Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uma Abordagem baseada em Linked Open Data para Diversificação de Recomendações

Nailson Boaz Costa Leite

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Leandro Balby Marinho
Carlos Eduardo Santos Pires

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

L533a Leite, Nailson Boaz Costa.

Uma abordagem baseada em Linked Open Data para diversificações de recomendações / Nailson Boaz Costa Leite. – Campina Grande, 2016.

73 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2015.

"Orientação: Prof. Dr. Leandro Balby Marinho, Prof. Dr. Carlos Eduardo Santos Pires.

Referências.

1. Sistemas de Recomendação. 2. Linked Open Data. 3. Recommender Systems. 4. Diversity. I. Marinho, Leandro Balby. II. Pires, Carlos Eduardo Santos. III. Título.

CDU 004.775(043)

'UMA ABORDAGEM BASEADA EM LINKED OPEN DATA PARA DIVERSIFICAÇÃO DE RECOMENDAÇÕES"

NAILSON BOAZ COSTA LEITE

DISSERTAÇÃO APROVADA EM 30/11/2015

LEANDRO BALBY MARINHO, Dr., UFCG
Orientador(a)

CARLOS EDUARDO SANTOS PIRES, Dr., UFCG
Orientador(a)

NAZARENO FERREIRA DE ANDRADE, D.Sc, UFCG
Examinador(a)

RODRYGO LUIS TEODORO SANTOS, Dr., UFMG
Examinador(a)

CAMPINA GRANDE - PB

Resumo

A diversidade é um conceito importante em Sistemas de Recomendação (SR), uma vez que recomendações diversificadas podem ajudar o usuário a encontrar itens mais interessantes e relevantes. Diversidade em SR pode ser principalmente alcançada considerando dois aspectos: (i) a semelhança entre os itens da lista de recomendações, sob a suposição de que quanto mais dissimilares os itens na lista, mais diversa ela é; e (ii) a cobertura dos atributos dos itens em um determinado âmbito, ou seja, quanto mais atributos cobertos na lista de recomendações (por exemplo, os gêneros na recomendação de artistas musicais), mais diversa é a lista. Dada a dificuldade de acessar ou extrair atributos dos itens de um SR, o aspecto (i) continua sendo um tópico predominante na literatura. No entanto, devido a Web Semântica e a iniciativa Linked Open Data (LOD), vários atributos comumente encontrados em diferentes domínios de recomendação (por exemplo, filmes, livros e músicas) tornaram-se disponíveis publicamente em bases de dados RDF, conectadas entre si na chamada LOD Cloud. Neste trabalho, propomos uma nova abordagem para a diversificação em SR, a qual explora as relações semânticas entre os atributos dos itens, extraídos de repositórios de LOD, bem como as suas várias dimensões de conteúdo. Outra contribuição deste trabalho é lidar com o típico *trade-off* entre a precisão e a diversidade das recomendações por meio da inferência do grau desejado de diversificação (utilizado como parâmetro do diversificador) diretamente do perfil do usuário. Realizou-se uma avaliação da abordagem proposta em dados reais de usuários coletados do Last.FM, uma rede social e um *scrobbler*, que armazena o histórico on-line das músicas escutadas pelo usuário. Mostrou-se que a abordagem proposta complementa e supera os trabalhos relacionados em métricas de diversidade e acurácia, por ser capaz de diversificar de modo ajustado ao perfil do usuário e de descobrir novas relações entre os itens a partir de seus atributos e suas relações semânticas, melhorando assim o *trade-off* entre a precisão e a diversidade da lista de recomendação em comparação com os demais algoritmos utilizados.

Abstract

Diversity is an important concept in Recommender Systems (RS) since diversified recommendations can help the users to find more interesting and relevant items. Diversity in RS is mainly achieved considering two aspects: (i) the similarity between the items in the recommendation list, under the assumption that the more dissimilar, more diverse is the list; and (ii) the coverage of items' attributes, i.e., the more attributes covered in the recommendation list (e.g., musical genres on musical artists recommendation), the more diverse is the list. Given that it is not always easy to access or extract the attributes of the items of a RS, (i) is still the predominant approach exploited in the literature. However, thanks to the Semantic Web and Linked Open Data (LOD) initiatives, several attributes commonly found in many recommendation domains (e.g. movies, books and music) are now publicly available in RDF databases, connected to each other in the LOD Cloud. In this work, we propose and implement a new approach for promoting diversification in RS that exploits the semantic relationships between item's attributes, both extracted from LOD repositories, found in LOD repositories as well as it several content dimensions. Another contribution of this work is that we tackle the typical trade-off between accurate and diversified recommendations by inferring the degree of diversification (used as a diversifier parameter) directly from the user profile. We conduct a thorough evaluation of our approach on real data collected from Last.FM, a social, online radio station and scrobbler, which stores online history of user-heard songs. It has been shown that the proposed approach complements and exceeds the work related to diversity and accuracy metrics, the algorithm is able to diversify the set of itens fitted to the user profile and discover new relationships between the items from their attributes and their semantic relations, thereby enhancing the trade-off between accuracy and diversity of recommendation list compared with other algorithms used.

Agradecimentos

Agradeço primeiramente a minha família, principalmente aos meus pais, Nírodes e Deusanilde, que foram os meus primeiros e mais importantes educadores, por sempre zelar por minha felicidade e bem estar e por serem os meus grandes e incansáveis incentivadores em todas as minhas conquistas e superações.

Agradeço a Giulia, uma mulher espetacular, linda, inteligente e bem humorada, que apesar da distância sempre me dá forças para encontrar a paz e conforto, nos tempos em que me sinto mais vulnerável e inseguro.

Aos meus grandes amigos e companheiros da UFCG, onde passei grande parte da minha estadia em Campina Grande, em especial, Caio, Gildo e Gustavo, que me auxiliaram com seus valiosos conselhos e ponderações além da descontração e alegria. Aos amigos que fiz no decorrer dessa dura jornada, que foi o mestrado. Nessa fase da minha vida fiz grandes e valorosos amigos que me apoiaram, me acompanharam e me ensinaram lições que guardarei para sempre.

Aos professores, em especial Nazareno Andrade, a quem devo muito. Sempre empenhados em oferecer o melhor para os seus alunos, para mim foi uma honra tê-los como tutores. Aos meus orientadores Leandro Balby e Carlos Eduardo, a quem tenho um enorme apreço e admiração, pela sua paciência, incentivo e orientação, que me guiaram no desenvolvimento desse trabalho.

A todos um muito obrigado, estou em eterna dívida com vocês.

Conteúdo

Lista de Símbolos

EILD - *Expected Intra-List Diversity*

AUC - *Area Under the ROC Curve*

BPR - *Bayesian Personalized Ranking*

HTML - *HyperText Markup Language*

IA-Select - *Intent Aware Select*

ILD - *Intra-List Diversity*

kNN - *k-Nearest Neighbour*

LOD - *Linked Open Data*

MAE - *Mean Average Error*

MAP - *Mean Average Precision*

MMR - *Maximal Marginal Relevance*

NDCG - *Normalized Discounted Cumulative Gain*

RDF - *Resource Description Framework*

RI - *Recuperação da Informação*

RMSE - *Root Mean Square Error*

SR - *Sistemas de Recomendação*

URI - *Uniform Resource Identifier*

XML - *eXtensible Markup Language*

Lista de Figuras

Lista de Tabelas

Lista de Códigos Fonte

Capítulo 1

Introdução

Este capítulo apresenta o contexto e a motivação da pesquisa descrita nesta dissertação. Primeiro, contextualizam-se os principais aspectos e desafios enfrentados pelos Sistemas de Recomendação (SR). Em seguida discutem-se a motivação do trabalho e a necessidade de recomendar listas diversificadas nos SR. Dado esse contexto, apresenta-se a definição do problema de pesquisa, oferecendo posteriormente um pequeno vislumbre de como o problema foi abordado neste trabalho.

As seções posteriores tratam da fundamentação teórica em que o trabalho está baseado e tem como foco a diversidade em SR e LOD. Nessa perspectiva, a sua estruturação compreende três partes principais: a primeira inclui as terminologias utilizadas no decorrer deste trabalho, a segunda descreve os conceitos e definições estabelecidas e aceitos pela comunidade científica sobre a diversidade em SR, além das métricas e os algoritmos do estado-da-arte desenvolvidos. A terceira parte apresenta o surgimento, os conceitos e tecnologias do LOD, utilizadas para um melhor entendimento do algoritmo LOD-Diversification proposto neste trabalho. Por fim é apresentada uma descrição da organização dos demais capítulos que compõem esta dissertação.

1.1 Motivação

Devido à crescente quantidade de dados disponíveis na Web, são necessárias ferramentas cada vez mais avançadas que auxiliem o usuário a encontrar informações relevantes. Essa necessidade vai além das ferramentas de Recuperação de Informação (RI) conhecidas atu-

almente, que satisfazem uma necessidade de informação pontual e específica. Muitas vezes a perspectiva do usuário é completamente invertida: é necessário destacar para o usuário o resultado antes mesmo do surgimento de uma necessidade de informação. Essa mudança está cada vez mais presente em sites populares de comércio eletrônico, *scrobblers*[1] de música e sistemas de *streaming* de mídias digitais, que perceberam a possibilidade de aumentar e diversificar suas vendas indicando para o cliente os itens que ele poderia estar interessado em consumir. Empresas como a Amazon[2], o Netflix[3] e o Spotify[4] apresentam esse tipo de serviço eletrônico disponibilizado na internet.

Os Sistemas de Recomendação (SR) [28] surgem com esse objetivo, ou seja, o de sugerir itens relevantes para usuários, selecionados a partir de um grande espaço de opções. Recomendações compreendem vários processos de tomada de decisão como, por exemplo, quais itens comprar, que tipo de músicas ouvir e até escolhas mais importantes, como a recomendação de um emprego ou um investimento na bolsa de valores.

A avaliação das recomendações geradas por diferentes SR continua sendo um tópico fundamental para a área de pesquisa, onde percebe-se que a grande maioria dos trabalhos de pesquisa existentes estão focados apenas na acurácia para medir a eficiência da recomendação, ou seja, quão diferentes os itens recomendados são dos itens realmente consumidos pelo usuário. No entanto, tópicos como novidade, diversidade e serendipidade têm recebido uma atenção crescente entre os pesquisadores [42; 41; 37; 35]. A diversidade em SR geralmente se aplica a um conjunto de itens e está relacionada a quão diferentes os itens são uns aos outros. Dada uma lista de recomendações ordenada por relevância (também conhecida como recomendação *Top-N*, onde N é um número de itens recomendados), a diversidade é geralmente calculada com base na dissimilaridade entre os itens da lista de recomendações [42]. A novidade, por sua vez, está relacionada à diferença entre a lista de recomendações gerada e o perfil do usuário, ou seja, a novidade mensura quão diferente (ou nova) a recomendação é em relação aos itens que o usuário consumiu anteriormente [11].

[1] Serviço que permite registrar o histórico de músicas escutadas pelos usuários em um sistema de *streaming* ou *player de música*.

[2] http://www.amazon.com/

[3] https://www.netflix.com/

[4] https://www.spotify.com/us/

Os pesquisadores [19; 40; 39] observaram que, em cenários reais de recomendação, somente o uso da acurácia não é suficiente para avaliar a efetividade de SR, pois recomendações acuradas não são necessariamente úteis. Embora as métricas de acurácia representem uma face muito importante de utilidade, há traços da satisfação do usuário que ela é incapaz de capturar. Segundo Herlocker et al. [19], o real valor de uma recomendação está em sugerir objetos que os usuários não descobririam por si mesmos. Um dos riscos de aplicações que utilizam somente a similaridade entre usuários ou itens para recomendar listas acuradas é que, quanto mais *feedback* for inserido no SR, mais os usuários estarão expostos a itens populares (itens que são muito consumidos por usuários), enquanto um conjunto de itens que poderiam ser relevantes serão negligenciados [41; 24].

Figura 1.1: Álbuns recomendados pela Amazon a partir da consulta do álbum *The Wall*, do grupo músical Pink Floyd. Busca realizada em setembro de 2015.

Pesquisadores acreditam que a chave para essa questão seja a diversidade, porém quando um SR se concentra muito na diversidade, em detrimento da acurácia, a acurácia do recomendador pode ser prejudicada [19; 41]. Nesse domínio, surge a necessidade de gerar listas com recomendações diversas sem que isso ocasione, necessariamente, uma grande perda de acurácia. Quando utilizamos um sistema de recomendação como, por exemplo, os utilizados em lojas de comércio eletrônico ou *scrobblers* de música, podemos nos deparar com a situação descrita na Figura 1.1, onde o algoritmo de recomendação da Amazon[5] recomenda itens em uma lista composta quase que puramente de novos álbuns do *Pink Floyd*. Para usuários que consumiram álbuns do *Pink Floyd* no passado, a recomendação, embora relevante, não

[5]http://www.amazon.com/

é útil nos seguintes aspectos:

- Não ajuda o usuário a descobrir novos artistas musicais;
- Não explora os diferentes estilos, gêneros musicais ou álbuns de diferentes artistas. As recomendações ficam restritas aos gêneros mais populares e mais conhecidos do usuário;
- Não surpreende o usuário com álbuns de diferentes países, culturas ou épocas.

Nesse contexto, desenvolvimentos recentes na área de LOD oferecem novas tecnologias desenvolvidas no contexto da Web Semântica (veja Seção 1.7) e tem possibilitado a definição de novas estratégias para explorar diversos tipos de relacionamentos entre os itens de recomendação e seus atributos.

