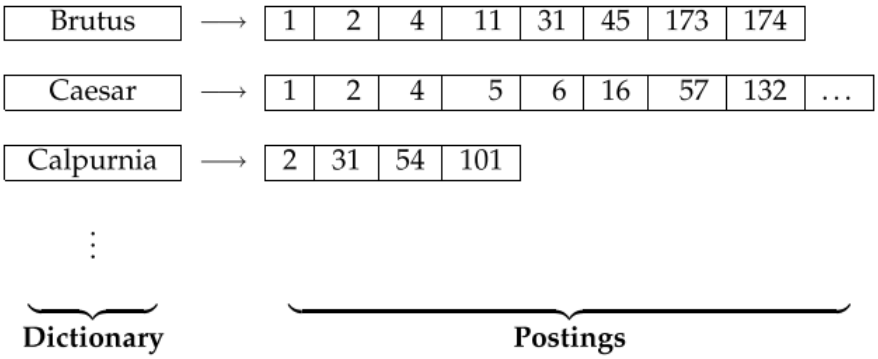


Figura 3.4: Índice Invertido
Fonte: Manning et al. (2008)

Os índices invertidos podem ser baseados em zonas, as zonas são metadados relacionados aos termos, como por exemplo, título, autor, data de publicação entre outros, como campos em um formulário. A construção do índice invertido baseado em zonas, pode ser realizada para cada zona de forma separada como na Figura 3.5. Desta maneira o dicionário de termos tem seu tamanho multiplicado pela quantidade de zonas. Nesta figura, o termo William é exibido três vezes, uma vez para cada zona associada (Manning et al., 2008).

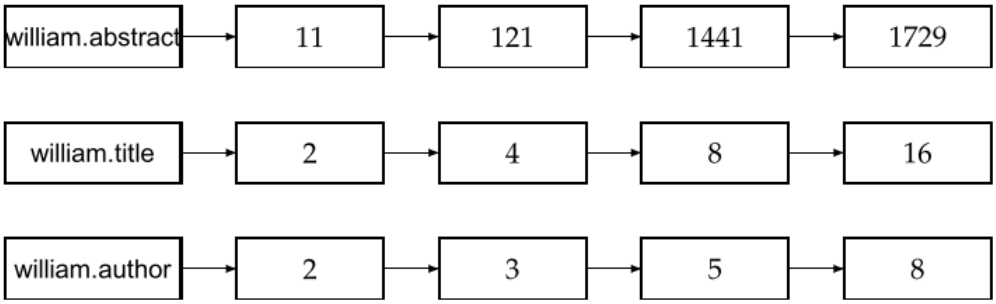


Figura 3.5: Índice de Zonas com a Zona associada ao dicionário
Fonte: Manning et al. (2008)

O tamanho do dicionário de termos, pode ser reduzido através da codificação das zonas junto a lista de documentos, ao invés do dicionário, como na Figura 3.6.

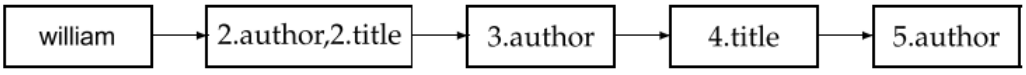


Figura 3.6: Índice de Zonas com a Zona associada a lista de documentos
Fonte: Manning et al. (2008)

No contexto deste trabalho, o índice de zonas construído possui um dicionário de termos, compostos por uma lista de termos extraídos dos literais e também recursos (por ser um URI cada recurso é representado como um termo). As zonas são os predicados que ligam estes dados às instâncias ou seja as arestas.

Como forma de exemplificar, considere a Figura 3.3. Dada a instância Museum_of_Ancient_Seafaring, alguns exemplos de termos extraídos da vizinhança são: http://dbpedia.org/ontology/Museum, Museum, Ancient, Seafaring, Mainz, 1994, http://dbpedia.org/resource/Germany e outros.

A Figura 3.7, mostra a proposta de construção do índice invertido, utilizando os predicados como zonas. O dicionário é composto pelos termos extraídos dos literais mais os URIs, como dbo/Museum. A lista de instâncias e predicados, que ligam o termo à instância, estão representados pelos retângulos do lado direito da figura.

Figura 3.7: Índice Invertido com Zonas representadas por Predicados

3.5 Estimativa de Similaridade

A estimativa de similaridade entre as instâncias é realizada por meio da comparação entre elas e ocorre conforme as etapas a seguir:

1. Iteração pelo dicionário de termos do índice invertido e, para cada termo, recuperação da respectiva lista de instâncias em função dos predicados de ligação.

2. Iteração pela lista de instâncias, comparando cada instância da lista com todas as posteriores.

3. Comparação, propriamente dita, entre as instâncias por meio de sua vizinhança.

Após comparar duas instâncias, o *score* obtido é atribuído a uma posição, em uma matriz de similaridade. Essa matriz é utilizada no processo de geração de listas ordenadas, através da ordenação das instâncias em função dos *scores*, evitando que duas instâncias sejam comparadas mais de uma vez.

Na matriz de similaridade, cada uma das linhas e cada uma das colunas representam as instâncias. A interseção entre uma linha e uma coluna, representa o *score* de similaridade entre duas instâncias. A Tabela 3.3 mostra um exemplo de matriz de similaridade. A célula destacada em cinza escuro representa o score de similaridade entre as instâncias 4 e 5.

Tabela 3.3: Exemplo de Matriz de Similaridade

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**...**
1	1	0.6	0.4	0.9	0.2	0.7	0.4	0.5	0.6	...
2	0.8	1	0.6	0.4	0.9	0.2	0.7	0.4	0.5	...
3	0.1	0.6	1	0.9	0.2	0.7	0.4	0.5	0.6	...
4	0.6	0.2	0.1	1	0.4	0.9	0.2	0.7	0.4	...
5	0.7	0.1	0.6	0.4	1	0.2	0.7	0.4	0.5	...
6	0.1	0.6	0.4	0.9	0.2	1	0.4	0.5	0.6	...
7	0.9	0.1	0.6	0.4	0.9	0.2	1	0.4	0.5	...
8	0.1	0.6	0.4	0.9	0.1	0.6	0.4	1	0.2	...
9	0.1	0.6	0.4	0.9	0.2	0.7	0.4	0.5	1	...
...	...	...	...	...	...	...	...	...	...	

A comparação entre as instâncias ocorre por meio da comparação entre suas vizinhanças, ou seja, os nodos adjacentes às instâncias. O processo de comparação foi realizado de duas maneiras diferentes e, para cada uma dessas maneiras, foram utilizados dois algoritmos de comparação, softTF-IDF e Cosseno.

Dado o Algoritmo 1, o processo de comparação ocorre da seguinte maneira. O algoritmo recebe como entradas as instâncias A e B, na linha 3 é recuperada a vizinhança de A, a qual será iterada através do laço entre as linhas 4 e 23. A cada iteração, a variável vA representa um vizinho de A e para cada vA é recuperada a vizinhança de B compatível. Por compatível entende-se todos os vizinhos que estejam ligados a instância B através de uma aresta rotulada com o mesmo predicado utilizado por vA. Na linha 7, é verificado se o conteúdo de vA é um literal. Se verdadeiro, ele é comparado com cada um dos vizinhos de B compatíveis, através das medidas de similaridade Cosseno ou SoftTF-IDF. O valor retornado é atribuído a variável *score*. Caso não seja um literal, ou seja, um recurso, linha 13, são verificados se os recursos são iguais ou não. Se forem iguais é adicionado 1.0 a variável *score*. Após realizar a verificação de vA com todos os vizinhos de B compatíveis, são recuperados os valores de TF-IDF do predicado, para cada uma das instâncias. A média entre eles é utilizada como fator de ponderação sobre a variável *score*.

Algorithm 1 Comparação entre instâncias com base no grafo de características

```
1:  function COMPARA(A, B)
2:      sim ← 0.0
3:      lvA ← recupera vizinhança de A
4:      for each vA ∈ lvA do
5:          score ← 0.0
6:          lvAux ← recupera vizinhança de B compatível com vA
7:          if vA é Literal then
8:              for each vAux ∈ lvAux do
9:                  if vA e vAux compartilham um token then
10:                     score ← score + cosseno entre vA e vAux
11:                 end if
12:             end for
13:         else
14:             for each vAux ∈ lvAux do
15:                 if vA igual a vAux then
16:                     score ← score + 1.0
17:                 end if
18:             end for
19:         end if
20:         tFIDFA ← TF-IDF do predicado para instância A
21:         tFIDFB ← TF-IDF do predicado para instância B
22:         sim ← sim + ((tFIDFA + tFIDFB)/2) * score
23:     end for
24:     jaccard ← sim/(|vizinhancadeA| + |vizinhancadeB| − sim)
25:     return jaccard
26: end function
```

Dado o algoritmo 2, o processo de comparação ocorre da seguinte maneira. O algoritmo recebe como entradas as instâncias A e B. Na linha 3, é recuperada a vizinhança de A, a qual será iterada no laço iniciado na linha 5, realizando a concatenação da vizinhança em um texto. O mesmo processo ocorre com a vizinhança de B, sua vizinhança é recuperada na linha 8, é iterada a partir da linha 10, realizando a concatenação da vizinhança em um texto. A comparação entre os textos resultantes ocorre na linha 13, através das mediadas de similaridade Cosseno ou SoftTF-IDF.