Por exemplo, os artistas *David Gilmour* e *Beyoncé* estão relacionados, pois ambos são artistas musicais que gravaram álbuns na *Columbia Records* e possuem os prêmios *Ivor Novello* e *Brit Awards*. Diariamente, mais e mais dados semânticos são publicados em grandes bases de dados na Web, seguindo os princípios do *Linked Open Data* (LOD) [5]. O projeto LOD tem o intuito de difundir boas práticas para publicar e interligar dados abertos de forma estruturada, assim como padronizar tecnologias e mecanismos que facilitam o processamento, o compartilhamento e o reuso de dados. Um exemplo de uma base de dados LOD é o projeto DBpedia [3], que possui milhares de tuplas no formato RDF [20] em diversos domínios de conhecimento, extraídos da base de dados da Wikipédia[6] e disponibilizados publicamente. Os dados e relacionamentos extraídos das bases de LOD são utilizados para fornecer as informações contextuais necessárias à construção do algoritmo de diversificação LOD-Diversification proposto neste trabalho.

1.2 Definição do Problema

Com o objetivo de melhorar a utilidade das listas de recomendação, foi desenvolvido um algoritmo que diversifica listas de recomendações baseado em grafos de relações semânticas entre os atributos dos itens. Assim, procurou-se responder as seguintes perguntas de pesquisa.

[6] https://www.wikipedia.org/

1.2.1 Problema de Negócio

- **Problema**: Os usuários estão recebendo muitas recomendações óbvias e redundantes em relação ao que já foi visto anteriormente.
- **Pergunta**: Como melhorar a recomendação para aumentar a satisfação e utilidade do sistema para os usuários?

1.2.2 Problema Técnico

- **Problema**: Gerar listas ranqueadas de recomendações de acordo com critérios de relevância e diversidade.
- **Pergunta**: É possível melhorar a diversidade das recomendações sem que haja uma grande perda em sua acurácia?

1.3 Contribuições

Este trabalho avança o estado-da-arte em SR nos seguintes aspectos:

- Utilização do conjunto de dados semânticos da base de LOD DBpedia, com o objetivo de extrair e explorar os atributos dos itens e os seus relacionamentos;
- Formulação e desenvolvimento do algoritmo LOD-Diversification, que explora o conteúdo e a semântica dos dados de LOD para diversificar listas de recomendações;
- O algoritmo proposto consegue superar os algoritmos *baselines* encontrados no estado-da-arte de diversificação em SR, em diversas métricas de acurácia e diversidade encontradas na literatura.

1.4 Formalização do Trabalho

Esta seção descreve algumas formalizações utilizadas no decorrer deste trabalho.

$U = \{u_1, u_2, ..., u_m\}$ representa o conjunto de usuários do sistema em questão;

$I = \{i_1, i_2, ..., i_l\}$ representa o conjunto de itens utilizados pelo sistema de interesse;

O conjunto $C = \{c_1, c_2, ..., c_i\}$ representa o conjunto de atributos que descrevem os itens. O subconjunto $C_{rel} = \{c_{rel,1}, c_{rel,2}, ..., c_{rel,i}\}$ é formado por atributos que possuam uma relação específica rel como, por exemplo, para rel=gênero musical podemos ter os atributos rock e indie e para o relacionamento rel=país de origem os atributos França, Argentina.

$R_u = (i_1, i_2, .., i_n)$ representa representa a lista ordenada de itens gerada pelos algoritmos de recomendação base para um usuário u, onde n é o tamanho da lista, o escore associado de um item para um usuário pode ser representado pela função $rating(u, i)$

1.5 Diversidade em Sistemas de Recomendação

Como a área de SR tem em parte suas origens na área de RI, diversos conceitos e técnicas já consolidados nessa área do conhecimento foram adotados ou incorporados pelos SR [32]. Segundo Mouzhi et al. [17], o conceito de diversidade em SR pode ser dividido em diversidade inerente (*Inherent Diversity*) e diversidade percebida (*Perceived Diversity*). Diversidade Inerente considera a diversidade como uma visão mais objetiva e é geralmente mensurada através do cálculo da dissimilaridade entre os itens recomendados na lista de recomendações. O cálculo da Diversidade Inerente se refere, por sua vez, à lista de recomendações para um único usuário e é aplicado a um conjunto completo de todas as listas recomendadas para os usuários em todo o sistema de recomendação (*Aggregate Diversity*). A diversidade individual (*Individual Diversity*) considera o quanto os itens são diferentes em relação a uma lista de recomendações referente a um determinado usuário, enquanto que a diversidade agregada (*Aggregate Diversity*) considera a diversidade total de recomendações para todos os usuários. A alta diversidade individual de recomendações não implica necessariamente em elevada diversidade agregada. Por exemplo, o sistema de recomendação não personalizado *MostPopular*, que recomenda os itens mais populares aos usuários, pode recomendar itens que não são semelhantes entre si. Como resultado, a lista de recomendações para cada usuário é diversa (ou seja, de alta Diversidade Individual), porém o sistema tem uma baixa diversidade agregada. Assim, o conceito de Diversidade Inerente compreende trabalhos como

o *Intra-List Diversity* [42], *Expected List Diversity* [37] e *adapted MMR* [14] e [1].

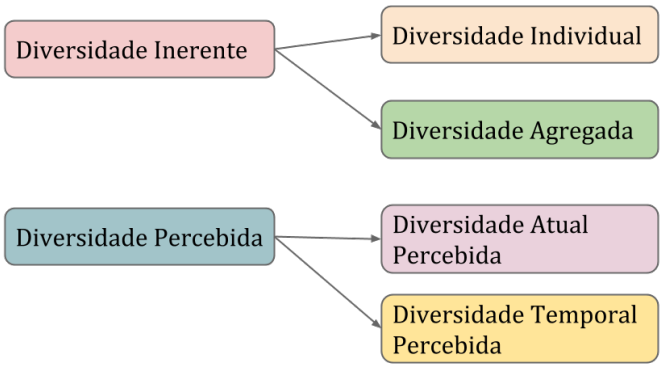

Figura 1.2: Diagrama entre os tipos de diversidade existentes.

O conceito de Diversidade Percebida entretanto, define a diversidade sob um ponto de vista mais subjetivo e pode ser mensurado somente a partir da avaliação explícita do usuário. Uma das maiores vantagens do uso da diversidade percebida é a captura real e direta da percepção do usuário, assim como a possibilidade de se avaliar a serendipidade[7] e o fator de diversidade/acurácia de forma mais realista. Lathia et al. [21] analisou o aumento da satisfação do usuário com o tempo em relação à técnica de *Diversidade Percebida* utilizada, denominada de Diversidade Temporal Percebida (*Temporal Perceived Diversity*). A diversidade temporal é a diversidade percebida pelo usuário ao longo de um período de tempo, já a Diversidade Atual Percebida (*Current Perceived Diversity*) significa a diversidade percebida por um usuário em um instante de tempo, a cada recomendação as listas são modificadas, não são estáticas.

A Tabela 1.1 demostra os trabalhos relacionados à diversidade dos SR e onde os seus trabalhos se encaixam nas divisões de diversidade citadas. Posteriormente, no Capítulo 2 2 e na Seção 5.1.1, entraremos em detalhes sobre os trabalhos mencionados que mais se adequam com a Diversidade Individual, que é o foco deste trabalho.

[7] A serendipidade diz respeito à novidade de recomendações e a forma como recomendações podem surpreender positivamente os usuários.

Tabela 1.1: Tabela Comparativa entre os trabalhos por tipos de diversidade.

	Diversidade Inerente		Diversidade Percebida	
	Individual	Agregada	Diversidate Atual Perc.	Diversidade Temporal Perc.
Carbonnel et al. 1998 [6]	X			
Ziegler et al. 2005 [42]	X			
Agrawal et al.2009 [2]	X			
Zou et al. 2010 [41]		X		
Lathia et al. 2010 [21]				X
Santos et al. 2011 [34]	X			
Castells et al. 2011 [8]	X			
Mouzhi et al. 2011 [17]			X	
Di Noia et al. 2014 [14]	X			

1.5.1 Métricas de Acurácia

nDCG é uma medida que mede a qualidade de ranking da lista de recomendações e calcula a utilidade (ganho) de um item com base no logaritmo do seu rank. Por ser calculada para cada usuário, a média leva em consideração que os usuários com mais itens no teste percorrem uma maior quantidade de eventos na lista ranqueada, diminuindo o peso do ranking com o acréscimo da quantidade de itens por usuário. A métrica nDCG é calculada dividindo-se o *Discounted Cumulative Gain* (DCG) pelo Ganho Acumulado Ideal (IDCG). Quanto maior o nDCG, melhor o ranking está classificado.

$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}} \tag{1.1}$$

A **F-measure** representa a média harmônica entre as métricas *Precision* e *Recall*:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{1.2}$$

1.5.2 Métricas de Diversidade

Nesta seção, apresentaremos as métricas de avaliação referentes à diversidade individual dos SR, além de métricas que são sensíveis à acurácia e ao ranking das listas de recomendação.

Intra-List Diversity - A Similaridade Intra-List (ILS) [42] é uma métrica de diversidade individual que calcula a similaridade entre os itens recomendados nas listas dos usuários de uma lista de recomendações para todos os usuários, com base em uma função de similaridade baseada em conteúdo (geralmente a similaridade do Cosseno ou Jaccard são aplicadas, retornando valores entre [-1,1]) e agrupa todos os seus valores em relação a um conjunto de tópicos específico. A *Intra-List Diversity* (ILD) normaliza o valor da dissimilaridade Intra-List. A métrica ILD representa bem o conceito de *Individual Diversity*, quanto maior o ILD maior será a diversificação da lista de recomendações. Entretanto a métrica ILD não é sensível a ordenação (ranking) dos itens na lista de recomendações em relação a lista original (na base de teste).

A fórmula do ILD é dada por:

$$ILD@k = \frac{1}{|U|} ILD_u@k \tag{1.3}$$

Para um usuário $u \in U$:

$$ILD_u@k(R) = \frac{1}{2} \sum_{i \epsilon R_u^k} \sum_{j \epsilon R_u^k} 1 - sim(i,j) \tag{1.4}$$

Onde:

Somatório par a par da similaridade entre todos os itens da lista de recomendação R.

A $sim(i,j)$ é o calculo da similaridade entre os itens i e j.

S-Recall - *Subtopic Recall* (S-Recall) [39] é uma métrica desenvolvida para sistemas de RI que se baseia na cobertura média de tópicos da lista de itens recomendados sobre o número total de tópicos existentes. Quanto maior o S-Recall maior a cobertura média de tópicos da lista de recomendações. Esta métrica retorna valores entre 0 e 1. O valor 1 ocorre quando, para todas as listas de recomendações para os usuários, todos os tópicos foram recomendados pelo menos uma vez. O S-Recall é uma métrica que considera um valor médio de cobertura e não é sensível ao ranking da mesma forma que a métrica ILD. Não é

o objetivo de um SR recomendar apenas a maior quantidade de atributos distintos possíveis, pois os usuários tendem a ter preferências por um subconjunto de atributos, uma técnica que prioriza o S-Recall recomendaria itens baseado na quantidade de atributos novos em relação aos itens já incluídos na lista de recomendações.

Formalmente, essa métrica é calculada como segue:

$$S - recall@k = \frac{|U_{i=1}^{K} topics(i)|}{|D|} \tag{1.5}$$

Sendo, $|D|$ o tamanho total do conjunto que contém todas os atributos existentes na base de dados.

A função $topics(i)$ retorna todos os atributos relacionados ao item i.

α-NDCG - É uma métrica de diversidade que utiliza o cálculo do nDCG [11] para dar peso ao ranking, o papel do parâmetro αno cálculo do score do vetor de ganho é responsável por recompensar geometricamente novos tópicos e de penalizar os tópicos redundantes, a partir de uma função de desconto log-harmônica de classificação. O parâmetro αcontrola a severidade da penalização. Quando α$= 0$ a métrica $\alpha{-}nDCG$ é equivalente à métrica $nDCG$.

$$\alpha - nDCG = \frac{\sum_{i=1}^{M} p_i S_i^{(\alpha-nDCG)}}{N^{(\alpha-nDCG)}} \tag{1.6}$$

EILD - A métrica *Expected Intra-List Diversity* (EILD) [37] generaliza a métrica ILD e introduz o fator de *rank-sensibility* e relevância.

$$EILD = \sum_{i_k, i_l, k \neq l} C_k disc(k) dist(l|k) p(rel|i_k, u) p(rel|i_l, u) dist(i_k, i_l) \tag{1.7}$$

- $disc(k)$ - a função que desconta do escore final a posição k do item avaliado da lista de recomendações. Pode ser implementada de diversas maneiras, como por exemplo, a função logarítmica $disc(k) = 1/log_2(k)$.

- $disc(l|k)$ - $disc(l|k) = disc(max(1, l - k))$, a função de desconto para a posição k, calcula a diferença da posição entre os itens k e l.

- $p(rel|i, u)$ - Representa a utilidade do item para o usuário, ou seja, quanto maior o escore que o item recebeu maior o peso dele para o cálculo: $p(rel|i, u) = \frac{2^{(max(0, r(u,i)))}}{2^{(max(R))}}$.

1.6 Topic Diversification

O algoritmo Topic Diversification [42] servirá de base para a explicação do algoritmo LOD-Diversification, apresentado na Seção 3. A ideia do *Topic Diversification* não considera somente a dissimilaridade dos atributos dos itens do usuário mas também considera a árvore taxonômica, a fim de expandir e generalizar o interesse do usuário em relação aos atributos dos itens recomendados. Iterativamente, depois de encontrar a dissimilaridade entre os itens e gerar uma lista de recomendações altamente diversificada o algoritmo possui um método para mesclar a lista original com a nova lista diversificada a partir do fator de diversificação λ. A seleção do item que será adicionado a nova lista de recomendações consiste na minimização da seguinte função objetivo:

$$f(i, u) = \lambda pos_R(i) + (1 - \lambda) pos_S(i) \tag{1.8}$$

Onde:

- O conjunto $S \subset R$ representa a nova lista de recomendações diversificada.
- $pos_R(i)$ - representa a posição do item i na lista de recomendações gerada pelo algoritmo base.
- $pos_S(i)$ - representa a posição do item i na lista de recomendações gerada pelo algoritmo base.