Algorithm 2 Comparação entre instâncias com base no conteúdo textual

```
1: function COMPARA(A, B)
2:     sim ← 0.0
3:     lvA ← recupera a lista de vizinhos de A
4:     nAS ← NULL
5:     for each vA ∈ lvA do
6:         nAS ← lastPredicate(vA) + lastElement(vA)
7:     end for
8:     lvB ← recupera a lista de vizinhos de B
9:     nBS ← NULL
10:    for each vB ∈ lvB do
11:        nBS ← lastPredicate(vB) + lastElement(vB)
12:    end for
13:    sim ← cosine(nAS, nBS)
14:    return sim
15: end function
```

Capítulo 4

Avaliação Experimental

Este capítulo descreve a avaliação experimental da abordagem proposta. Na Seção 4.1, são apresentadas as coleções de dados utilizadas. Na Seção 4.2, é apresentada a métrica de avaliação adotada. Na Seção 4.3, são apresentadas as *baselines* selecionadas. Na Seção 4.4, são apresentados os procedimentos utilizados na realização dos experimentos. Finalmente na Seção 4.5, são apresentados os resultados.

O desenvolvimento da abordagem e os experimentos foram executados em um computador com processador Intel Core I5-6200U, com 8GB de memória RAM, e sistema operacional operacional *Microsoft Windows 10 Home*. Para codificação foi utilizado o ambiente de desenvolvimento integrado *Eclipse Java EE IDE for Web Developers*, versão *Neon.3 Release (4.6.3)*, com a linguagem *Java,* na versão 1.8.0_-171. Todo o código desenvolvido está disponível no GitHub, acessível através do *link* ***https://github.com/italompereira/cbrslod***.

4.1 Coleções de Dados

Para avaliação da proposta foram utilizadas duas coleções de dados, extraídas do *endpoint* DBpedia[1]. A primeira é composta por instâncias de museus e a segunda por instâncias de filmes. No domínio de museus, foram coletadas 10153 instâncias contendo 111 predicados, em média, por instância no primeiro nível e, no domínio de filmes, 131296 instâncias, com média de 121 predicados por instância, também no primeiro nível. Em cada um dos domínios de informação, após o processo de comparação, foi gerada uma matriz de similaridade contendo o *score* de similaridade entre cada par de instâncias.

Para realizar a avaliação da eficácia da abordagem em relação às *baselines*, foram usados dois *Ground Truths*, os quais serão discutidos nas Subseções 4.1.1 e 4.1.2, relacionados aos domínios de museus e filmes, respectivamente.

[1]http://DBpedia.org/sparql

4.1.1 Construção do *ground truth* para o domínio de museus

A construção do *ground truth* no domínio de museus seguiu o mesmo procedimento adotado em (Ruback et al., 2017). Nesse trabalho, os autores, para realizar a avaliação da eficácia, optaram pela criação de uma base de dados, explorando informações de um conhecido site de história da arte, denominado SmartHistory[2]. O SmartHistory é uma organização sem fins lucrativos, que torna possível o aprendizado gratuito da história da arte. O site fornece vários artigos discutindo as obras mais importantes, desde a arte antiga à arte contemporânea (Ruback et al., 2017).

Ruback et al. (2017) apontam duas justificativas para a escolha do SmartHistory como fonte de informações, para a construção do *Ground Truth*: a primeira, é o fato de ser uma rica fonte de dados sobre museus, cujos autores são especialistas no domínio, e a segunda é devido ao fato de suas informações serem independentes da DBpedia.

A classificação dos museus através do SmartHistory, foi realizada com base em seus artigos. Cada artigo, normalmente está relacionado a uma obra de arte e é categorizado, conforme uma classificação hierárquica. A classificação inclui períodos de tempo, movimentos de arte, entre outros, além de citar o museu que abriga a obra. A similaridade entre dois museus é definida conforme as categorias encontradas nos artigos, que citam os museus. A similaridade foi computada utilizando a similaridade do cosseno, entre as categorias do museus.

4.1.2 Construção do *ground truth* para o domínio de filmes

Para este *ground truth* foi utilizada a base *MovieSim-1*. Em (Colucci et al., 2016), os autores avaliaram algoritmos de similaridade item-item, no domínio de filmes, com base na percepção de similaridade de usuários. Para realização da avaliação foi necessário criar uma base de dados de similaridade de filmes, com base na percepção dos usuários.

Conforme Colucci et al. (2016), para criação dessa base de dados, denominada MovieSim-1, participaram do experimento 14 estudantes de graduação, com média de idade de 25 anos. Este conjunto de dados possui 3803 rótulos binários de 14 usuários e 143 filmes exclusivos selecionados. A tarefa dos usuários foi selecionar um filme e então avaliar a similaridade em relação a outros filmes. A avaliação foi realizada de forma binária, ou seja, é similar ou não é similar. Para cada filme selecionado, o usuário poderia avaliar até 20 outros filmes como sendo similares ou não.

Os filmes exibidos aos usuários foram obtidos a partir do movieLens[3]. Por este

[2]http://smarthistory.org
[3]https://grouplens.org/datasets/movielens/

motivo, as listas de filmes exibidas não eram verdadeiramente randômicas. Foram selecionados e exibidos os filmes mais populares do movieLens, tendo como base as 20 categorias mais populares do TMDb. Essa opção foi justificada pelos autores com a afirmação de que listas puramente randômicas, tendiam a exibir filmes não conhecidos pelos usuários.

4.2 Métrica de avaliação

Para estimar a similaridade entre as instâncias da LOD, foi realizada sua comparação através dos objetos conectados a elas por meio dos predicados. Após a comparação foi gerada uma matriz de similaridade. Para cada um dos itens desta matriz, foi gerada uma lista de instâncias similares, ordenadas em função índice de similaridade estimado. Cada uma das listas ordenadas foram comparadas às listas correspondentes, geradas pelos *Ground Truths*, conforme o domínio da informação avaliado. Para esta comparação foi utilizada a métrica NDCG, descrita na Seção 2.1.3, usada para avaliar a qualidade de *rankings*. A opção por essa métrica é justificada pela necessidade de avaliação da qualidade das listas ordenadas de itens, geradas pela abordagem proposta, em relação as listas de itens do *Ground Truth*. Os resultados foram comparados aos resultados obtidos pelas *baselines* e os resultados são discutidos na Seção 4.5.

4.3 *Baselines*

Para avaliação da eficácia da abordagem proposta foram usados os resultados de duas ferramentas como baselines, uma para cada domínio de informação. A primeira, denominada SELEcTor, é relacionada ao domínio de museus e é descrita na Subseção 4.3.1. A segunda, denominada *The Movie Database* (TMDb), é um banco de dados de filmes e séries online e aberto, ela está descrita na Subseção 4.3.2.

4.3.1 *SELEcTor*

Em (Ruback et al., 2017), é proposta o SELEcTor. Esse trabalho tem como objetivo descobrir instâncias similares na LOD, a partir da exploração de listas ordenadas de características das instâncias.

Nos experimentos realizados, foram utilizadas instâncias extraídas da DBpedia, pertencentes ao domínio museus. Entre os predicados (metadados) vinculados às instâncias de museus, Ruback et al. (2017) selecionaram, movimentos de arte e trabalhos de arte, sendo que, os movimentos de arte classificam cada trabalho de arte em um museu. A similaridade entre as instâncias, é estimada com base na

quantidade trabalhos de arte, em cada movimentos de arte. A ideia dessa proposta baseia-se no fato de que, dois museus que tenham movimentos de artes similares, também serão similares, e o grau de similaridade é medido conforme as listas de movimento de arte, ordenadas conforme o número de trabalhos de arte. Na Tabela 4.1, são exibidas duas instâncias e suas respectivas listas de movimentos de arte, ordenadas de acordo com a quantidade de obras de arte em cada movimento. A comparação entre as duas listas foi feita com base na métrida *Ranked Biased Overlap* (Webber et al., 2010).

Tabela 4.1: Comparação de características ordenadas

Fonte: Ruback et al. (2017)

J Paul Getty ranked features	Louvre ranked features
dbr:Symbolism_(arts)	dbr:Romanticism
dbr:Baroque	dbr:High_Renaissance
dbr:Expressionism	dbr:Neoclassicism
dbr:Romanticism	dbr:Baroque
dbr:High_Renaissance	dbr:Italian_Renaissance
dbr:Dutch_Golden_Age_painting	dbr:Dutch_Golden_Age_painting
dbr:Academic_art	dbr:The_Renaissance
dbr:Post-Impressionism	dbr:Classicism
dbr:Mannerism	dbr:Realism_(arts)
	dbr:Flemish_Baroque_painting
	dbr:Early_Netherlandish_painting
	dbr:Caravaggisti

Avaliação do SELEcTor

Para avaliar o SELEcTor em relação ao SmartHistory, foi necessário uma seleção de museus comuns entre a DBpedia e o SmartHistory. Para tal foram adotados dois critérios. O primeiro critério foi a seleção de 32 museus da DBpedia, com pelo menos oito trabalhos de arte, em pelo menos cinco movimentos de arte; o segundo foi a seleção dos museus comuns entre as duas bases, totalizando 12 museus.

A avaliação foi realizada através da comparação das listas de museus similares de cada um dos museus, geradas pelo SELEcTor, com as respectivas listas geradas a partir do SmartHistory. A Tabela 4.2 mostra as listas relacionadas ao museu Getty. Para realizar a comparação foi utilizada a métrica NDCG, discutida neste trabalho na Seção 2.1.3.

4.3.2 *The Movie Database (TMDb)*

TMDb foi avaliado em (Colucci et al., 2016). Nesse trabalho foi realizada a avaliação de algoritmos de similaridade item-item no domínio filmes, com base na percepção

Tabela 4.2: Comparação entre SELEcTor e Ground Truth

Fonte: Ruback et al. (2017)

SELEcTor Getty similars	**SmartHistory Getty similars**
dbr:Metropolitan_Museum_of_Art	dbr:Metropolitan_Museum_of_Art
dbr:Louvre	dbr:Vatican_Museums
dbr:Kunsthistorisches_Museum	dbr:Louvre
dbr:Museum_of_Fine_Arts,Boston	dbr:National_Gallery_of_Art
dbr:Vatican_Museums	dbr:Art_Institute_of_Chicago
dbr:Uffizi	dbr:Museum_of_Fine_Arts,Boston
dbr:National_Gallery_of_Art	dbr:Musée_d'Orsay
dbr:Musée_d'Orsay	dbr:Philadelphia_Museum_of_Art
dbr:Philadelphia_Museum_of_Art	dbr:Kunsthistorisches_Museum
dbr:Museum_of_Modern_Art	dbr:Uffizi
dbr:Art_Institute_of_Chicago	dbr:Museum_of_Modern_Art

de similaridade de usuários. Dentre as propostas avaliadas, o TMDb foi a que obteve o melhor desempenho.

O TMDb é um banco de dados de filmes e séries de TV, criado em 2008 e é mantido por sua comunidade[4]. Em seu site, estão disponíveis diversas API's, cuja finalidade é compartilhar informações sobre os filmes armazenados. Através do uso de uma dessas API's, foram obtidos todos os filmes similares, dado um filme, usados para comparação com a abordagem desenvolvida nesse trabalho. As listas de filmes similares foram obtidas por meio da API disponível em:

https://api.themoviedb.org/3/movie/TMDbID/similar.

O uso do TMDb como baseline é justificado pelo fato de ter apresentado melhor resultado, segundo Colucci et al. (2016). No entanto, como descrito em (Colucci et al., 2016) é importante ressaltar, que o algoritmo de similaridade usado pelo TMDb não é conhecido, e foi tratado como uma caixa preta.

Avaliação do TMDb

O TMDb e outros três conjuntos de dados foram usados para realização da avaliação em (Colucci et al., 2016). A qual, foi realizada com base na precisão de cada umas das listas de similaridade item-item, em relação a lista gerada com base na percepção de similaridade dos usuários. Dentre as listas, a que obteve melhor precisão, foi a lista extraída do site TMDb, com uma taxa de 62%, cujo o algoritmo não foi conhecido.

[4]https://www.themoviedb.org/about

4.4 Configuração dos Experimentos

Para avaliar a eficácia da abordagem proposta, foram realizados experimentos com cada coleção de dados, comparando às *baselines* citadas na Seção 4.3. Nas subseções 4.4.1 e 4.4.2, são descritos os processos para realizar os experimentos, considerando as coleções de dados e *baselines*. Inicialmente serão discutidas as configurações comuns realizadas em ambos domínios.

A abordagem proposta foi aplicada em ambos domínios de informação. Primeiramente, foram coletadas as instâncias na DBpedia, relativas as instâncias selecionadas para os testes, juntamente com os respectivos predicados. Considerando que os dados na DBpedia são armazenados em forma de grafos RDF, foram conduzidos experimentos considerando predicados com até dois níveis de distância das instâncias.

Para cada um dos predicados foram estimadas a seguintes métricas: cobertura, frequência do predicado na coleção, ou seja, dada a quantidade de instâncias, qual a frequência do predicado em função do número de instâncias; discriminalidade, diversidade dos resultados retornados para cada predicado e objeto; PF-IPF, frequência do predicado dentro da instância e frequência inversa do predicado na coleção, esta métrica é usada como forma de ponderar os predicados no processo de comparação.

A Tabela 4.3 mostra predicados obtidos na DBpedia no domínio de museus e os respectivos valores para as métricas, discriminalidade, cobertura e IPF, calculados em função da coleção de dados coletados. A métrica PF é estimada em função de cada uma das instâncias, por este motivo não é exibida nesta tabela.

Tabela 4.3: Exemplos de predicados coletados na DBpedia no domínio de museus

Discriminability	Coverage	IPF	Level	Predicate
1,000	1,000	1,000	1	http://xmlns.com/foaf/0.1/name
1,000	1,000	1,000	1	http://xmlns.com/foaf/0.1/homepage
1,000	1,000	1,000	1	http://dbpedia.org/ontology/abstract
1,000	1,000	1,000	1	http://dbpedia.org/ontology/wikiPageRedirects
0,939	1,000	1,000	1	http://dbpedia.org/ontology/numberOfVisitors
0,818	1,000	1,000	1	http://purl.org/dc/terms/subject
0,945	0,938	1,064	1	http://dbpedia.org/ontology/location
1,000	0,625	1,470	1	http://dbpedia.org/ontology/director
0,350	1,000	1,000	2	http://dbpedia.org/ontology/author
0,770	0,313	2,163	2	http://dbpedia.org/ontology/movement

Com o objetivo de obter os melhores limiares de cobertura e discriminalidade dos predicados, estejam estes a um nível de distância das instâncias, ou a dois níveis de distância, foram executadas duas estratégias para avaliar sua qualidade. Na primeira estratégia, é realizada a poda de predicados que estejam abaixo de um limiar mínimo de cobertura e discriminalidade.

A segunda estratégia, elaborada após a execução dos testes com a primeira,

visa cobrir possíveis falhas, como por exemplo, intervalos não testados. Para isso, nessa estratégia, foram definidos limites inferiores e superiores para cada uma das variáveis.

Em ambas estratégias, os experimentos foram realizados com predicados presentes apenas do primeiro nível, ou com predicados presentes no primeiro e no segundo nível, separadamente. Para cada lista de predicados selecionados, foram coletados os nodos objetos, ou vizinhança das instâncias, considerando a representação na forma de grafos RDF. Após a coleta das vizinhanças das instâncias, o processo de comparação foi realizado com base em dois métodos propostos, como pode ser observado nos Algoritmos 1 e 2. Além disso, foram usados dois algoritmos de similaridade de texto, Cosseno e SoftTF-IDF.

O processo de comparação foi realizado considerando dois algoritmos de comparação, além de dois algoritmos de similaridade de textos. Com isso foram obtidos 2x4 resultados para cada iteração.

4.4.1 Experimentos com a coleção de Museus

Para comparar os resultados da abordagem proposta com a *baseline*, na coleção Museus, foram selecionadas instâncias com base em dois critérios, assim como em (Ruback et al., 2017): primeiro, foram selecionados museus com pelo menos oito trabalhos de arte em pelo menos cinco movimentos de arte, esta seleção resultou em 32 museus; segundo, foram selecionados entre os 32 museus, apenas aqueles comuns na DBpedia e no SmartHistory, a lista resultante desta seleção foi composta por 16 instâncias de museus. A Tabela 4.4 mostra a lista de museus resultantes.

Experimentos

Para execução dos experimentos foi necessário reproduzir o SELEcTor, bem como a construção do *Ground Truth*. Foram executados todos os processos descritos em (Ruback et al., 2017) e, além disso, houve a necessidade de contato com os autores do trabalho para maior detalhamento. Toda a reprodução foi feita utilizando a linguagem Java.

4.4.2 Experimentos com a coleção de Filmes

Foi usado como base para os experimentos o conjunto de dados MovieSim-1. Este representa os resultados de filmes similares, com base na percepção de usuários. Este conjunto de dados foi criado nos experimentos realizados em (Colucci et al., 2016), e possui informações de aproximadamente 1416 filmes.