1.7 Ontologias e Web Semântica

Atualmente, a Web 2.0 se caracteriza por um imenso conjunto de documentos em linguagens de marcação como o HTML ou XML, largamente utilizadas no desenvolvimento de sites. Essas linguagens Web descrevem o conteúdo e a apresentação da informação através de páginas renderizadas pelos *browsers*. Porém, estes tipos de linguagem descreve somente a estrutura léxica e sintática do conteúdo do documento, deixando a estrutura semântica à parte dessa descrição. Somando-se à enorme quantidade de informações hoje disponíveis na Internet, recuperar alguma informação relevante tornou-se uma tarefa árdua.

Como resposta à demanda de aplicações capazes de processar as informações contidas na internet Tim Berner's Lee, em um artigo [4] define o que vem a ser a Web semântica, onde o conteúdo utilizado em páginas Web possui uma estrutura semântica. O objetivo dessa definição é permitir que as máquinas façam um melhor processamento dessas informações. Essa estrutura semântica poderia ser largamente utilizada em atividades como o comércio eletrônico e em sistemas de RI e SR.

Para a criação da Web semântica são necessárias estruturas de representação de conhecimento que se ajustam ao ambiente distribuído e descentralizado que é a internet. A constante evolução de sistemas de representação de conhecimento deu origem às ontologias. As ontologias são uma forma de representação de conhecimento que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre esses. Essa tecnologia adapta-se perfeitamente à necessidade de compartilhamento, reuso e manipulação do conhecimento dos sistemas inteligentes.

1.7.1 RDF

Resource Description Framework (RDF) é um framework e uma linguagem baseada em XML schema para descrever ontologias na Web semântica. O modelo de dados RDF é um grafo dirigido rotulado onde os vértices correspondem a entidades e as arestas são propriedades para conectá-los, ambos descritos e conectados através de Identificadores Uniformes de Recursos (URIs) únicos na Web. A semântica de tais propriedades é explicitamente modelada por meio de um esquema ontológico representado em RDF ou OWL. O RDF 1.7.1 ilustra um exemplo de uma ontologia em RDF com as definições do objeto que representa o artista *David Gilmour*[8] e suas relações de gêneros musicais, tópicos e cidade de origem.

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dbp="http://dbpedia.org/property/"...
  <rdf:Description rdf:about="http://dbpedia.org/resource/David_Gilmour">
    <foaf:name xml:lang="en">David Gilmour</foaf:name>
```

[8] http://dbpedia.org/page/David_Gilmour

```
        <dbo:birthYear rdf:datatype="http://www.w3.org/2001/XMLSchema#
            gYear">1946</dbo:birthYear>
        <dbp:placeOfBirth xml:lang="en">Cambridge, Cambridgeshire,
            England</dbp:placeOfBirth>
        <dbo:genre rdf:resource="http://dbpedia.org/resource/Art_rock" />
            ...
        <dct:subject rdf:resource="http://dbpedia.org/resource/
            Category:Pink_Floyd_members" />
        <dct:subject rdf:resource="http://dbpedia.org/resource/
            Category:Ivor_Novello_Award_winners" />
        <dct:subject rdf:resource="http://dbpedia.org/resource/
            Category:Pedal_steel_guitarists" />...
  </rdf:Description>
</rdf:RDF>
```

1.7.2 SPARQL

SPARQL (um acrônimo recursivo para *SPARQL Protocol and RDF Query Language*) é uma linguagem de consulta para RDF, ou seja, uma linguagem de consulta semântica para os banco de dados NoSQL[9] com o uso do "document-key-value", tal qual o sistema MongoDB[10]. A consulta pode ser distribuída para múltiplos *endpoints* de SPARQL (serviços Web que aceitam as consultas em SPARQL e retornam o resultado). O código fonte 1.1 em SPARQL abaixo ilustra uma consulta onde são solicitados todos os subgêneros de Rock na base da DBpedia.