Para realização dos experimentos foi necessário o uso de dados de diferentes

Tabela 4.4: Lista de Museus comuns entre a DBpedia e SmartHistory

Museus
Art_Institute_of_Chicago
J._Paul_Getty_Museum
Kunsthistorisches_Museum
Louvre
Metropolitan_Museum_of_Art
Musée_d'Orsay
Museo_del_Prado
Museum_of_Fine_Arts,_Boston
Museum_of_Modern_Art
National_Gallery_of_Art
National_Gallery
Philadelphia_Museum_of_Art
Tate_Britain
Tate_Modern
Uffizi
Vatican Museums

fontes. Foram usados dados da DBpedia e, fontes usadas nos experimentos realizados em (Colucci et al., 2016). Por este motivo foi necessário identificar os filmes comuns no MovieSim-1, DBpedia e TMDb. Para essa identificação entre os filmes presentes no MovieSim-1 e DBpedia foi utilizada a distância de Levenshtein. Já para filmes presentes em MovieSim-1 e TMDb, foi utilizado o conjunto de dados disponível em MovieLens 20M. As instâncias do MovieSim-1 e MovieLens compartilham o mesmo identificador e as instâncias MovieLens e TMDb também compartilham os mesmos identificadores.

O conjunto de dados MovieLens 20M, contém informações de classificação de usuários sobre filmes e possui mais de 27000 instâncias. Este conjunto de dados é disponibilizado pelo GroupLens[5], laboratório de pesquisa do Departamento de Ciência da Computação e Engenharia da Univerdade de Minnesota.

O conjunto de dados MovieSim-1 compreende informações de 1416 filmes, destes, 176 não possuem correlação com a DBpedia e, outros 18 não possuem correlação com TMDb. Na lista resultante, foram removidos os filmes com menos de oito filmes similares, em pelo menos uma das bases. Esse procedimento foi necessários pois muitos filmes, principalmente retornados pela API do TMDb, tem um número muito pequeno de filmes similares ou mesmo, o que impacta na avaliação da eficácia da abordagem. Com isso restaram 51 filmes com 159 e 1292 predicados no primeiro e no segundo níveis, respectivamente.

[5]https://grouplens.org/datasets/movielens/

Experimentos

Para execução dos experimentos, foi necessário construir as listas ordenadas relativas ao TMDb e também ao MovieSim-1. Os dados do TMDb foram obtidos através de uma de suas API's[6]. Na documentação é possível encontrar duas delas, com funções bastante semelhantes, contudo sem descrição dos algoritmos utilizados. A primeira API, retorna filmes similares, dado um identificador. A documentação informa apenas que serão retornados filmes similares. Contudo, informa que não tem a mesma função de recomendação encontrada em seu site. A segunda API, retorna filmes recomendados, dado um identificador. Esta API retorna os mesmos filmes disponíveis no sistema de recomendação do site, contudo, também não é informado o algoritmo de recomendação utilizado.

Ao pesquisar os *forums* de suporte da API[7] do TMDb, foi encontrada a informação de que a lista de filmes similares é construída com base na lista de metadados e palavras-chave do filmes e a lista de recomendações é construída com base nas avaliações do usuários. Os experimentos foram executados considerando a API que retorna filmes similares, pois com base nas informações do *forum*, utiliza o mesmo princípio, executado nestes trabalho.

4.5 Resultados e Discussão

Nesta Seção, são discutidos os resultados obtidos na avaliação experimental. Nessa discussão, são analisados os resultados nos domínios de museus e de filmes. Como foram realizados experimentos com base em duas estratégias, elas serão discutidas e comparadas.

4.5.1 Domínio de Museus

Considerando a primeira estratégia para definição dos limiares, na qual são definidos apenas limites mínimos, foram definidos os valores 0.5 e 0.9 para cobertura e discriminabilidade, respectivamente, isto para os predicados no primeiro nível no domínio de museus. Para predicados no segundo nível foram definidos os valores 0.9 e 0.6. Estes valores foram definidos de forma empírica, usando o Algoritmo 2 em conjunto com o Soft TF-IDF como função de similaridade.

Considerando a segunda estratégia, na qual são definidos limites mínimos e máximos para poda de predicados, foram definidos os seguintes valores para predicados do primeiro nível, 0.1 e 1.0 para cobertura e 0.8 e 1.0 para discriminalidade. Para o segundo nível foram definidos os valores 0.9 e 1.0 para cobertura e 0.6 e 1.0

[6]https://developers.themoviedb.org/
[7]https://www.themoviedb.org/talk/59efc7fcc3a3680682004dba

para discriminalidade. Em ambos os casos foi usado o Algoritmo 2 em conjunto com o Soft TF-IDF como função de similaridade. Esses resultados apontam que, dentre os predicados mais relevantes, seja do primeiro ou do segundo nível, estão os predicados cujo valor do limite superior de cobertura e discriminalidade é 1.0, ou seja, predicados com 100% de cobertura e que tenham 100% de discriminalidade. Isso corrobora a realização dos testes realizados com a primeira estratégia, na qual são eliminados predicados infrequentes e não discriminatórios.

Com base nos limiares obtidos para o segundo nível, em ambas estratégias experimentadas, é notável que foram obtidos os mesmos valores para cobertura e discriminalidade, afinal [0.9] e [0.6] e [0.9-1.0] e [0.6-1.0] representam os mesmos intervalos.

As Tabelas 4.5 e 4.6 mostram a lista de museus similares ao museu "Louvre"[8], estas listas são resultado de experimentos com a primeira e a segunda estratégia, respectivamente. As primeira e segunda colunas contêm as instâncias similares, classificadas usando os predicados de primeiro e segundo nível, respectivamente. A terceira coluna contém os dados do *SmartHistory* (Ruback et al., 2017). Estes resultados foram obtidos aplicando o Algoritmo 2 com o Soft-TFIDF como função de similaridade. Com base nos resultados presentes nestas tabelas, é perceptível que a lista de itens similares fornecida pela abordagem proposta, é muito similar à lista do *SmartHistory*. Os valores presentes nas tabelas, obtidos com base na primeira e segunda estratégias de teste, divergem apenas nos itens da primeira coluna, os quais são resultado de experimentos com dados dos predicados de primeiro nível. Uma possível justificativa para a diferença entre os resultados, pode estar relacionada ao fato da realização dos experimentos ter sido realizada no *endpoint* DBpedia, o qual não é estático, com intervalo de meses de diferença.

Tabela 4.5: Instâncias Similares a ***Louvre*** - Limiar Mínimo

Predicates Level 1	Predicates Level 2	*SmartHistory*
National_Gallery	National_Gallery_of_Art	National_Gallery
Museo_del_Prado	Metropolitan_Museum_of_Art	National_Gallery_of_Art
Metropolitan_Museum_of_Art	Museo_del_Prado	Metropolitan_Museum_of_Art
Musée_d'Orsay	National_Gallery	Museo_del_Prado
National_Gallery_of_Art	Musée_d'Orsay	Kunsthistorisches_Museum
Philadelphia_Museum_of_Art	Kunsthistorisches_Museum	Museum_of_Fine_Arts,_Boston
Vatican_Museums	Museum_of_Modern_Art	Uffizi
Art_Institute_of_Chicago	Art_Institute_of_Chicago	Vatican_Museums

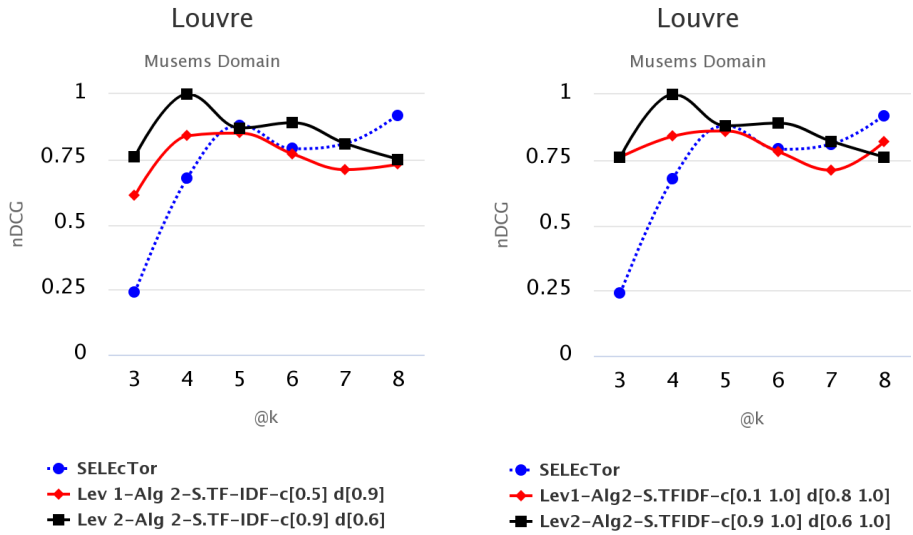
As Figuras 4.1a e 4.1b mostram graficamente os resultados obtidos para a instância Louvre, em comparação com o SELEcTor sob a medida nDCG. Foram obtidos valores nDCG igual a 1,00 e 0,89 para índices k iguais a 4 e 6, respectivamente, usando predicados de segundo nível. Ainda considerando predicados de segundo nível, a

[8]https://www.louvre.fr/en

Tabela 4.6: Instâncias Similares a ***Louvre*** - Limiar Mínimo e Máximo

Predicates Level 1	Predicates Level 2	*SmartHistory*
Metropolitan_Museum_of_Art	National_Gallery_of_Art	National_Gallery
National_Gallery	Metropolitan_Museum_of_Art	National_Gallery_of_Art
Museo_del_Prado	Museo_del_Prado	Metropolitan_Museum_of_Art
Musée_d'Orsay	National_Gallery	Museo_del_Prado
National_Gallery_of_Art	Musée_d'Orsay	Kunsthistorisches_Museum
Vatican_Museums	Kunsthistorisches_Museum	Museum_of_Fine_Arts
Philadelphia_Museum_of_Art	Museum_of_Modern_Art	Uffizi
Kunsthistorisches_Museum	Art_Institute_of_Chicago	Vatican_Museums

abordagem proposta supera o SELEcTor nos índices k de 3 a 7. Considerando a Figura 4.1a, usando predicados de primeiro nível, a abordagem supera o SELEcTor com índices k igual a 3 e 4 e tem resultados próximos em k igual a 5 e 6. A Figura 4.1b mostra uma pequena melhora no resultado com predicados de primeiro nível com o índice k igual a 3.



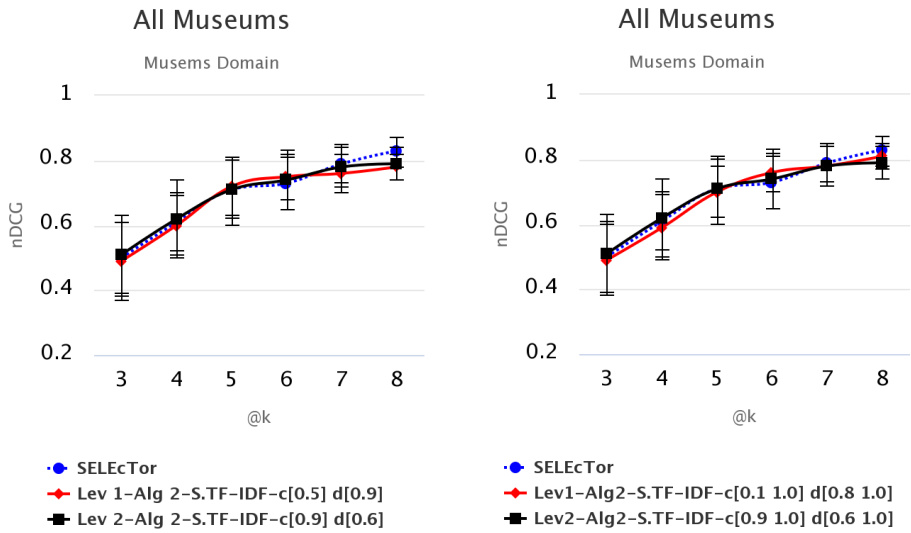
(a) *Limiar mínimo* (b) *Limiar mínimo e máximo.*

Figura 4.1: Museus similares ao Louvre.

A Figuras 4.2a e 4.2b mostram a média dos resultados obtidos no domínio dos museus, sob a medida nDCG. Percebe-se que, para todos os valores de k, a abordagem proposta com uso de predicados de primeiro ou segundo nível está estatisticamente empatada ao SELEcTor, considerando um intervalo de confiança de 90%.

Como dito anteriormente, a abordagem proposta não precisa de um especialista de domínio para escolher os predicados relevantes e ponderá-los. Assim, os resultados da abordagem proposta nesse trabalho são muito competitivos, considerando a *baseline*, sem definição e ponderação manual dos predicados relevantes.

Os resultados atuais foram obtidos após realizar experimentos com vários limiares, conforme descrito em 4.4.



(a) *Limiar mínimo* (b) *Limiar mínimo e máximo.*

Figura 4.2: Média dos resultados no domínio de museus com NDCG.

As Tabelas 4.7 e 4.8 mostram listas ordenadas de predicados selecionados no domínio de museus, conforme os limiares de cobertura e discriminalidade com melhor resultado. Também são mostrados os valores de discriminabilidade (coluna 1), frequência do predicado (coluna 2), IDF (coluna 3) e o nível de cada predicado (coluna 4). Apesar de não ter sido feita uma análise por especialistas de domínio, os resultados obtidos com estes predicados comprovam sua relevância. Além disso é possível perceber importantes predicados na lista, como, por exemplo, *http://purl.org/dc/terms/subject*, citado em (Di Noia et al., 2012), entre outros.

É importante ressaltar que nem todos os predicados exibidos nas tabelas foram considerados, por exemplo, wikiPageID e wikiPageRevisionID têm valores numéricos e, por este motivo, são filtrados no pré-processamento, não sendo utilizados no processos de comparação entre instãncias para obtenção de similaridade.

4.5.2 Domínio de Filmes

As Figuras 4.3a, 4.3b, 4.4a e 4.4b mostram gráficos comparando a abordagem proposta com a *baseline*. As Figuras 4.3a e 4.3b mostram os resultados obtidos com a instância "Iron Man 3", com uso de limiar mínimo e também com uso de limiares mínimo e máximo para poda de predicados, respectivamente. Observa-se que a abordagem proposta, em os ambos gráficos, tem os melhores resultados em k igual a 4 e 5, usando os predicados de primeiro nível, e produz os piores resultados usando os predicados de segundo nível.

Contudo, ao analisar os itens sugeridos como similares na coluna 2 das Tabelas 4.9 e 4.10 , estes aparetam ser muito semelhantes ao filme "Iron Man 3". Nessa tabela,

Tabela 4.7: Predicados relevantes de museus retornados pelo Algoritmo 2, considerando predicados de primeiro nível, com limiares de cobertura entre [0.1-1.0] e discriminalidade entre [0.8-1.0].

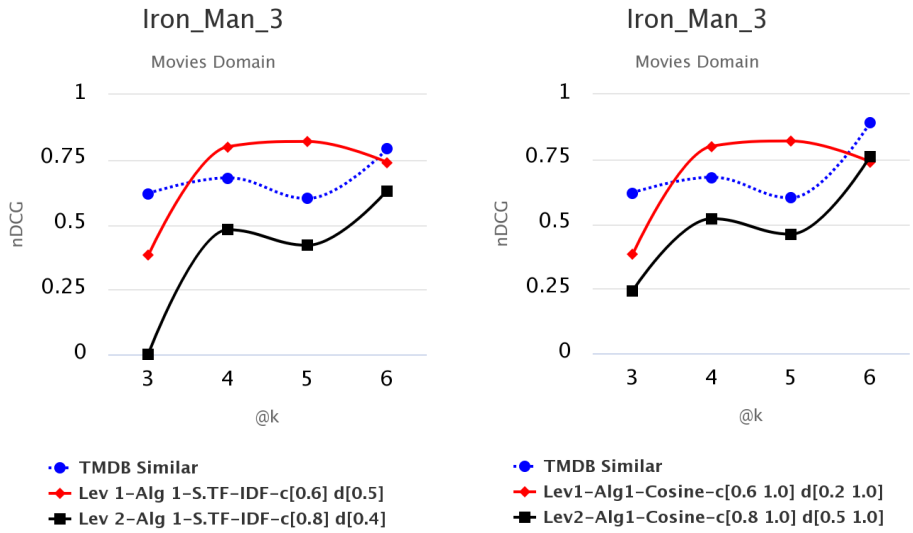
Discriminability	Coverage	IDF	Level	Predicate
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/name
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/homepage
1,000000	1,000	1,000000	1	http://www.georss.org/georss/point
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/abstract
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRedirects
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/depiction
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/primaryTopic
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRevisionID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageExternalLink
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/thumbnail
0,990305	1,000	1,000000	1	http://dbpedia.org/ontology/museum
0,939394	1,000	1,000000	1	http://dbpedia.org/ontology/numberOfVisitors
0,818584	1,000	1,000000	1	http://purl.org/dc/terms/subject
0,945946	0,938	1,064539	1	http://dbpedia.org/ontology/location
1,000000	0,688	1,374693	1	http://dbpedia.org/property/longitude
1,000000	0,688	1,374693	1	http://dbpedia.org/property/latitude
0,937500	0,688	1,374693	1	http://dbpedia.org/ontology/wikiPageDisambiguates
1,000000	0,625	1,470004	1	http://dbpedia.org/property/established
1,000000	0,625	1,470004	1	http://dbpedia.org/ontology/director
1,000000	0,625	1,470004	1	http://dbpedia.org/property/caption
0,944444	0,563	1,575364	1	http://dbpedia.org/property/publictransit
1,000000	0,375	1,980829	1	http://dbpedia.org/property/owner
1,000000	0,375	1,980829	1	http://dbpedia.org/property/mapCaption
0,909091	0,375	1,980829	1	http://dbpedia.org/property/location
1,000000	0,313	2,163151	1	http://dbpedia.org/ontology/knownFor
0,909091	0,313	2,163151	1	http://dbpedia.org/ontology/board
1,000000	0,250	2,386294	1	http://dbpedia.org/ontology/significantBuilding
1,000000	0,250	2,386294	1	http://dbpedia.org/ontology/award
1,000000	0,250	2,386294	1	http://dbpedia.org/property/director
1,000000	0,188	2,673976	1	http://dbpedia.org/property/title
1,000000	0,188	2,673976	1	http://dbpedia.org/property/alt
1,000000	0,188	2,673976	1	http://dbpedia.org/property/width
1,000000	0,188	2,673976	1	http://dbpedia.org/property/imageCaption
1,000000	0,188	2,673976	1	http://dbpedia.org/property/image
1,000000	0,188	2,673976	1	http://dbpedia.org/property/patrons
1,000000	0,125	3,079442	1	http://dbpedia.org/property/collectionSize
1,000000	0,125	3,079442	1	http://dbpedia.org/property/coordinatesDisplay
1,000000	0,125	3,079442	1	http://dbpedia.org/property/space
1,000000	0,125	3,079442	1	http://dbpedia.org/property/designation1Number
1,000000	0,125	3,079442	1	http://dbpedia.org/property/header
1,000000	0,125	3,079442	1	http://dbpedia.org/property/architect
1,000000	0,125	3,079442	1	http://dbpedia.org/property/designation
1,000000	0,125	3,079442	1	http://dbpedia.org/property/name
1,000000	0,125	3,079442	1	http://dbpedia.org/property/designation1Date
1,000000	0,125	3,079442	1	http://dbpedia.org/property/coordinatesRegion
1,000000	0,125	3,079442	1	http://dbpedia.org/property/align
1,000000	0,125	3,079442	1	http://dbpedia.org/property/designation1Offname
1,000000	0,125	3,079442	1	http://dbpedia.org/property/nowAt