```
SELECT * WHERE {
{ ?s <http://dbpedia.org/ontology/musicSubgenre>
     <http://dbpedia.org/resource/Rock_music>. FILTER isIRI(?s).
}
```

Código Fonte 1.1: Exemplo de consulta em SPARQL que retorna todos os subgeneros do gênero musical Rock

[9] *Not Only SQL* (NoSQL) é um termo genérico para uma classe definida de banco de dados não-relacionais que rompe uma longa história de banco de dados relacionais.

[10] https://www.mongodb.org/

1.7.3 Linked Open Data

O termo *Linked Data* se refere a um conjunto de boas práticas para publicar e conectar dados estruturados na Web Semântica. Todas essas bases de dados conectadas constituem a chamada *Web of Data* que, diferentemente da *Word Wide Web*, permite criar uma Web com dados estruturados, facilitando o desenvolvimento de softwares que utilizam esses dados. O *Linked Data* utiliza toda a tecnologia da Web Semântica como, por exemplo, o RDF, que provê um *framework* para a criação de base de dados em conjunto com as URIs utilizadas como identificadores únicos globais para os dados. A linguagem SPARQL, por sua vez, permite que consultas às bases RDF possam ser realizadas através de *endpoints* SPARQL, o HTTP é utilizado para o acesso e transporte dos dados na rede.

Atualmente, um grande esforço para conectar e publicar bases de dados abertos no formato de *Linked Data*, chamado *Linked Open Data* (LODs), está sendo empregado por diversas comunidades ligadas a Web Semântica. Um exemplo é o projeto LinkedBrainz[11], que disponibiliza e unifica a base de dados do *Music Brainz*[12] em uma ontologia RDF, conectada a LOD cloud. Serviços de redes sociais com música, como o Last.FM, utilizam os índices únicos do *Music Brainz* chamados de MBIds para referenciar unicamente uma música, artista ou álbum. A Figura 1.3 é uma representação de uma parte do projeto LOD, formando a chamada *Linked Open Data Cloud*.

1.8 Estrutura da Dissertação

A fim de melhor atender ao propósito desta dissertação e suas contribuições, optou-se por organizá-la e estruturá-la em sete capítulos principais, divididos em seções e subseções. O Capítulo 1 apresentou uma fundamentação teórica, onde foram definidos os conceitos de diversidade utilizados na pesquisa, métricas de avaliação dos SR existentes, assim como as tecnologias utilizadas no LOD e sua estrutura semântica. O Capítulo 2 expõe trabalhos sobre a diversidade em SR, tal como desenvolvimento de métricas e soluções para a diversidade de listas de recomendação e também trabalhos que abordam o *trade-off* entre acurácia e diversidade. O Capítulo 3 apresenta de maneira teórica e exemplificada o algoritmo proposto,

[11] http://linkedbrainz.org/

[12] http://musicbrainz.org/

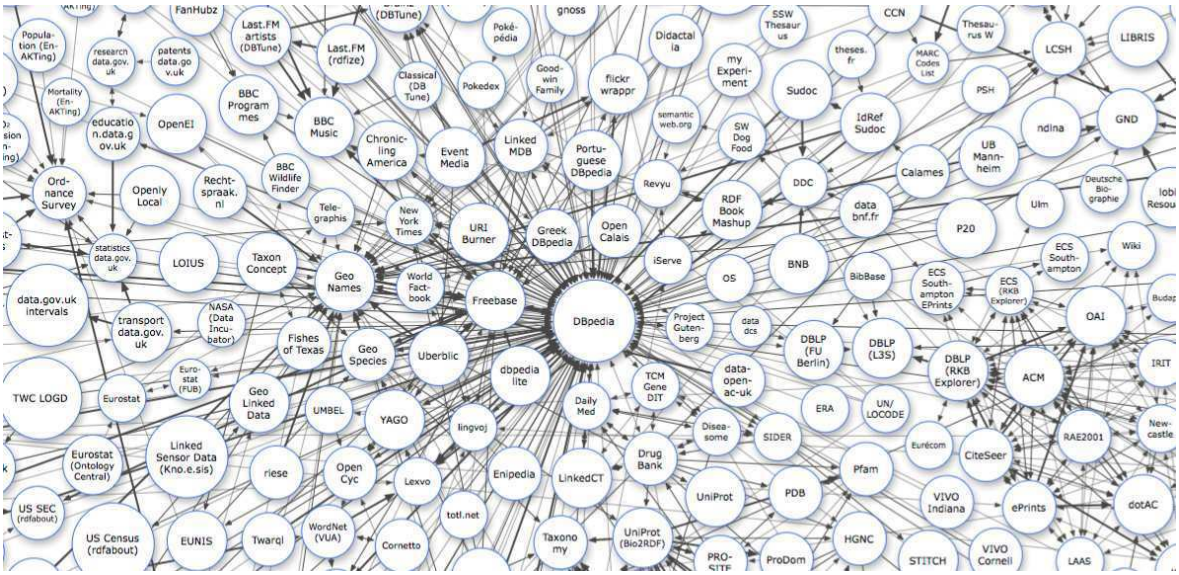

Figura 1.3: Diagrama do *Linked Open Data Cloud* (LOD Cloud), gerada em setembro de 2011

chamado de LOD-Diversification. O algoritmo aplica as relações semânticas da base de LOD no intuito de gerar listas de recomendação diversas e acuradas. O Capítulo 4 descreve as bases de dados utilizadas na pesquisa, apresenta também uma exploração e estatísticas acerca da coleta e tratamento dos dados. O Capítulo 5 expõe a metodologia seguida e os resultados do trabalho, em particular as análises comparativas entre os diversos algoritmos utilizados e seus resultados para as métricas de diversidade e acurácia. Finalmente, no Capítulo 6, são apresentadas as conclusões e considerações finais sobre este trabalho.

Capítulo 2

Trabalhos Relacionados

Foram identificados trabalhos que desenvolvem estratégias e métricas para avaliar a diversidade em SR, trabalhos que tratam o dilema entre a acurácia e a diversidade e por fim, artigos que fazem o uso de LOD para enriquecer a base de dados dos SR com conteúdo. Os trabalhos apresentados neste capítulo foram encontrados tanto por meio de sistemas de busca como o Google Acadêmico[1] e bibliotecas digitais da ACM[2], quanto percorrendo as referências dos principais trabalhos que eram encontrados e na análise de artigos produzidos nas principais conferências de SR e RI. Os trabalhos foram classificados em três grupos. O primeiro grupo trata do problema da diversidade e acurácia com o uso de LOD para SR, o segundo aborda o problema da diversidade em SR e RI e o terceiro aborda o dilema entre acurácia e diversidade em SR.

Neste capítulo, esses trabalhos são analisados com o objetivo de destacar as semelhanças e diferenças em relação ao trabalho apresentado nesta dissertação. Por fim, é apresentada uma tabela comparativa entre as estratégias de diversificação que mais se assemelham com o contexto de diversidade abordado neste trabalho. Destaca-se, também, de que forma as propostas e os resultados apresentados se comparam ao LOD-Diversification.

[1] https://scholar.google.com.br/

[2] http://www.acm.org/

2.1 Diversidade em SR e RI

O estudo da diversidade nos sistemas de RI está frequentemente relacionado a criação de abordagens para a redução da redundância e ambiguidade nos resultados das buscas. Assim como no estudo da diversidade nos SR, o problema da cobertura e novidade enfrentado pelos sistemas de RI [33; 11; 7] é classificado como um problema de complexidade NP-completo [26], onde temos dois objetivos conflitantes. Geralmente quando aumentamos a cobertura de um sistema de RI a busca apresentará muitos resultados ambíguos e redundantes, em contraposição, quando a novidade for alta a cobertura retornada pelo sistema de busca será baixa. Como resultado, para atingir um ranking diversificado ideal ambos os objetivos devem ser perseguidos.

Como o problema da diversificação nos sistemas de RI e SR é NP-Completo a estratégia é chegar em uma aproximação do resultado ótimo em um tempo polinomial. A abordagem de utilizar algoritmos gulosos para otimizar uma certa função objetivo pode ser utilizada, onde a função objetivo é utilizada controlar o trade-off entre as dimensões consideradas. Os trabalhos destacados abaixo utilizam diferentes tipos de funções objetivo e estratégias para diversificar as listas de recomendação e manter as listas acuradas e relevantes.

Em [40] Ziegler et al. propõem a métrica de diversidade denominada *Intra-List Diversity*, que oferece uma definição formal da diversidade em SR a partir da dissimilaridade apresentada entre os itens de uma lista de recomendação. O problema da diversidade em SR é formalizado como a otimização de duas funções objetivo, a similaridade da preferência do usuário e a diversidade entre itens recomendados.

Para minimizar a similaridade *intra-list*, os autores definem um algoritmo "guloso" de *re-ranqueamento* intitulado *Topic Diversification*, descrito na Seção 1.6, que diversifica a lista de recomendações *Top-N*, geradas por SR, pela seleção iterativa de itens que otimizam o *trade-off* entre o valor da recomendação original com a nova lista construída. O algoritmo explora a taxonomia dos atributos dos itens que serão recomendados para reordenar a lista. Para a avaliação, foi utilizada uma árvore taxonômica de livros recomendados pela *Amazon* no intuito de recomendar livros. Os experimentos foram realizados de maneira offline e online e demonstraram um melhor desempenho em relação a outros algoritmos estado-da-arte.

Vargas et al. [37] propõem um *framework* formal que unifica e generaliza várias métricas estado-da-arte e as aperfeiçoa com propriedades configuráveis, que não estão presentes nas avaliações passadas. Os autores identificam três conceitos principais para a novidade e diversidade de SR: escolha, descoberta e relevância, os quais formam a base para o desenvolvimento do *framework*. A avaliação foi realizada com as bases de dados do *MovieLens* e do Last.FM. As novas métricas conseguem capturar e avaliar a diversidade (EILD, descrita na Seção 1.5.2) e a novidade das listas de recomendação (EFD, EPD), assim como adicionam o valor da relevância a partir do ranking original da base de teste.

As métricas de ILD e EILD foram utilizadas para avaliar a diversidade e a acurácia das listas de recomendação geradas pelos algoritmos de diversificação, na seção de experimentação deste trabalho.

2.2 Dilema entre Diversidade e Acurácia

Ogawa et al. [25] desenvolveram, no contexto de recomendação de DVDs para comércio eletrônico, um algoritmo Top-N que diversifica os tópicos relacionados aos itens, visando afetar minimamente a acurácia do sistema. Para tal, o algoritmo cria uma matriz de similaridade *item x item* utilizando dados colaborativos dos *ratings* de usuários e, a partir do grafo criado, é aplicado um algoritmo de clusterização de Newman [18]. Cada um desses grupos de itens representa uma tendência de preferência, sendo selecionados os itens mais bem ranqueados de cada tópico.

Em [42], Vargas et al. identificam a diversidade do perfil do usuário através da extração de sub-perfis para um usuário, no intuito de refletir a natureza dos interesses dos usuários, gerando uma lista de recomendação para cada um dos sub-perfis. Em seguida, o algoritmo combina a lista gerada através de uma estratégia gulosa. A principal intuição por trás do trabalho é que alguns usuários podem preferir diversificação das recomendações, enquanto outros não. Além disso, os usuários podem estar inclinados à diversificação única com respeito a algumas dimensões de itens específicos (por exemplo, atributos item como diretor e ano no domínio filme) e não estar interessados em diversas sugestões relacionadas com os outros (como por exemplo, os gêneros de filmes como drama ou terror).

No intuito de diversificar listas de recomendações e tratar o *trade-off* entre diversidade e

acurácia, Noia et. al. [14] desenvolveram um modelo que captura as tendências dos usuários para recomendações diversas, em várias dimensões de atributos dos itens. Para tal, foram utilizados os valores de entropia e o tamanho do perfil do usuário para classificar os usuários em quatro quadrantes: a combinação entre a (alta/baixa) entropia de Shannon e o tamanho (grande/pequeno) do perfil baseado na mediana de todos os usuários. A decisão do uso da mediana força os usuários a se dividirem em dois grupos (metade dos usuários devem ter o tamanho do perfil menor do que a mediana). O fator adaptativo $\omega[0,1]$, presente na função de similaridade entre dois itens, determina a importância da diversidade para aquele dado atributo. Com essa mudança, os autores tentam predizer a importância daquela dimensão de atributos. A avaliação foi feita utilizando a base de dados do Movielens 1M, com um milhão de notas de filmes dadas pelos usuários, com as métricas Precisão, ILD e a sua média normalizada. Os resultados foram comparados com a lista de recomendações original, o algoritmo de re-ranqueamento MMR original e o MMR adaptativo, que é proposto no trabalho.

2.3 LOD em SR

Noia et al. [13] desenvolveram um SR baseado em conteúdo no contexto de recomendação de filmes que utiliza exclusivamente bases de dados de LOD, tais como DBpedia[3], *Freebase*[4] e *LinkedMDB*[5]. Para calcular a similaridade entre os objetos RDF, foi proposto um modelo clássico de espaço vetorial *(VSM)* [30], que é um modelo baseline para sistemas de RI. A similaridade entre dois objetos (filmes) para uma relacionamento específico (por exemplo, direção, gênero e atores) é calculada pela correlação entre os vetores de atributos que representam os itens e é quantificada pelo cosseno do ângulo entre eles. Os valores associados a cada elemento do vetor são obtidos pela fórmula *TF-IDF* (Term Frequency–Inverse Document Frequency) [36], bastante conhecida em sistemas de RI. Para descoberta de pesos para as propriedades, foram utilizadas duas abordagens. Na primeira foi desenvolvido um algoritmo genético que otimiza a busca pelos coeficientes dada uma função de *fitness*. A segunda abordagem foi baseada na análise estatística do SR colaborativo da Amazon. O treinamento do recomendador foi realizado a partir de dados extraídos da base de dados do

[3] `http://wiki.dbpedia.org/Datasets`

[4] `https://www.freebase.com/`

[5] `http://linkedmdb.org/`

sistema *Movielens*[6]. Sua avaliação foi realizada com as métricas Precisão e Recall. Os resultados mostraram que o uso de conteúdo de LOD melhorou a acurácia do recomendador em comparação com outros SR baseados em conteúdo, que utilizam palavras-chave extraídas de filmes e textos relacionados.

Como extensão do artigo anterior, Noia et al. [12] apresentam uma nova abordagem para o uso de LOD em SR utilizando um modelo de RI clássico, o *bag of words*. Cada item é representado por um vetor de pesos indicando o grau de associação entre o item e o recurso com respeito a uma propriedade. Os pesos são calculados separadamente para cada propriedade utilizando a fórmula *TF-IDF*. Os dados de treinamento foram novamente extraídos da base de dados do sistema *Movielens*. O algoritmo SVM foi utilizado como classificador. Na avaliação, comparou-se o impacto de diferentes tipos de propriedades do LOD, sendo os melhores resultados referentes às propriedades *subject*, *broader* e *wikilink* do DBpedia. Foram realizadas comparações entre diversos algoritmos de recomendação híbridos, baseados em conteúdo e de filtragem colaborativa. Os resultados mostram que o algoritmo proposto foi bem similar à filtragem colaborativa utilizando o *coeficiente de Pearson* como distância de similaridade.

Em uma publicação mais recente, Ostuni et al. [25] apresentam o *Semantic Path-based Ranking (SPRANK)*, um algoritmo de recomendação híbrido que computa as *Top-N* recomendações de feedback implícito. O *SPRANK* utiliza uma abordagem *path-based* que explora as conexões entre os itens por meio das ligações extraídas da base de dados do DBpedia. Todos os caminhos formados entre dois itens são classificados como *collaborative path*, *context path* ou *hybrid path*. Cada tipo de caminho tem o seu peso e quanto mais conexões esses itens tiverem mais relacionados os itens serão. Para o aprendizado do ranking *Top-N* foi adaptado um algoritmo de RI *BagBoo* [27], que utiliza um algoritmo que combina o *Random Forest* com o *Gradient Boosted Regression Tree (GBRT)* [15]. Para a avaliação, foi utilizada a base de dados do *MovieLens* e do Last.FM. Foram feitas algumas filtragens dos dados para que houvesse muitos casos de *coldstart* (início frio), onde o SR tem pouca ou nenhuma informação sobre o usuário ou item que deseja recomendar. O algoritmo *SPRANK* foi comparado com diversos outros algoritmos no contexto de *coldstart* e dados esparsos, tendo apresentado resultados que melhoraram os algoritmos estado-da-arte.

[6] `http://grouplens.org/datasets/movielens/`

2.3.1 Tabela comparativa entre Algoritmos de Diversificação

A Figura 2.1 apresenta uma tabela comparativa entre os algoritmos de diversificação de SR considerados estado-da-arte, incluindo os trabalhos de Ziegler et al. [42] e Vargas et al. [37] que serão utilizados neste trabalho na Seção 5, como baselines para a avaliação do algoritmo LOD-Diversification.

Tabela comparativa entre algoritmos de diversidade

Técnica	ano	autor	ac/div	método	Base de dados	limitações	métricas
Topic Diversification	2005	Ziegler et. al.	controla a ac/div, através da variável de controle θF [0-1] (quanto maior a variável θF maior a diversificação)	reranking; utiliza a taxonomia de features dos itens para calcular distribuir o score do vetor de features até a raiz da árvore, a partir de um fator de propagação k que reduz o score recebido a cada nível.	Book Crossing (offline, online)	Necessita de uma árvore taxonômica das features dos itens; Considera somente as features de nível menor na árvore (generaliza o perfil do usuário); As variáveis de controle fator de diversificação θF , assim como o fator de propagação k devem ser ajustados empiricamente; A diversificação não é personalizada para os diferentes usuários.	intra-list diversity; recall; precision;
IA-Select (adaptado por Vargas)	2009 2011	Agrawal et. al.	diversidade; função objetivo que maximiza a diversidade	reranking; algoritmo guloso que otimiza a função objetivo; calcula a distribuição da feature em relação ao perfil do usuário e do item; Features explicitas e fatores latentes da fatoração de matrizes são utilizadas para avaliação.	MovieLens 100k	Conta com ratings explicitos para o cálculo da função objetivo; Não se preocupa com a similaridade e nem existe um fator de diversificação que a controle.	nDCG-IA ERR-IA intra-list α-nDCG
MMR (adaptado por Vargas)	1998 2011	Carbonnel et. al.	controla a ac/div, através da variável de controle λ [0-1] (quanto menor o λ. maior será a diverisificação)	reranking; algoritmo guloso que maximiza a função objetivo; O componente de diversidade da função objetivo é determinada através da combinação linear entre o rating e a similaridade máxima da nova lista calculada pelo cosseno (não há similaridade negativa) entre as features compartilhadas dos itens.	MovieLens 100k	Conta com ratings explicitos para o cálculo da função objetivo; A variável de controle λ deve ser ajustado empiricamente; A diversificação não é personalizada para os diferentes usuários.	nDCG-IA ERR-IA intra-list α-nDCG

Figura 2.1: Tabela comparativa.

Capítulo 3

LOD Diversification

Uma abordagem de diversificação amplamente utilizada em SR é denominada *re-ranking diversification* [42; 40]. A abordagem apresenta um procedimento de rearranjo no ordenamento e na seleção de itens que pode ser aplicado a qualquer SR que gere uma lista de recomendações Top-N, com a finalidade de aumentar a sua utilidade para o usuário e evitar o *topic-overfitting*, onde o SR recomenda somente os tópicos conhecidos pelo usuário. Geralmente, para a realização dessa tarefa, são utilizados algoritmos gulosos que, além de diversificar a recomendação, são utilizados para regular o *trade-off* entre diversidade e acurácia, ou seja, capazes de aumentar a diversidade da recomendação sem uma grande perda na acurácia da recomendação em relação ao perfil do usuário.

O algoritmo LOD-Diversification, proposto neste capítulo, apresenta um novo método de re-ranqueamento, que explora as relações semânticas entre os itens e os seus atributos, disponíveis em bases de LOD públicas na Web. Tomando como base o algoritmo *Topic-Diversification*, descrito na Seção 3.3, o algoritmo desenvolvido consegue expandir a relação de distância entre os itens construindo um modelo baseado em Grafo a partir de um conjunto de atributos e seus relacionamentos, não se restringindo apenas às relações *"tipo_de"* contidas em uma estrutura taxonômica.

Neste capítulo, descreveremos os passos necessários para a execução do algoritmo de diversificação LOD-Diversification. Na seção 3.1, será descrito o procedimento de mapeamento e coleta das bases de dados de LOD, assim como a geração das listas de recomendação dadas como entrada para o algoritmo de re-ranqueamento. Em seguida, na seção 4.2, será discutido o modelo baseado em grafo utilizado no algoritmo proposto. Por fim, o algoritmo,

seu funcionamento e pseudo-código são apresentados com a exemplificação de cada passo realizado na seção 4.3.

3.1 Pré-Processamento

A etapa de pré-processamento do algoritmo LOD-Diversification consiste em: *i*) Mapeamento e conexão dos itens com a base de LOD, bem como a extração dos atributos dos itens e as relações entre estes atributos; *ii*) Seleção e extração do conjunto de atributos utilizados na diversificação e de suas relações semânticas; *iii*) Seleção do SR utilizado e a geração de uma lista de recomendações, fornecida como entrada para o algoritmo de re-ranqueamento.

3.1.1 Mapeamento e Extração da Informação da base de LOD

Para o enriquecimento da base de dados com o conteúdo disponível na base de LOD, foi necessário o mapeamento entre os itens a serem recomendados e o seu respectivo identificador (*URI*) na base de LOD escolhida.

Quando tal mapeamento não está disponível ou explícito na base de dados, é necessário aplicar uma técnica de *matching* entre os itens e os conceitos da ontologia, utilizando algum atributo compartilhado entre ambas as bases de dados como, por exemplo, o nome do artista ou da música com o atributo *label* de entidades do tipo *artist* contidos na base de LOD.

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?o, ?s
WHERE {
    ?o rdf:type dbpedia-owl:Artist.
        ?o <http://xmlns.com/foaf/0.1/name> ?n.
        ?o <http://dbpedia.org/ontology/birthYear> ?y.
        FILTER regex(str(?n), "David Gilmour")
        FILTER ?y = "1946-01-01"^^xsd:date
}
```

Código Fonte 3.1: Exemplo de mapeamento da URI para o artista David Gilmour

Para cada URI que representa um item no SR, é feita a extração dos atributos selecionados para diversificar a recomendação na base de LOD. Por exemplo, a seguinte consulta do

Código Fonte 3.2 em SPARQL na ontologia da DBpedia recupera todos os atributos de gêneros musicais relacionados ao artista *David Gilmour*:

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE {
    <http://dbpedia.org/resource/David_Gilmour>
    <http://dbpedia.org/ontology/genre> ?o.
}
```

Código Fonte 3.2: Exemplo de consulta em SPARQL

O resultado da consulta ilustrada em 3.2 são os atributos:

```
http://dbpedia.org/resource/Art_rock
http://dbpedia.org/resource/Progressive_rock
http://dbpedia.org/resource/Psychedelic_rock
```

Estes atributos são utilizadas para a diversificação dos itens contidos na lista de recomendação. Para tal, o algoritmo LOD-Diversification relaciona os atributos extraídos a partir das relações semânticas contidas na base LOD. No exemplo, temos um conjunto de gêneros musicais C={*Art rock*, *Progressive rock*, *Psychedelic rock*}, associados ao artista *David Gilmour* e pertencentes aos atributos *MusicGenre*. O gênero Art rock, por sua vez, possui algumas relações semânticas como mostra a Figura 3.1 que representa uma parte da rede semântica disponível na base de LOD da DBpedia. Os gêneros estão relacionados aos seus *subgêneros*, *gêneros derivados* e as suas *origens estilísticas*.

3.1.2 Geração da Lista de Recomendações

Nossa abordagem também segue o conceito de re-ranqueamento, ou seja, tomamos como entrada uma lista de recomendações gerada por um algoritmo base e reordenamos os itens da forma como será descrita no resto deste capítulo, bastando somente ao diversificador reordenar a lista original e retornar um subconjunto reordenado da primeira recomendação.

Para esta finalidade, são geradas recomendações ranqueadas top-N utilizando, por exemplo, algoritmos de SR que não utilizam os atributos dos itens para gerar recomendações. Dessa forma, é possível combinar métodos colaborativos com métodos baseados em conteúdo e assim conseguir um resultado mais ajustado para o usuário final.

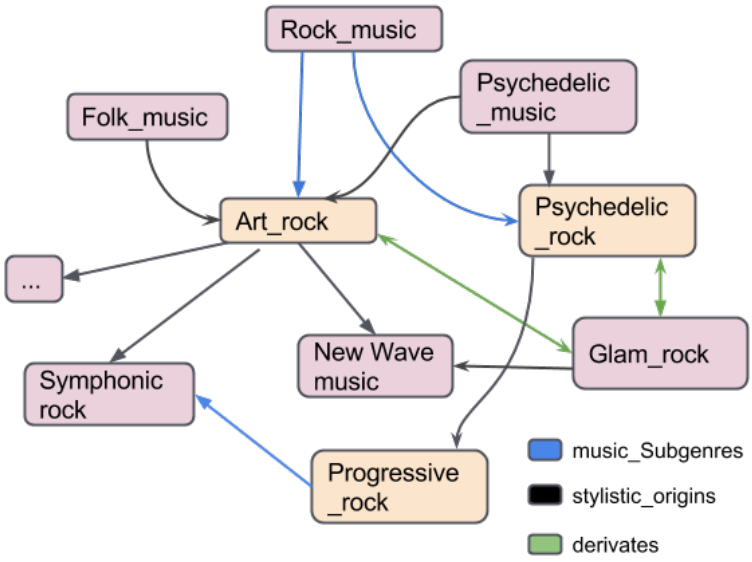

Figura 3.1: Parte de rede semântica descrevendo gêneros musicais.

A escolha do tamanho da lista de recomendações base pode influenciar diretamente o resultado do algoritmo. Ao incrementar N há uma maior probabilidade de aumentar a cobertura e a diversidade de atributos no conjunto de itens retornados. Ao aumentar o tamanho da lista para um valor próximo ou igual ao da lista de recomendações base, perde-se a capacidade de avaliar o modelo utilizando métricas que não são sensíveis ao ranqueamento como, por exemplo, as métricas ILS e S-Recall da nova lista diversificada será igual a da lista base original, limitando o algoritmo a realizar o reordenamento dos itens da lista original mantendo sempre os mesmos itens originais.

3.2 Modelo baseado em grafo

Como discutido no início deste capítulo, foi desenvolvido um modelo baseado em grafos para representar as relações semânticas entre os atributos dos itens recomendados. Esse modelo serve como guia para o algoritmo LOD-Diversification ao permitir uma melhor exploração dos itens, de maneira ajustada ao perfil do usuário, utilizando atributos extraídos da base de dados de LOD.

3.2.1 Construção do Grafo de Atributos

O conjunto $C_{i,rel} = \{c_1, c_2, ..., c_j\}$ é o subconjunto de todos os atributos de uma determinada relação rel, extraída de uma base de LOD, relacionados a um item $i \in I$. Cada atributo é utilizado para descrever o item em um determinado âmbito. No exemplo da Figura 3.1, pode-se dizer que *Progressive Rock* é um atributo do tipo gênero musical relacionado ao artista *David Gilmour*. Na tarefa de diversificação de atributos, a ideia principal é construir um grafo a partir desses atributos com base no conjunto $C_{i,rel}$ e, a partir de relações semânticas contidas na base de dados LOD, expandir esse grafo e distribuir níveis de relevância ou pesos entre seus vértices de acordo com a proximidade do vértice em relação à categoria inicial, como exemplificado na Figura 4.4 a seguir.

Dado um grafo $G = (V, R)$, onde V representa um conjunto de vértices e R um conjunto de relacionamentos entre os vértices, define-se inicialmente o conjunto de vértices como $V = C_{\text{rel}}$, onde C_{rel} é o conjunto de atributos da relação que desejamos aplicar o processo de diversificação e $R = C_{rel} \times C_{rel}$ o conjunto de relações entre os atributos extraídos a partir de conexões semânticas entre os itens da base de dados de LOD.

Como é possível observar no exemplo da Figura 4.4, considerando o contexto de recomendação de Artistas Musicais e a aplicação da diversificação ao conjunto de atributos de gênero $C_{genero} = \{artrock, progressiverock, psycodelicrock\}$, extraídos a partir do artista *David Gilmour*, tais gêneros são adicionados como novos vértices ao grafo criado. Em seguida, para cada gênero adicionado, seleciona-se novos gêneros que compartilham alguma relação semântica na base de LOD. Por exemplo, o gênero *Art Rock* possui uma relação de *stylistic origem* com os gêneros *Symphonic rock, New Wave music* e *Glam Rock*. Outros gêneros são adicionados de forma iterativa até que se atinja o limite de profundidade desejado, por fim um grafo dirigido conexo é retornado.

3.2.2 Distribuição de Pesos

Durante a etapa de criação do grafo de atributos, o algoritmo LOD-Diversification distribui pesos ou scores para cada vértice (atributo) da estrutura, começando inicialmente pelos atributos diretamente ligados ao item. Com a utilização do algoritmo *Breadth-first search (BFS)*, pode-se definir níveis de profundidade e diminuir o score à medida que o atributo selecio-

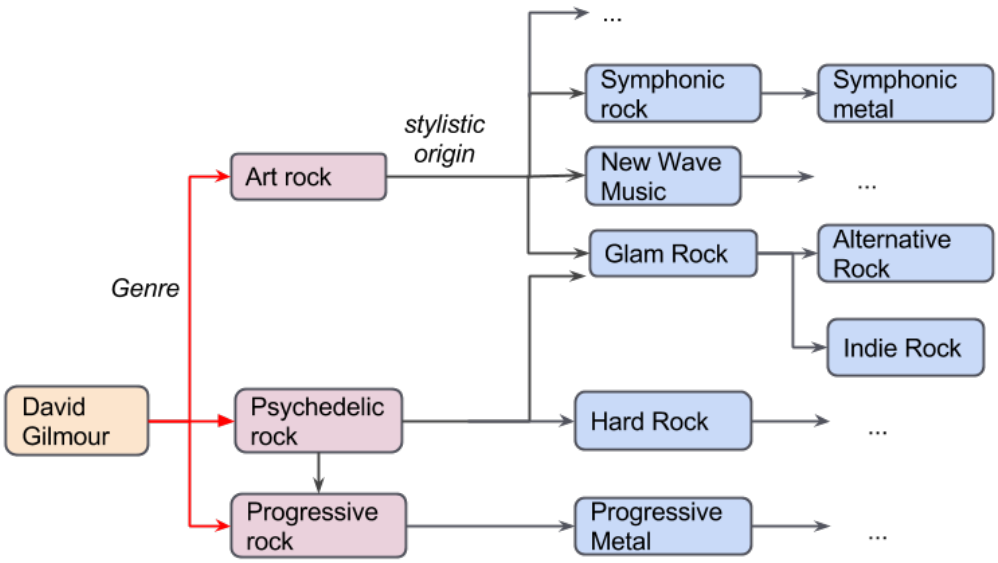

Figura 3.2: Construção de um grafo de atributos para o algoritmo LOD-Diversification.

nado se distancia do vértice inicial perdendo especificidade a cada novo nível ao percorrer o grafo de atributos. Por essa razão, o algoritmo escolhido para distribuir os scores no grafo de atributos se baseia no algoritmo de BFS, onde a cada nova iteração são selecionados novos atributos e para eles o nível no grafo é incrementado.

A fórmula do score de interesse (3.1) é calculada pelo score do vértice conector de menor nível (David Gilmour) e inversamente proporcional ao seu nível de profundidade, multiplicado pelo número de conexões ou vértices adjacentes. À medida que o algoritmo avança nos níveis do grafo, o valor do escore diminui proporcionalmente em relação ao fator de propagação k, como explicado na fórmula 3.1:

$$sco(c_i) = \frac{k * sco(parent(c_i))}{level(parent(c_i)) + |adj(parent(c_i)|} \tag{3.1}$$

Tal que a fórmula 3.1 possui os seguintes componentes:

- $sco(c_i)$ representa o escore atual da categoria c_i;
- $parent(c_i)$ representa o vértice c_j que participa da relação $(c_j, c_i) \in R$;
- k é o fator de propagação, permite um ajuste da dissipação do escore à medida que o algoritmo caminha em profundidade nos nós do grafo;

- A função $adj(c_i)$ retorna os vértices adjacentes ao atributo c_i;
- $level(c_i)$ representa o nível de profundidade que o vértice se encontra no algoritmo BFS.

Utilizando como exemplo o artista musical *David Gilmour*, que pertence aos gêneros {*Art Rock, Progressive Rock, Pychedelic Rock*}, a categoria *Art rock* terá, com $k = 1$, o $score = 1/(1+3) = 0.25$. A partir de seus gêneros derivados (relacionamento *derivate*), podem-se extrair os gêneros musicais {Glam Rock, New Wave Music, Gothic Rock, Post-Rock}, cada um com $score = 1 * 0.25/(2+4) = 0.041$ e $level = 2$. Dando continuidade ao algoritmo, a partir do gênero *Glam Rock* pode-se extrair os gêneros *Alternative Rock* e *Indie Rock*, com *level* = 3 e $score = 1 * 0.041/(3+2) = 0.0083$. De maneira similar, o algoritmo consegue descobrir novos atributos, atribuindo novos escores até criar o grafo de gêneros derivados do artista *David Gilmour*. Por fim, a normalização dos pesos (3.2) ocorre de acordo com um valor arbitrário *s*, que é dividido pelo número de itens na lista de recomendações e funciona como fator de normalização dos pesos dos atributos que descrevem os itens.

$$\begin{aligned} si &= s/|R|; \\ sco\_norm_i &= si/\sum_{i=1}^{|C|} sco(c_i); \\ sco(c_i) &= sco\_norm_i * sco(c_i) \end{aligned} \tag{3.2}$$

Ao dividir o $sco = 100$ para o item *David Gilmour*, tem-se $s = 300/3$. Assumindo que o somatório de pesos de cada vértice é igual a 2.23, pode-se atribuir à categoria *Art rock* o $score = 0.25 * (100/2.23)$, com valor igual a 0.11. A saída dessa etapa do algoritmo LOD-Diversification é um vetor de pesos representando o item i, do tipo $\vec{v_j}. = (v_{i_1}, v_{i_2}, ..., v_{i_{|C_{i,relation}|}})$, onde $v_{i,k}$ representa o score para a categoria $c_k \in C_{i,relation}$.

3.2.3 Similaridade baseada em Grafo

O algoritmo LOD-Diversification utiliza estratégias baseadas no cálculo da similaridade dos atributos dos items. Para tal, um tópico fundamental na tarefa de re-ranqueamento é a representação dos itens recomendados ao usuário em relação aos seus atributos e o método de similaridade utilizado. Como mencionado nas Seções *4.2.1* e *4.2.2*, os passos para criação

e distribuição de pesos no grafo de atributos servem para montar o vetor de atributos representando cada item. Para calcular a similaridade entre itens, foi escolhida a correlação de *Pearson*. Assim, para o vetor de atributos v_i e v_j, o cálculo de similaridade é dado por:

$$PearsonSim(v_i, v_j) = \frac{\sum_{k=0}^{|C|} (v_{i_k} - v_i) \times (v_{j_k} - v_j)}{\sqrt{\sum_{k=0}^{|C|} (v_{i_k} - v_i)^2 \times \sum_{k=0}^{|C|} (v_{i_k} - v_i)^2 \times \sum_{k=0}^{|C|} (v_{j_k} - v_j)^2}} \tag{3.3}$$

3.2.4 Mensurando a Propensão da Diversidade para o Usuário

Recentemente, alguns trabalhos [14; 38] vem abordando o uso do perfil do usuário para medir a propensão do mesmo em relação à diversidade, ou seja, a inclinação do usuário em relação a itens diversos baseada em seu comportamento, a partir da diversidade apresentada no histórico do usuário poderemos inferir a propensão do usuário à diversidade.

Tomando por base os trabalhos anteriores e a análise da base de dados utilizada na validação e teste do trabalho aqui exposto, verificou-se que os SR estado-da-arte não geram listas de recomendações com a diversidade próxima a ideal observada através do perfil do usuário. Como é possível perceber na Figura 3.3, para a avaliação do ILD dos gêneros musicais contidos em cada uma das listas, os SR base (representados pelos algoritmos BPRMF e KNN) recomendam listas de itens com uma diversidade menor do que a percebida tanto no treino quanto no teste (Train e Test).

Para contornar essa questão, a ideia é utilizar o perfil do usuário para inferir a propensão do usuário a recomendações diversas a partir da diferença entre o nível de diversidade (ILD) do usuário na lista base de recomendação com a diversidade apresentada em seu histórico. Para um dado conjunto de atributos C, o valor de $Prop_{u,C}$ é dada por:

$$Prop_{u,C} = ILD_{u_i,C}(T) - ILD_{u_i,C}(R) \tag{3.4}$$

- T o conjunto de itens que o usuário consumiu, representando o seu perfil;
- R a lista de itens recomendada para o usuário u_i por um SR.

Com o intuito de normalizar o valor de $Prop_{u,a}$, a normalização por desvio padrão (sd) é utilizada, de acordo com a fórmula a seguir.

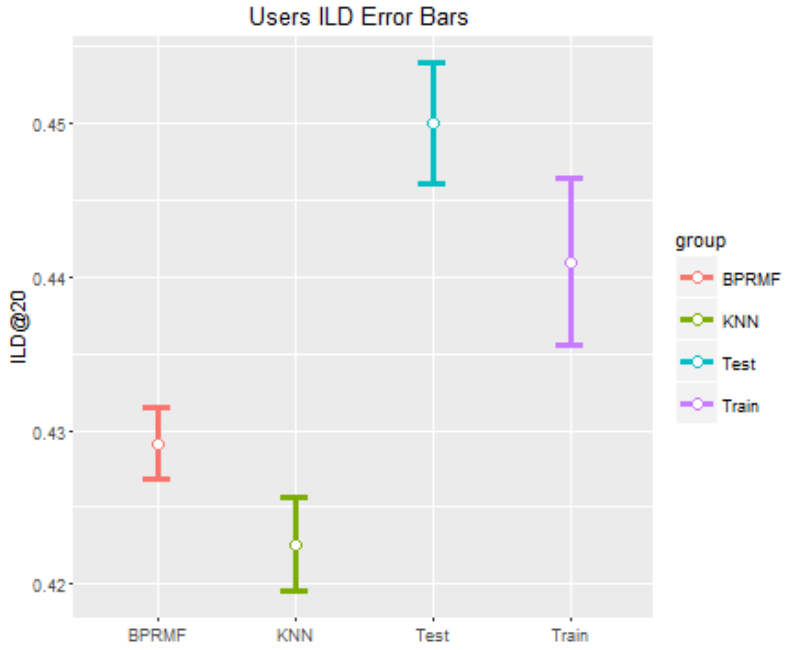

Figura 3.3: Desempenho ILD para gêneros musicais por cada lista de itens.