Tabela 4.8: Predicados relevantes de museus retornados pelo Algoritmo 2, considerando predicados de segundo nível, com limiares de cobertura entre [0.9-1.0] e discriminalidade entre [0.6-1.0].

Discriminability	Coverage	IDF	Level	Predicate
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/name
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/homepage
1,000000	1,000	1,000000	1	http://www.georss.org/georss/point
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/abstract
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRedirects
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/depiction
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/primaryTopic
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRevisionID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageExternalLink
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/thumbnail
0,990305	1,000	1,000000	1	http://dbpedia.org/ontology/museum
0,939394	1,000	1,000000	1	http://dbpedia.org/ontology/numberOfVisitors
0,818584	1,000	1,000000	1	http://purl.org/dc/terms/subject
0,945946	0,938	1,064539	1	http://dbpedia.org/ontology/location
1,000000	1,000	1,000000	2	http://xmlns.com/foaf/0.1/thumbnail
0,971469	1,000	1,000000	2	http://dbpedia.org/property/imageFile
0,943645	1,000	1,000000	2	http://xmlns.com/foaf/0.1/depiction
0,943510	1,000	1,000000	2	http://dbpedia.org/ontology/thumbnail
0,942857	1,000	1,000000	2	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
0,942857	1,000	1,000000	2	http://xmlns.com/foaf/0.1/primaryTopic
0,933051	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageID
0,933051	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageRevisionID
0,929039	1,000	1,000000	2	http://xmlns.com/foaf/0.1/name
0,912788	1,000	1,000000	2	http://dbpedia.org/ontology/abstract
0,883756	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageDisambiguates
0,809276	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageExternalLink
0,791304	1,000	1,000000	2	http://dbpedia.org/ontology/occupation
0,759718	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageRedirects
0,726496	1,000	1,000000	2	http://xmlns.com/foaf/0.1/homepage
0,696466	1,000	1,000000	2	http://dbpedia.org/property/caption
0,670245	1,000	1,000000	2	http://dbpedia.org/property/widthMetric
0,661631	1,000	1,000000	2	http://dbpedia.org/property/heightMetric
0,617054	1,000	1,000000	2	http://dbpedia.org/property/year
0,964706	0,938	1,064539	2	http://dbpedia.org/property/otherTitle
0,949367	0,938	1,064539	2	http://dbpedia.org/property/video
0,904110	0,938	1,064539	2	http://www.georss.org/georss/point
0,751645	0,938	1,064539	2	http://dbpedia.org/ontology/location
0,709253	0,938	1,064539	2	http://dbpedia.org/ontology/birthPlace

a terceira coluna corresponde aos resultados TMDb. Vale lembrar que, esse *Ground Truth* é resultado do trabalho desenvolvido em (Colucci et al., 2016) e, nesse caso, itens similares retornados por nossa abordagem podem não ter sido avaliados por esse trabalho.

As Figuras 4.4a e 4.4b mostram a média dos resultados no domínio de filmes. A abordagem proposta está estatisticamente empatada com o TMDb Similar (90% de confiança) em todos os valores de k, exceto $k = 3$. O resultado apresentado na Figura 4.4b apresenta uma queda na qualidade média em relação a Figura 4.4a, este fato pode ser justificado por uma alteração na seleção das instâncias de filmes para



(a) *Limiar mínimo* (b) *Limiar mínimo e máximo.*

Figura 4.3: Filmes similares a *Iron Man 3*.

Tabela 4.9: Instâncias similares a ***Iron Man 3*** - Limiar Mínimo

Predicates Level 1	Predicates Level 2	Users List
Captain_America:_The_Winter_Soldier	Iron_Man_(1931_film)	Captain_America:_The_First_Avenger
Captain_America:_The_First_Avenger	Captain_America:_The_Winter_Soldier	Man_of_Steel_(film)
Guardians_of_the_Galaxy_(film)	Guardians_of_the_Galaxy_(film)	Thor_(film)
Man_of_Steel_(film)	Captain_America:_The_First_Avenger	Captain_America:_The_Winter_Soldier
Thor_(film)	X-Men:_Days_of_Future_Past	The_Avengers_(1998_film)
X-Men:_Days_of_Future_Past	Batman_Begins	Iron_Man_(1931_film)

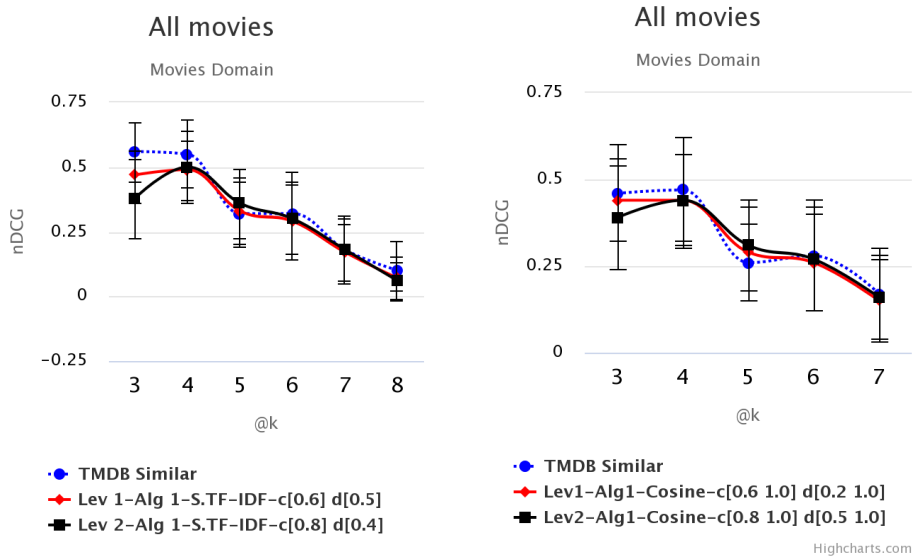
Tabela 4.10: Instâncias similares a ***Iron Man 3*** - Limiar Mínimo e Máximo

Predicates Level 1	Predicates Level 2	Users List
Captain_America:_The_Winter_Soldier	Iron_Man_(1931_film)	Captain_America:_The_First_Avenger
Captain_America:_The_First_Avenger	Captain_America:_The_Winter_Soldier	Man_of_Steel_(film)
Guardians_of_the_Galaxy_(film)	Captain_America:_The_First_Avenger	Thor_(film)
Man_of_Steel_(film)	Guardians_of_the_Galaxy_(film)	Captain_America:_The_Winter_Soldier
Thor_(film)	X-Men:_Days_of_Future_Past	The_Avengers_(1998_film)
X-Men:_Days_of_Future_Past	Man_of_Steel_(film)	Iron_Man_(1931_film)

este experimento, neste caso foram selecionados filmes com no mínimo sete filmes similares em MovieSim-1 ou TMDb, desta maneira a quantidade de filmes na base de teste aumento e possibilitou fez com que alguns filmes estivessem presentes em apenas uma das bases.

Como exemplo de predicados selecionados por nossa abordagem, as Tabelas 4.11 e 4.12 mostram uma lista ordenada de predicados selecionados no domínio dos filmes. Nestas tabelas, também são mostrados os valores de discriminabilidade (coluna 2), frequência do predicado (coluna 3), IDF (coluna 4) e o nível de cada predicado. Vale ressaltar que, não foram considerados todos os predicados exibidos na tabela, por exemplo, wikiPageID e wikiPageRevisionID têm valores numéricos e são filtrados no pré-processamento.