$$Prop\_norm_{u,a} = Prop_{u,a} - \frac{\text{média}(Prop_a)}{sd(Prop_a)} \tag{3.5}$$

A fórmula da propensão do usuário será utilizada para determinar o fator de diversificação utilizado no algoritmo LOD-Diversification, que recebe valores entre $[0, 1]$. Recomendações com o nível de diversidade próximos ao perfil do usuário irão receber um fator de diversificação baixo e consequentemente nova lista de recomendações será mais parecida com a lista base. Por outro lado, quando a diferença entre os níveis de diversidade for positivamente alta, o algoritmo LOD-Diversification retornará uma nova lista de recomendações proporcionalmente mais diversa do que a original.

3.3 Algoritmo LOD-Diversification

O algoritmo LOD-Diversification baseia-se na ideia proposta por Ziegler et al. [42], de explorar as relações existentes entre os atributos, com o intuito de diversificar a lista de recomendações. Além de explorar o grafo de atributos, o LOD-Diversification tenta inferir a propensão do usuário em relação ao nível de diversidade, utilizado pelo algoritmo para um

melhor ajuste da diversificação e explora as relações semânticas contidas em bases de dados de LOD. O algoritmo utiliza ainda estratégias gulosas que otimizam uma função objetivo descrita na Equação 3.6. Essa função objetivo tem a função de controlar o *trade-off* entre a lista diversidade (lista diversificada S) e acurácia (lista base R de recomendação).

$$f(u,i) = \lambda rank_R(i) + (1-\lambda) rank_S(i) \tag{3.6}$$

Onde:

- O conjunto $S \subset R$ representa a nova lista de recomendações diversificada.
- $rank_R(i)$ - representa um valor normalizado entre [0,1] que representa a posição do item i na lista de recomendações gerada pelo algoritmo base.
- $rank_S(i)$ - representa o ranking normalizado entre [0,1] que representa a posição do item i na lista.

A tupla $R_u = (i_1, i_2, ..., i_n)$ corresponde à entrada para o algoritmo LOD-Diversification em conjunto com o valor de $depth$ e o tamanho da lista original, representada pela variável m. O pseudo código 1 ilustra os passos do algoritmo proposto. O primeiro passo, correspondente a *linha 2* do algoritmo, é calcular o valor da propensão normalizada do usuário $u \in U$ à diversidade $Prop_{u,norm} \in [0,1]$, veja a Equação 3.4. Esse valor é necessário para o cálculo da função objetivo para cada item candidato na nova lista de recomendações. Para cada item $i \in R_u$, é feita uma chamada a função lod_graph, na linha 5, que percorre o grafo semântico de cada uma de seus atributos até a profundidade representada pelo valor $depth$, que é passado como parâmetro, chamamos de função *lod_graph* as etapas correspondentes as Seções 3.2.1 e 3.2.2, responsáveis pela construção e distribuição de pesos para grafo de atributos. Cada item i passa a ter um novo vetor de atributos $c \in C$ que considera novos atributos extraídos do grafo extraído da base de LOD. Na linha 7 definimos que $S_u(1) = R_u(1)$, ou seja, o primeiro item da lista de recomendações base é sempre adicionado no começo da nova lista de recomendações $S_u = [i_1, i_2, ..i_m]$, com o objetivo de manter uma alta acurácia para a nova lista e de obter a base para o cálculo da diversidade entre o o primeiro item da lista com os itens candidatos da lista de recomendações base.

Para cada entrada da lista de itens candidatos, é calculada a dissimilaridade entre o item candidato e a nova lista de recomendações. Com base nesse cálculo, a lista é ordenada e os itens com maior diversidade em relação a nova lista de recomendações ficam nas primeiras posições da lista de itens candidatos. Por fim, é realizado um *merge* entre a lista B_u e a lista original R_u de acordo com o fator de diversificação extraído do perfil do usuário mencionado anteriormente e o *rating* normalizado, provido pelo SR selecionando iterativamente o item da lista B_u que minimiza a função objetivo até que a lista diversificada S_u esteja completa, ou seja, $|S_u| = m$. Por esse motivo, o tamanho da lista de entrada R_u deve ser consideravelmente maior do que o tamanho m da nova lista S_u. Nos experimentos, considerou-se gerar listas de recomendações *Top@20* e *Top@30* a partir de listas de recomendação base *Top@50* geradas por SR estado-da-arte e *baselines*. Evitamos a avaliação da diversidade para listas de recomendações menores, como *Top@5* ou *Top@10*, com o objetivo de obter uma melhor avaliação em métricas como S-Recall e ILD.

Algorithm 1 Pseudocódigo do algoritmo LOD-Diversification

procedure LOD-DIVERSIFICATION($R_u, depth, m$)
    $B_u \leftarrow R_u$
    for each item i in R_u **do**
        $B_{u,i} \leftarrow lod\_graph(i, depth)$
    end for
    $S_u(1) \leftarrow B_u(1)$
    for $z = 2$ to m **do**
        $B_u \leftarrow not\_intersectS_u$
        for each item i in B_u **do**
            $v_{i,dist} \leftarrow dist(i, S_u)$
        end for
        $B_u \leftarrow order_inv(B_u, v_{dist})$
        for each item b in B_u **do**
            $scores(b) \leftarrow (1 - weight(R_{u,b})) \times (1 - U_prop_{norm}) + (1 - weight(B_{u,b})) \times (U_prop_{norm})$
        end for
        $S_u(z) \leftarrow B_u(min(scores))$
    end for
    Return S_u
end procedure

3.4 Considerações Finais

O algoritmo LOD-Diversification utiliza técnicas de re-ranqueamento e explora os grafos dos atributos para melhor ajustar a diversificação das recomendações. O valor de $depth$ no algoritmo LOD-Diversification define a profundidade máxima que o algoritmo percorre o grafo. O fator de propensão pode, por sua vez, ser utilizado para aumentar ou diminuir a diversidade da lista e determinar o quão distante da lista original o usuário está propenso a aceitar e consumir as recomendações.

Capítulo 4

Bases de Dados

Além da comodidade, facilidade e portabilidade de consumir uma mídia digital, o grande acervo e a disponibilidade a qualquer hora do dia através da Internet permite aos consumidores explorar cada vez mais novas possibilidades, na descoberta de novos artistas, filmes ou gêneros musicais. Um diferencial da recomendação musical se trata da sua grande oferta de estilos musicais, com culturas, épocas e línguas diferentes, além de uma variedade de gostos e tipos de ouvintes. Por essa razão, no intuito de ajudar os consumidores de músicas digitais a explorar os diferentes tipos de categorias musicais existentes de uma forma personalizada e de melhor avaliar o algoritmo desenvolvido, já que contém uma base ampla de usuários e bases de LOD disponíveis, escolhemos aplicar o conceito de diversificação utilizando o algoritmo LOD-Diversification na recomendação de artistas musicais.

4.1 Last.FM

O *Last.FM* [1] é uma rede social musical que tem como principal característica o *scrobbling*, que funciona como um serviço que permite registrar o histórico de músicas escutadas pelos usuários em um sistema de *streaming* ou *player* de música, como o *Spotify*[2] ou *Media Player*. Além disso, o sistema fornece outros recursos como: serviço de rádio online, recomendador de artistas, criação de *playlists*, estatísticas básicas acerca do perfil de uso do usuário, informações detalhadas sobre os artistas, seus eventos futuros e notícias.

[1] http://www.lastfm.com

[2] https://www.spotify.com/us/

Com o objetivo de prover dados públicos, suscitar e facilitar estudos na área musical, o Last.FM provê uma API RESTful[3] que possibilita o acesso aos dados presentes em sua base de dados, através de uma interface *Web Service*. Com a API, é possível coletar informações acerca de usuários (incluindo o histórico musical), músicas, álbuns, artistas, entre outras.

4.1.1 Coleta de Dados

Nos experimentos realizados neste trabalho, os usuários e os seus históricos de *scrobbling* de artistas do Last.FM foram os sujeitos da pequisa. A base de dados foi coletada utilizando a API pública do Last.FM e contém o histórico dos artistas mais escutados por usuário e seus respectivos *playcounts*, ou seja, a lista ordenada de todos os artistas escutados e número de vezes em que o usuário escutou aquele artista. Coletaram-se todos os artistas escutados pelos usuários desde o seu cadastro no Last.FM até maio de 2015, junto com o conjunto total de *playcounts*. O método da API utilizado foi o *getTopArtists*, que possibilita coletar a partir do identificador do usuário do Last.FM uma lista com os artistas mais escutados pelo usuário, onde o tamanho da lista pode ser o número total de artistas escutados por ele. A amostragem que constituiu esse conjunto de usuários do Last.FM foi coletada a partir do trabalho acerca de novidade em SR de Andryw et al. [23]. O processo de coleta e filtragem resultou em uma amostra de 8.514 usuários coletados, com um total de 34.852 artistas musicais diferentes e 1.380.000 pares ordenados entre usuários e artistas.

4.1.2 Análise dos Dados

Como mencionado em estudos anteriores [9][10], a análise sobre o consumo de músicas de artistas pelos usuários tem em geral uma distribuição assimétrica, ou seja, o chamado efeito da "cauda longa". Poucos artistas têm muitos *playcounts* e muitos artistas têm poucas *playcounts*. Como é possível observar na Figura 4.1, a distribuição acumulada de *playcounts* para os artistas, na base de dados coletada, é totalmente assimétrica. Os top-50 artistas mais escutados por exemplo, agrupam 15,58% de todos os *playcounts* do conjunto de dados e o top-500 51%. Os usuários possuem uma esparsidade de 0,9953 na base coletada, calculada

[3]Application Programming Interface - conjunto de rotinas fornecidas por um software para que aplicativos acessem suas funcionalidades

a partir da porcentagem de itens que foram escutados dividido pelo número de itens total.

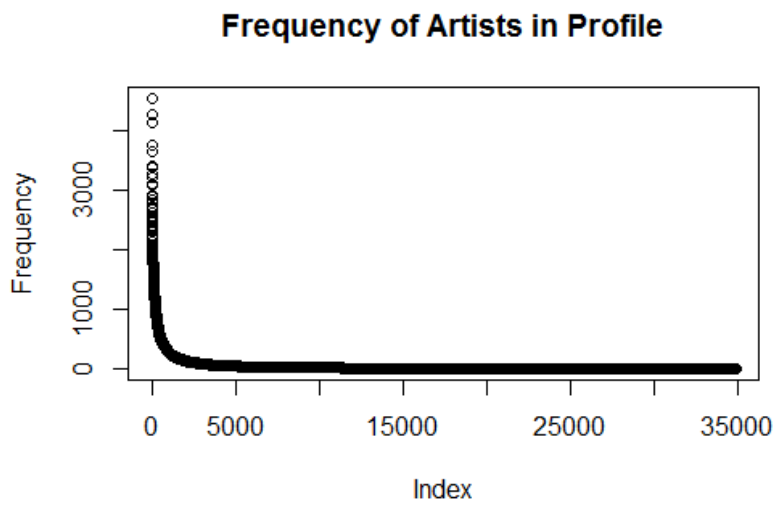

Figura 4.1: Distribuição da quantidade de artistas no perfil dos usuários.

Tabela 4.1: Artistas musicais mais populares na base de dados

Artista	Soma de todos os playcounts por Artista
The Beatles	3,496,386
Radiohead	3,431,030
Arctic Monkeys	2,502,529
Muse	2,275,396
Coldplay	2,205,384

A Figura 4.2 apresenta a distribuição do tamanho do perfil do usuário na base de dados coletada.

O número de *playcounts* foi normalizado para uma escala de [0,1], obtendo um escore proporcional de artistas por usuário e dividindo o número de reproduções pela contagem total do perfil do usuário. Para evitar a inconsistência nos dados de usuários com playcounts atípicos ou outliers, foram aplicados os seguintes filtros:

- Tamanho do Perfil: restringiu-se o número de artistas para o perfil do usuário para conter no mínimo 50 artistas, a fim de evitar problemas de avaliação e situações de *coldstart*;

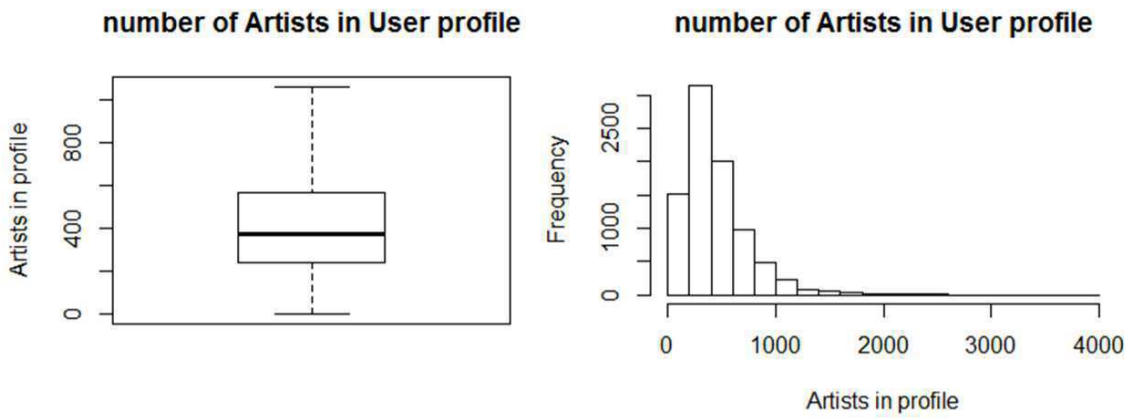

Figura 4.2: Distribuição do número de playcounts de artistas que foram escutados por usuários na base do Last.FM.

- Playcount: como mencionado anteriormente, o banco de dados é composto por feedback implícito, ou seja, considerou-se como feedback positivo os artistas que foram ouvidos pelo usuário. Para melhorar a precisão desta inferência, foram removidos os artistas ouvidos menos que 10 vezes por um usuário e, para remover possíveis *outliers*, foram retirados os artistas com *playcounts* maiores que 40.000.

4.1.3 Agrupamento de Perfis dos Usuários

Com o objetivo de identificar e analisar os diversos perfis dos usuários, agrupá-los para melhor entender o seu comportamento e descobrir se realmente há diferenças em relação à diversidade dos usuários, foram utilizadas métricas de clusterização para entender se há de fato um padrão de consumo musical, quais são eles e como podemos dividi-los. Para identificar estes grupos, foi aplicado um algoritmo de agrupamento baseado em densidade (DBscan) nos dados que caracterizam os usuários que são o foco deste trabalho, os atributos dos artistas escutados. Para representar o gosto musical de um usuário, foi utilizada a equação 4.1 que gera um vetor de pesos para cada atributo dos itens que o usuário consumiu, no intuito de criar uma matriz de dissimilaridade fornecida como entrada para o algoritmo DBScan [31].

$$u_{i,cat} = \sum_{i \in I} \sum_{c \in C} playcount_{norm}(i) * score(c) \tag{4.1}$$

Onde:

O score do atributo c é a importância do atributo para o item, também normalizado para um valor entre 0 e 1.

O cálculo da métrica de similaridade par a par entre todos os usuários da base de dados foi realizado entre os vetores de atributos, extraídos da base da DBpedia. Assim, a dissimilaridade entre dois usuários foi calculada a partir do complemento da distância do cosseno ($1 - cosseno$), dado pela Equação4.2:

$$Cos_{rel}(u_i, u_j) = \frac{\sum_{k=1}^{n} v_{i,k} * v_{j,k}}{\sqrt{\sum_{i=1}^{n} v_{i,k}^2} * \sqrt{\sum_{j=1}^{n} v_{j,k}^2}} \tag{4.2}$$

Onde o vetor v_i representa o vetor de pesos para os atributos relacionados ao usuário i.

O algoritmo de agrupamento DBScan não necessita explicitamente do número de grupos resultantes, como é o caso do KMeans. Entretanto, é necessária a declaração do número mínimo de pontos (*minpts*) dentro de um raio (*Eps*). Para determinar o valor de *Eps*, foi utilizado o "método de joelho"que determina uma configuração de grupos que não adicionem muita heterogeneidade, evidenciado a partir da curvatura máxima do gráfico (joelho) gerado pela distância dos k vizinhos mais próximos. Aplicando esse método, o valor definido para Eps foi 0.135. O valor de *minPoints*, por outro lado, foi determinado de forma empírica, após diversas execuções do algoritmo. A escolha do melhor resultado da clusterização se deu a partir da maior média de silhueta encontrada, com *minPoints* = 100. O resultado do agrupamento utilizando a categoria de gênero musical é apresentado na Figura 4.3, na qual é possível distinguir alguns grupos de usuários distintos como, por exemplo, o grupo de Metal com o de Indie Rock e alguns com uma certa proximidade como, por exemplo, os usuários que escutam Indie Rock com o rock Alternativo. No gráfico, pode-se perceber que os usuários mantém um certo padrão de consumo de músicas de certos gêneros musicais. Além disso, esses gêneros possuem um certo tipo de relação de proximidade, como vemos mais similaridade entre os ouvintes de Heavy Metal com os ouvintes de Hard Rock do que com os de Hip Hop, por exemplo.

O agrupamento mostrou-se uma maneira eficaz de observar o comportamento dos usuá-

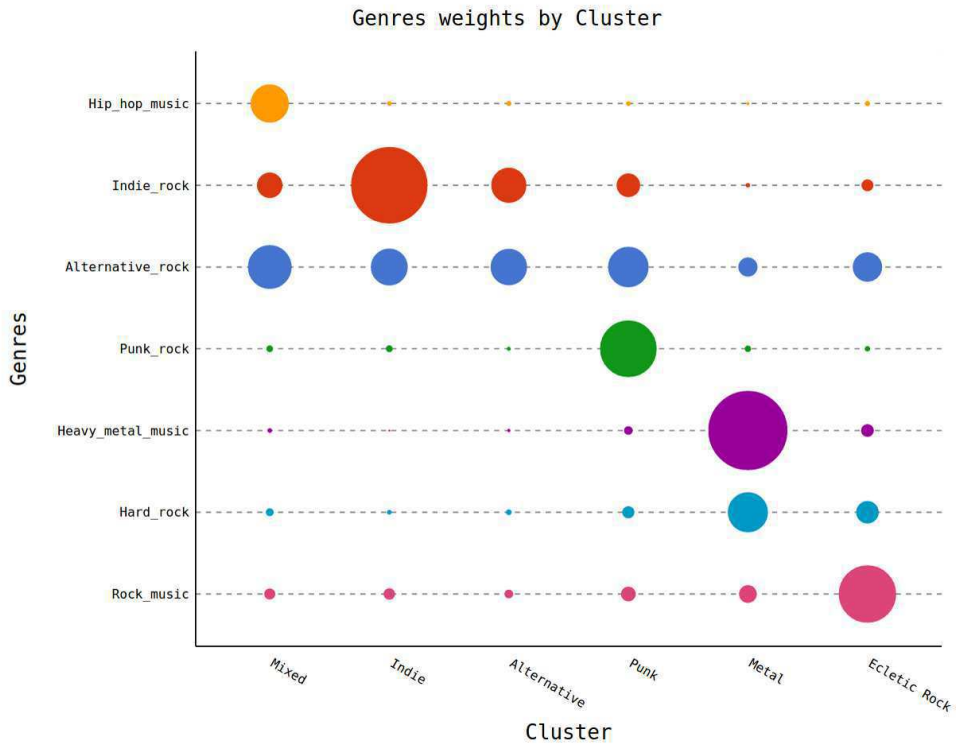

Figura 4.3: Agrupamento de usuários por gênero musical.

rios como, por exemplo, o de manter uma certa diversidade de gêneros musicais afins. O algoritmo proposto, a LOD-Diversification, pretende explorar esses relacionamentos implícitos entre os atributos dos itens para melhor diversificar a lista de recomendações. Entretanto, não se pretende recomendar itens de atributos totalmente contraditórias às contidas no histórico (já conhecido) do usuário. Em outras palavras, a ideia é diversificar a recomendação sem entretanto denegrir de forma negativa a acurácia do SR.

4.1.4 Particionamento dos Dados

Os dados coletados do Last.FM foram particionados aleatoriamente entre treino e teste utilizando métodos de validação cruzada 5-Kfold, divididos em 80% para treino e 20% para teste. Na validação cruzada Kfold, o conjunto de dados é dividido em k subconjuntos. A partir desses subconjuntos, um é retido e usado como conjunto de teste. Os demais subconjuntos são usados como conjunto de treino. Esse processo é repetido k vezes, cada vez com um subconjunto diferente de teste.

Avaliações *off-line* são formas rápidas, econômicas e fáceis de serem realizadas em uma grande quantidade de dados, com diversos conjuntos de dados e diferentes algoritmos. No entanto, esse tipo de avaliação só permite o cálculo de previsões para itens que foram, na verdade, avaliados pelo usuário, de modo que a esparsidade limita o conjunto de itens que podem ser avaliados. Apesar disso, métricas *off-line* não podem medir satisfação verdadeira do usuário.

4.2 Mapeamento para a base de LOD DBpedia

Dado que a abordagem proposta é baseada em bases de dados LOD, é necessário mapear os artistas do Last.FM para objetos na base de LOD. Neste trabalho, a base de dados LOD escolhida é a DBpedia[4]. A DBpedia é um dos principais projetos na *Linked Open Data Cloud* e é considerada a base com mais conexões entre todas as bases de LOD existentes. Seu crescimento se deu com a extração de informações estruturadas da Wikipedia[5]. Atualmente, conta com mais de 3.77 milhões de objetos. Todas essas informações são descritas no formato RDF e disponibilizadas na Web para consultas em SPARQL através dos seus *endpoints*.

Cada artista do Last.FM no conjunto de dados (34.852 artistas) foi mapeado para o seu respectivo objeto *dbpedia-owl:musicalArtist*, *dbpedia-owl:Band* ou *dbpedia-owl:Person*.

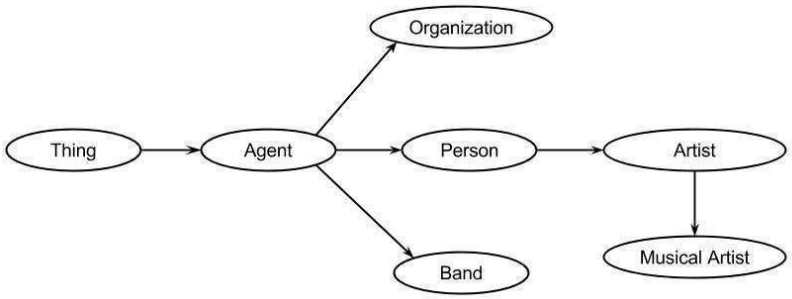

Figura 4.4: Sub-árvore da base de LOD DBpedia, cada vértice representa um objeto e cada aresta uma relação taxonômica.

Para cada artista ou banda musical, extraíram-se vários atributos com o objetivo de utili-

[4]dbpedia.org

[5]Wikipédia é considerada a maior enciclopédia virtual da atualidade. Ela conta com um sistema colaborativo em que autores escrevem e corrigem os artigos criados.

zar os atributos dos itens como base para a diversificação das listas de recomendações usando o LOD-Diversification. A partir da lista de todas as relações existentes para artistas musicais na base do DBpedia, determinaram-se as relações *Genre*, *Subjects* e *Origin* como as mais relevantes e importantes para o usuário final.

Gêneros musicais (Genre)

Representado pela relação *dbpedia-owl:genre*, gênero é a categorização do estilo musical em tipos que podem ser distinguidos de outros tipos de músicas. Pode ser aplicado a um artista musical, uma banda, um álbum ou uma música. Na base de dados, foram coletados os gêneros de 28.360 Artistas (81.37% de todos os artistas da base de dados) na DBpedia. No total, foram retornados 2.909 diferentes gêneros musicais na base de dados. Tem-se em média dois gêneros musicais relacionados a cada artista e no máximo 28 gêneros descrevem um único artista musical na base de dados. A Tabela 4.2 reflete os gêneros musicais mais populares da base de dados.

Tabela 4.2: Gêneros musicais mais populares na base de dados

Gênero Musical	Soma dos scores por gênero
Pop_music	3048.81
Rock_music	2353.87
Alternative_rock	2008.27
Hip_hop_music	1851.55
Jazz	1685.30

A categoria de gênero musical por sua vez possui um conjunto de relacionamentos a serem explorados pelo algoritmo LOD-Diversification: *Subgenre*, *FusionGenre*, *Stylistic_Origin* e *Derivate*. Alguns detalhes e análises realizadas a partir dos grafos extraídos na DBpedia se encontram no Apêndice C desde documento.

- *Sub Gênero*: a relação dbo:musicSubgenre representa um tipo de relação de especialização ou subdivisão de gêneros musicais. Esse tipo de relação pode ser representado por uma árvore taxonômica. A árvore gerada tem 481 diferentes vértices (gêneros) e 509 arestas. A Figura 5.5 ilustra um sub-grafo com os subgêneros do estilo Rock;

- *Fusão de Gênero*: a relação dbp:fusiongenres representa a fusão entre dois ou mais

gêneros, ou seja, gêneros musicais que combinam diferentes elementos formando um novo gênero musical. Por exemplo, o gênero *Eletrohouse* tem características da música Eletrônica, Pop, Synthpop, Eletroclash e Tech house;

- *Origem Estilística*: a relação dbo:stylisticOrigin representa as raízes, os estilos musicais historicamente anteriores, que evoluíram, foram adaptados e, a partir dos quais, um novo estilo ou gênero foi originado. O gênero *Synthpop* (também conhecido como Technopop) surgiu em parte a partir do gênero Punk rock, que abandonou a ênfase da autenticidade do punk, e adotou características da música eletrônica e do New Wave;

- *Gênero Derivado*: um gênero derivado dbo:derivate é uma criação expressiva de um novo gênero que inclui os principais elementos dos gêneros originais. Adaptações cinematográficas e arranjos musicais são tipos comuns de criações derivadas. Por exemplo, *Synthpop* (também conhecido como technopop) é derivado do rock e dos gêneros Eletrônica e Glam rock.

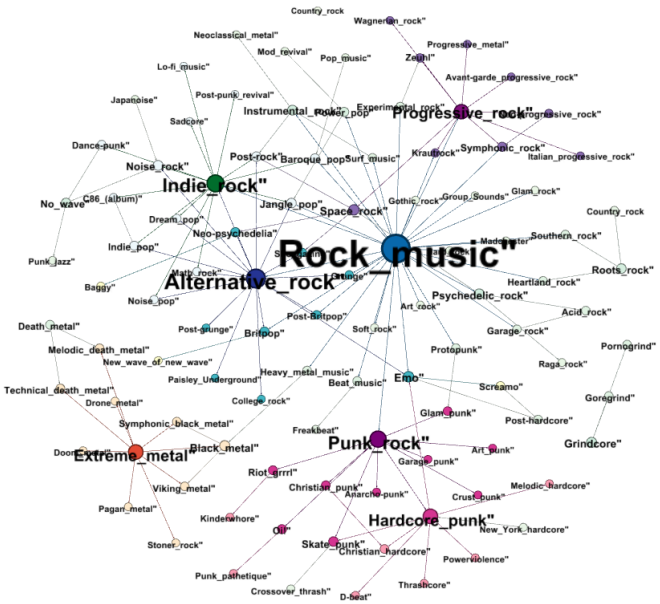

Figura 4.5: Sub-grafo da base de LOD DBpedia, onde cada vértice representa um gênero musical e cada aresta uma relação de "tipo_de", nela podemos ver o claramente gênero Rock e suas subdivisões. O algoritmo de PageRank foi utilizado para destacar os gêneros mais conectados no grafo.

Tópicos (Subject)

Representados pelo objeto *dct:subject*, um *subject* é tipicamente uma palavra-chave, uma frase ou classificação que categoriza um objeto na DBpedia. Por exemplo, a banda *Imperied* tem o tópico Grupo Musical Sueco (Swedish_musical_groups) que representa o grupo de todas as bandas suecas da DBpedia. No total, foram retornados 31.874 diferentes tipos de tópicos relacionados aos artistas da base de dados. Verificou-se que 96% (22.445) dos artistas possuem tópicos que os descrevem. Em média, um artista possui 5 Tópicos associados a ele e, no máximo, existem 97 tópicos para um artista.

Tabela 4.3: Tópicos mais populares da base de dados

Tópico	Soma de todos os scores por tópico (%)
Musical_quartets	0.015594066
American_indie_rock_groups	0.013729920
Grammy_Award-winning_artists	0.013272062
Musical_quintets	0.012670162
Brit_Award_winners	0.008079869

Os Tópicos na DBpedia são estruturados hierarquicamente por meio de relações do tipo skos:broader, responsáveis por estabelecer relações de generalização entre os tópicos disponíveis. Estes podem ser representados em uma árvore taxonômica de atributos.

Origem (Origin)

O objeto dbp:origin representa a cidade ou país de onde aquele artista ou banda foi originada. A banda *Imperied*, por exemplo, teve origem em Estocolmo (dbr:Stockholm), Suécia. Na base de dados, existem 7.355 diferentes origens para os 29.498 (85%) artistas que possuem essa relação.

Os atributos de origem na DBpedia apresentam relações relacionadas às informações geográficas com os tipos "dbo:country"e "dbo:continent"e aspectos culturais, como "dbo:language".