Alguns predicados têm um significado especial e podem afetar os resultados,



(a) *Limiar mínimo*

(b) *Limiar mínimo e máximo.*

Figura 4.4: Média dos resultados no domínio de filmes com NDCG.

como "seeAlso". Ele é usado para indicar que um recurso pode fornecer informações adicionais sobre um recurso específico. No entanto, a abordagem proposta não pode distinguir esses predicados dos demais, devido a forma de remoção baseada nos limiares.

4.5.3 Análise dos parâmetros

Com objetivo de avaliar a influência dos limiares nos resultados foi realizada uma análise de sensibilidade a estes parâmetros, conforme descrito nesta Seção.

Considerando a primeira estratégia, foram avaliadas todas as combinações de limiares de cobertura e de discriminalidade. Os testes iniciaram com a seleção dos predicados com mais de 90% de cobertura e mais de 90% de discriminalidade e a cada iteração, o percentual da variável discriminalidade foi reduzido na ordem de 10%, até o limite inferior de 10%. Ao alcançar este limite, o percentual da variável cobertura foi decrementado em 10% e foram testadas todas as combinações com a variável discriminalidade.

A ideia desta estratégia é eliminar predicados que não sejam suficientemente relevantes, seja em função de cobertura ou de discriminalidade. O limite inferior de 10% foi considerado com base na ideia de que, predicados com menos de 10% de cobertura ou discriminalidade não trariam bons resultados além de consumir um tempo computacional muito grande, podendo inviabilizar o teste. Mesmo antes da realização deste experimento, havia a expectativa, com base dedutiva, de que os melhores resultados seriam obtidos com os limiares mais altos, o que pode ser comprovado, conforme análise dos resultados na Seção 4.5.

Tabela 4.11: Predicados relevantes de filmes retornados pelo Algoritmo 1, considerando predicados de primeiro nível, com limiares de cobertura entre [0.6-1.0] e discriminalidade entre [0.2-1.0].

Discriminability	Coverage	IDF	Level	Predicate
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/primaryTopic
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRevisionID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/abstract
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
0,938776	1,000	1,000000	1	http://xmlns.com/foaf/0.1/name
0,548077	1,000	1,000000	1	http://purl.org/dc/terms/subject
0,886364	0,977	1,023530	1	http://dbpedia.org/ontology/Work/runtime
0,886364	0,977	1,023530	1	http://dbpedia.org/ontology/runtime
0,865385	0,977	1,023530	1	http://dbpedia.org/ontology/director
0,446809	0,930	1,072321	1	http://dbpedia.org/ontology/distributor
1,000000	0,907	1,097638	1	http://dbpedia.org/ontology/wikiPageRedirects
0,987097	0,884	1,123614	1	http://dbpedia.org/ontology/wikiPageExternalLink
0,884615	0,860	1,150282	1	http://dbpedia.org/ontology/wikiPageDisambiguates
1,000000	0,837	1,177681	1	http://dbpedia.org/ontology/gross
0,810811	0,837	1,177681	1	http://dbpedia.org/ontology/budget
0,657895	0,814	1,205852	1	http://dbpedia.org/ontology/musicComposer
0,868545	0,767	1,264693	1	http://dbpedia.org/property/title
0,771429	0,744	1,295464	1	http://dbpedia.org/ontology/cinematography
0,218750	0,744	1,295464	1	http://purl.org/linguistics/gold/hypernym
0,225806	0,674	1,393904	1	http://dbpedia.org/property/country
0,718750	0,651	1,428996	1	http://dbpedia.org/ontology/editing
0,952381	0,628	1,465363	1	http://dbpedia.org/ontology/writer
0,837209	0,628	1,465363	1	http://dbpedia.org/ontology/producer
0,923077	0,605	1,503104	1	http://xmlns.com/foaf/0.1/homepage

Considerando a segunda estratégia, o experimento iniciou com o intervalo dos limiares de 10% e 20%, tanto para a cobertura quanto para a discriminalidade. Após a primeira iteração, a cobertura se manteve e o intervalo da discriminalidade foi incrementado em 10%, passando a 10% e 30%, desta maneira sucessivamente até o limite superior de 100%. Após este limite, os percentuais de cobertura foram mantidos e foi ajustado o limite inferior da discriminalidade para 20% e superior para 30%. Estes testes foram sucessivos até a execução de todas as combinações possíveis de intervalos de cobertura e discriminalidade.

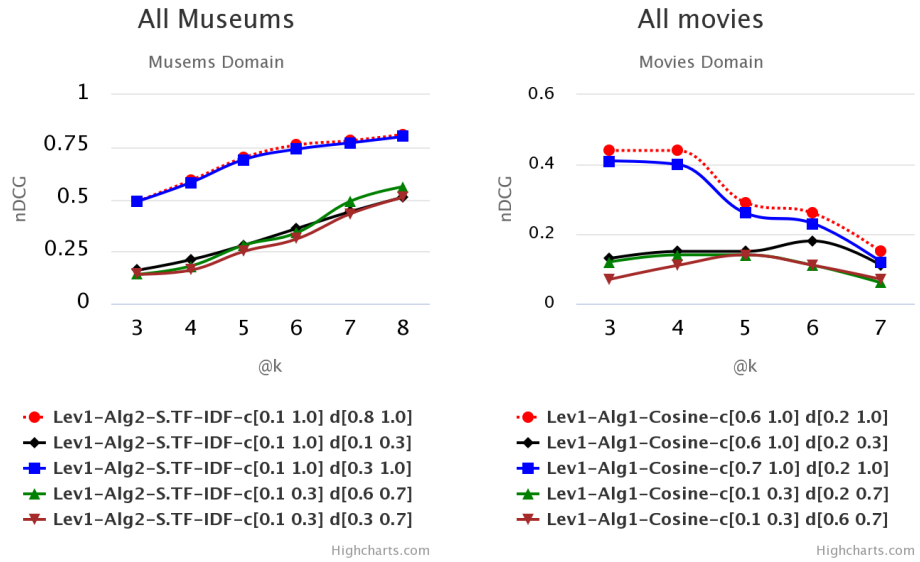
As Figuras 4.5a e 4.5b, mostram os resultados obtidos com diferentes valores para os limiares dos intervalos de cobertura e discriminalidade, considerando predicados de primeiro nível, nos domínios de museus e de filmes. Ao analisar ambos os gráficos, percebe-se o impacto, nos resultados, da remoção de predicados com valores de cobertura e discriminalidade próximos aos 100%, ou seja, limiares superiores dos intervalos, tanto de cobertura quanto de discriminalidade, cujos valores são inferiores a 100%. Na Figura 4.5a, o melhor resultado foi obtido com os limiares de cobertura no intervalo [0.1-1.0] e de discriminalidade [0.8-1.0], muito próximo ao segundo melhor resultado, cuja diferença está no aumento do intervalo de discriminalidade para [0.3-1.0]. Resultados bem superiores ao pior registrado, cujos valores de limiares são

Tabela 4.12: Predicados relevantes de filmes retornados pelo Algoritmo 1, considerando predicados de segundo nível, com limiares de cobertura entre [0.8-1.0] e discriminalidade entre [0.5-1.0].

Discriminability	Coverage	IDF	Level	Predicate
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/primaryTopic
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/wikiPageRevisionID
1,000000	1,000	1,000000	1	http://dbpedia.org/ontology/abstract
1,000000	1,000	1,000000	1	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
0,938776	1,000	1,000000	1	http://xmlns.com/foaf/0.1/name
0,548077	1,000	1,000000	1	http://purl.org/dc/terms/subject
0,886364	0,977	1,023530	1	http://dbpedia.org/ontology/Work/runtime
0,886364	0,977	1,023530	1	http://dbpedia.org/ontology/runtime
0,865385	0,977	1,023530	1	http://dbpedia.org/ontology/director
1,000000	0,907	1,097638	1	http://dbpedia.org/ontology/wikiPageRedirects
0,987097	0,884	1,123614	1	http://dbpedia.org/ontology/wikiPageExternalLink
0,884615	0,860	1,150282	1	http://dbpedia.org/ontology/wikiPageDisambiguates
1,000000	0,837	1,177681	1	http://dbpedia.org/ontology/gross
0,810811	0,837	1,177681	1	http://dbpedia.org/ontology/budget
0,657895	0,814	1,205852	1	http://dbpedia.org/ontology/musicComposer
0,989355	1,000	1,000000	2	http://dbpedia.org/ontology/birthPlace
0,832972	1,000	1,000000	2	http://xmlns.com/foaf/0.1/primaryTopic
0,832972	1,000	1,000000	2	http://xmlns.com/foaf/0.1/isPrimaryTopicOf
0,817938	1,000	1,000000	2	http://xmlns.com/foaf/0.1/name
0,783476	1,000	1,000000	2	http://dbpedia.org/ontology/birthDate
0,781348	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageDisambiguates
0,775956	1,000	1,000000	2	http://xmlns.com/foaf/0.1/surname
0,744526	1,000	1,000000	2	http://dbpedia.org/ontology/thumbnail
0,744526	1,000	1,000000	2	http://xmlns.com/foaf/0.1/depiction
0,721096	1,000	1,000000	2	http://dbpedia.org/ontology/abstract
0,704403	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageID
0,704403	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageRevisionID
0,686213	1,000	1,000000	2	http://dbpedia.org/ontology/writer
0,542178	1,000	1,000000	2	http://dbpedia.org/ontology/wikiPageRedirects
0,790451	0,977	1,023530	2	http://dbpedia.org/property/caption
0,678154	0,977	1,023530	2	http://dbpedia.org/ontology/director
0,569005	0,977	1,023530	2	http://dbpedia.org/ontology/producer
0,520000	0,977	1,023530	2	http://xmlns.com/foaf/0.1/givenName
0,591892	0,953	1,047628	2	http://dbpedia.org/ontology/creator
0,815789	0,930	1,072321	2	http://dbpedia.org/ontology/birthName
0,781298	0,907	1,097638	2	http://dbpedia.org/ontology/starring
0,633588	0,907	1,097638	2	http://xmlns.com/foaf/0.1/homepage
0,503546	0,837	1,177681	2	http://dbpedia.org/ontology/foundedBy
0,735656	0,814	1,205852	2	http://dbpedia.org/ontology/company
0,687783	0,814	1,205852	2	http://dbpedia.org/ontology/composer
0,571429	0,814	1,205852	2	http://dbpedia.org/ontology/award
0,513889	0,814	1,205852	2	http://dbpedia.org/ontology/spouse

[0.1-0.3] e [0.3-0.7], para cobertura e discriminalidade, respectivamente. Todos os resultados mostrados na Figura 4.5a foram obtido com uso do Algoritmo 2 com uso do SoftTF-IDF.

A Figura 4.5b, mostra os resultados da variação dos limiares no domínio de museus. O melhor resultado foi obtido com limiares [0.6-1.0] para cobertura e [0.2-1.0] para discriminalidade e, o pior resultado foi obtido com os limiares [0.1-0.3] e [0.6-0.7], também para cobertura e discriminalidade.



(a) *Comparação da variação de limiares mínimos e máximos no domínio de museus*

(b) *Comparação da variação de limiares mínimos e máximos no domínio de filmes.*

Figura 4.5: Sensibilidade do resultado a variação dos limiares

Capítulo 5

Conclusões e Trabalhos Futuros

Nesse trabalho, foi proposta uma abordagem para estimar a similaridade entre itens usando instâncias da LOD e seus relacionamentos semânticos, sem identificação prévia das características mais relevantes.

Com o objetivo de identificar de maneira automática quais os predicados mais relevantes, foram utilizadas duas métricas, a cobertura e a discriminalidade. A intuição com a cobertura é determinar a relevância dos predicados em função da sua frequência na coleção, com isso é possível eliminar predicados pouco frequentes, que tem pouca influência no processo de estimativa de similaridade, podendo até mesmo aumentar o nível de ruído. Já a discriminalidade determina quão únicos são os valores dos objetos obtidos pelo predicado. A intuição com esta métrica é eliminar predicados não suficientemente discriminantes. Para realizar a ponderação dos predicados, foi adotada uma solução inspirada no TF-IDF, realizando a ponderação dos predicados com base em sua frequência nas instâncias e também na coleção.

Para avaliar experimentalmente a abordagem proposta foram utilizadas duas coleções, contendo instâncias de museus e filmes coletados na DBpedia. Os resultados mostraram que o método é capaz de calcular a similaridade entre instâncias e gerar listas de itens, potencialmente úteis para um SR baseado em conteúdo. Além disso, teve eficácia semelhante às *baselines* sem intervenção de especialistas de domínio.

Como trabalhos futuros, pretende-se empregar técnicas de mineração de texto e aprendizado de máquina para estimar a similaridade entre itens. Além disso, pretende-se investigar quais os efeitos do uso de *word embeddings* nos algoritmos de comparação de instâncias da abordagem. Outro aspecto a ser considerado é a validação da abordagem proposta, através da incorporação a um Sistema de Recomendação. Podendo ser incorporada a recomendadores híbridos, combinando técnicas de recomendação baseadas em conteúdo (ainda explorando o LOD) com técnicas de filtragem colaborativa.

Referências Bibliográficas

Alfarhood, S. D. (2017), Exploiting Semantic Distance in Linked Open Data for Recommendation, PhD thesis, University of Arkansas, Fayetteville, https://scholarworks.uark.edu/etd/1882.

Berners-Lee, T. (2006), 'Linked data - design issues'. Acessado em: 27/03/2018. **URL:** *https://www.w3.org/DesignIssues/LinkedData.html*

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.

Burke, R. (2007), *Hybrid Web Recommender Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 377–408.

Colucci, L., Doshi, P., Lee, K.-L., Liang, J., Lin, Y., Vashishtha, I., Zhang, J. and Jude, A. (2016), Evaluating item-item similarity algorithms for movies, *in* 'Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems', CHI EA '16, San Jose, California, USA, pp. 2141–2147.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D. and Zanker, M. (2012), Linked open data to support content-based recommender systems, *in* 'Proceedings of the 8th International Conference on Semantic Systems', I-SEMANTICS '12, Graz, Austria, pp. 1–8.

Di Noia, T. and Ostuni, V. C. (2015), *Recommender Systems and Linked Open Data*, Springer International Publishing, Cham, Switzerland, pp. 88–113.

Di Noia, T., Ostuni, V. C., Tomeo, P. and Sciascio, E. D. (2016), 'Sprank: Semantic path-based ranking for top-n recommendations using linked open data', *ACM Transactions on Intelligent Systems and Technology (ACM TIST)* **8**(1), 1–34.

Figueroa, C., Vagliano, I., Rocha, O. R. and Morisio, M. (2015), 'A systematic literature review of linked data-based recommender systems', *Concurr. Comput. : Pract. Exper.* **27**(17), 4659–4684.

Heath, T. and Bizer, C. (2011), 'Linked data: Evolving the web into a global data space', *Synthesis lectures on the semantic web: theory and technology* **1**(1), 1–136.

Järvelin, K. and Kekäläinen, J. (2002), 'Cumulated gain-based evaluation of ir techniques', *ACM Trans. Inf. Syst.* **20**(4), 422–446.

Leal, J. P., Rodrigues, V. and Queirós, R. (2012), Computing semantic relatedness using dbpedia, *in* '1st Symposium on Languages, Applications and Technologies, SLATE 2012, Braga, Portugal, June 21-22, 2012', pp. 133–147.

Leng, H., De La Cruz Paulino, C., Haider, M., Lu, R., Zhou, Z., Mengshoel, O., Brodin, P.-E., Forgeat, J. and Jude, A. (2018), Finding similar movies: dataset, tools, and methods, *in* 'Proceedings of the 8th International Conference on Semantic Systems', WSCG'2018, Plzen, Czech Republic, pp. 115–124.

Manning, C. D., Raghavan, P. and Schütze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA.

McCrae, J. P. (2018), 'The linked open data cloud'. Acessado em: 02/05/2018. **URL:** *http://lod-cloud.net/*

Milne, D. and Witten, I. H. (2008), An effective, low-cost measure of semantic relatedness obtained from wikipedia links, *in* 'in Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI', Press, Chicago, Illinois, USA, pp. 25–30.

Mirizzi, R., Di Noia, T., Ragone, A., Ostuni, V. and Di Sciascio, E. (2012), Movie recommendation with dbpedia, Vol. 835, Bari, Italy, pp. 101–112.

Ricci, F., Rokach, L. and Shapira, B. (2015), *Recommender Systems Handbook*, Vol. 54, 2nd edn, Springer Publishing Company, Incorporated, New York, NY, USA.

Ruback, L., Casanova, M. A., Renso, C. and Lucchese, C. (2017), Selector: Discovering similar entities on linked data by ranking their features, *in* '2017 IEEE 11th International Conference on Semantic Computing (ICSC)', San Diego, CA, USA, pp. 117–124.

Santos, A. P. d. (2017), Sistema de recomendação baseado em agrupamento usando propagação de afinidades, Master's thesis, Universidade Federal do Amazonas, https://tede.ufam.edu.br/handle/tede/6258.

Song, D., Luo, Y. and Heflin, J. (2017), 'Linking heterogeneous data in the semantic web using scalable and domain-independent candidate selection', *IEEE Transactions on Knowledge and Data Engineering* **29**(1), 143–156.

Srinivasan, U. and Mani, C. (2018), 'Diversity-ensured semantic movie recommendation by applying linked open data', *International Journal of Intelligent Engineering and Systems* **11**, 275–286.

Webber, W., Moffat, A. and Zobel, J. (2010), 'A similarity measure for indefinite rankings', *ACM Transactions on Information Systems* **28**(4), 20:1–20:38. **URL:** *http://doi.acm.org/10.1145/1852102.1852106*

Yu, L. (2011), *A Developer's Guide to the Semantic Web*, Springer-Verlag Berlin Heidelberg.