Tabela 4.4: Origens mais populares da base de dados

Origem	Soma de todos os scores por origem (%)
United States	0.09381078
Germany	0.03045766
London, England	0.02901604
England	0.02867098
Japan	0.02038378

4.3 Considerações Finais

A base de dados escolhida (Last.FM) apresenta um domínio interessante e fértil para a exploração da diversidade em SR. A base de dados LOD utilizada disponibiliza um rico conjunto de relacionamentos e conteúdo utilizado neste trabalho. Entretanto, devem-se elencar alguns problemas identificados na base do DBpedia, tal como:

- *Confiança na Informação*: as informações contidas na Wikipédia são geradas colaborativamente por meio de uma comunidade e é possível que existam problemas de dados faltantes, informações erradas ou desatualizadas para algumas páginas;
- *Extração de Dados*: pode haver alguns problemas no processo de extração e transformação em RDF da Wikipédia para o DBpedia. Além de dados faltantes, pode ocorrer problemas de dados desatualizados;
- *Dados Duplicados*: a DBpedia (e também a Wikipédia) define 2.887 diferentes gêneros musicais, onde 330 deles contêm rock em seus descritores. "Rock and roll music"é representado por 18 diferentes URIs e nem todas elas possuem o relacionamento de igualdade "same_as".

Capítulo 5

Avaliação Experimental

Este capítulo apresenta a metodologia experimental escolhida e os algoritmos selecionados, assim como a discussão acerca dos resultados empíricos obtidos através da avaliação do algoritmo proposto em comparação a outras soluções identificadas na literatura. As partes fundamentais no desenvolvimento do framework de avaliação foram codificadas na linguagem de programação Java 8, $C\#$ foi utilizado na geração de listas de recomendações para os usuários e Python foi usado na coleta de dados do Last.FM. Além do algoritmo proposto, LOD-Diversification, baseado em grafos, outros cinco algoritmos de diversificação foram implementados para avaliação e comparação com o algoritmo proposto.

5.1 Metodologia

A pesquisa realizada é do tipo experimental e tem como finalidade principal comparar o comportamento dos algoritmos de recomendação do estado-da-arte e baselines, descritos na seção 5.1.1, com o LOD-Diversification, a partir de métricas de acurácia e diversidade utilizando a base de dados de artistas musicais coletada do Last.FM. De um modo geral, as perguntas de pesquisa que queremos responder são as seguintes:

- **P1** - Os algoritmos de diversificação do estado-da-arte (MRR, ESWC14, IA-Select), os baselines (MaxDiversity, Random) e o algoritmo proposto (LOD-Diversification) apresentam diferenças em relação às métricas de diversidade (ILD, S-Recall)?
- **P2** - Os algoritmos de diversificação estado-da-arte (MRR, ESWC14, IA-Select), os

baselines (MaxDiversity, Random) e o algoritmo proposto (LOD-Diversification) apresentam diferenças em relação às métricas de acurácia (NDCG, F-Measure)?

- **P3** - Os algoritmos de diversificação estado-da-arte (MRR, ESWC14, IA-Select), os baselines (MaxDiversity, Random) e o algoritmo proposto (LOD-Diversification) apresentam diferenças em relação às métricas de acurácia/diversidade (EILD, $\alpha nDCG$)?

5.1.1 Design Experimental

Com o objetivo de comparar os resultados obtidos e apresentá-los da melhor forma possível, um design fatorial completo com 5 replicações foi escolhido para guiar o experimento. Primeiramente, são definidos os melhores resultados obtidos pelos SR tradicionais, variando-se os parâmetros de entrada para os algoritmos de recomendação base UserKNN (número máximo de vizinhos) e o BPRMF (número de fatores latentes e taxa de aprendizado), descritos na seção 5.1.1, utilizando o método de *Grid Search*, que será descrito na Seção abaixo. Os algoritmos *MostPopular* e *Random* não foram citados pois não têm parâmetros. Selecionado o melhor resultado (para a métrica nDCG), para cada algoritmo, serão aplicadas as técnicas de diversificação e novamente o *Grid Search* escolherá o melhor ajuste de parâmetro otimizando a métrica de EILD.

Ajuste de Hiper-parâmetro

Uma vez que a técnica de validação cruzada é utilizada para generalizar o desempenho de um algoritmo, o processo de ajuste de hiper-parâmetro (*Hyperparameter Tuning*) é necessário para suprimir um problema da aprendizagem de máquina, no sentido de otimizar uma função objetivo e encontrar o melhor conjunto de parâmetros para uma dada métrica de avaliação como, por exemplo, para encontrar a melhor combinação de parâmetros para os algoritmos de diversificação, utilizamos a métrica EILD que considera tando a diversidade quanto a acurácia. Não se pode deliberadamente utilizar o conjunto de teste para o processo de ajuste de hiper-parâmetro, pois em SR reais não se têm inicialmente as informações futuras para poder ajustar os parâmetros de um modelo. Na literatura de aprendizagem de máquina, o conjunto de teste é uma parte separada dos dados que são utilizados somente para avaliar os parâmetros finais de um algoritmo, produzidos pelo processo de avaliação. Isso garante que

nenhuma informação no conjunto de treinamento/validação é utilizada para a otimização dos parâmetros do algoritmo e produz um resultado com imparcialidade.

Para a avaliação dos algoritmos de recomendação e diversificação, foi utilizado o método de *Grid Search* [22], que consiste em utilizar o conhecimento sobre o problema para identificar os melhores intervalos para os hiper-parâmetros. Em seguida, o algoritmo *Grid Search* seleciona os parâmetros a partir de vários pontos desses intervalos, geralmente distribuídos uniformemente. O ajuste de hiper-parâmetro treina um algoritmo usando cada combinação de parâmetros e selecionando a combinação com melhor desempenho. Alternativamente, é possível repetir a busca em um domínio mais específico, centrado em torno dos parâmetros que executam os melhores valores obtidos na função objetivo.

Algoritmos Utilizados

Recomendações base Não-Personalizadas

Para recomendações não-personalizadas de artistas musicais, foi utilizado o algoritmo *Most Popular*, que ranqueia os artistas de acordo com sua popularidade, ou seja, os itens são ponderados de acordo com sua frequência, i.e., quantidade de vezes que eles foram ouvidos no passado. Também foi utilizado o baseline Random, que seleciona e ranqueia os artistas candidatos aleatoriamente.

Recomendações base Personalizadas

Os demais algoritmos geram funções de ranqueamento personalizadas. O algoritmo *k-Nearest Neighbour* (kNN) é um classificador que busca os *k* elementos do conjunto de treinamento mais similares com um elemento alvo. Estes *k* elementos são chamados de k-vizinhos mais próximos. Verificam-se quais são as classes desses k vizinhos e a classe mais frequente é atribuída à classe do elemento desconhecido.

A recomendação do algoritmo *User-based kNN* primeiramente computa uma matriz de similaridades entre todos os usuários baseando-se na correlação dos itens na base de treino. Durante a recomendação, o algoritmo *kNN* busca os k usuários mais similares com o usuário alvo e ordena os artistas candidatos a partir das preferências dos k usuários selecionados. Nos experimentos, o valor k variou entre 50, 70 e 100 e o melhor resultado foi o valor de $k = 100$.

O terceiro algoritmo utilizado como base foi o *BPRMF*, modelo de fatoração de matrizes

que otimiza a função de ranqueamento *Bayesian Personalized Ranking* (BPR). A otimização dessa função é matematicamente análoga à otimização da métrica AUC, que pode ser vista como uma métrica de ranking. Esse algoritmo possui dois hiper-parâmetros principais que foram variados: a quantidade de fatores latentes (70 e 100) e a taxa de aprendizado (0.01, 0.05 e 0.1).

Métodos de Diversificação Comparados Para comparar os resultados da abordagem proposta, foram usados dois algoritmos de diversificação gulosos e dois baselines. Os diversificadores reordenam os Top@50 itens devolvidos pelos SR considerados estado-da-arte e, para cada usuário, os algoritmos retornam os top@30/top@20 itens de uma nova lista já diversificada. O primeiro algoritmo é Relevância Máxima Marginal (MMR) [Carbonell et al. (1998)], com a seguinte função objetivo:

$$(1 - \lambda)\widehat{r}_{norm}(u, i) + \lambda max_{j \in S}(1 - sim(i, j)) \tag{5.1}$$

Onde:

- $\widehat{r}_{norm}(u, i)$ é o *rating* normalizado entre [0,1] de um usuário u para um item i.

O componente de diversidade da função objetivo é controlado por λ através da combinação linear da classificação do novo item, normalizada entre [0, 1], e a distância máxima entre o item e a nova lista, calculada pela similaridade do cosseno (sem similaridade negativo) de seus itens. Ele utiliza a variação adaptativa do algoritmo MMR usar uma heurística para modelar diversidade propensão do usuário para os atributos de cada item, usando o tamanho de entropia e perfil para classificar o usuário em um dos quatro quadrantes. Para cada quadrante, foi definido manualmente um peso que representa o desconto em função de similaridade para usuários em que quadrante para um atributo. Outro algoritmo de reranqueamento é o *Intent Aware Select* (IA-Select) [2] que gulosamente reorganiza a lista pela pontuação normalizada do produto e o recurso de distribuição e seguintes do usuário e o item com a distribuição anterior da mesma função nos itens re-classificados anteriores. IA-Select maximiza a seguinte função objetivo:

$$\sum_{f \in F} p(f|u)p(f|i)\widehat{r}_{norm}(u, i) \prod_{j \in S}((1 - p(f|j)\widehat{r}_{norm}(u, j)) \tag{5.2}$$

Onde $p(f|u)$ representa a distribuição do atributo no usuário e $p(f|i)$ a distribuição por item. Para efeito de comparação, também foram utilizados os algoritmos baselines Random (RND), que aleatoriamente re-ranqueia a lista de recomendações.

No recente desafio da European Semantic Web Conference (ESWC 2014), o objetivo foi utilizar *Linked Open Data* (LOD) para melhorar a diversidade das recomendações. Petar et al. [29], o vencedor do desafio ESWC para o ano de 2014, considerou a combinação de ambos, precisão (F-measure) e diversidade (ILD) da lista de recomendações. Esse método funciona de forma semelhante à diversificação gulosa, porém só diversifica a lista a partir da posição *m* da lista original. O valor da variável *m* serve como um fator que controla o *trade-off* entre precisão e diversidade. Quanto maior o valor de *m*, mais a nova lista será parecida com a lista original. Para diversificar a lista abaixo da posição *m*, o método insere itens que não compartilham atributos com os m primeiros itens da lista. Para melhorar o método, o presente trabalho diversifica o restante da lista com uma abordagem gulosa.

Para efeito de comparação, também usamos o baseline de diversificação aleatória (RND), que aleatoriamente re-ranqueia a lista de recomendações e a diversificação MaxDiversity, que de forma gulosa obtém o item mais diversificado até que a lista de recomendações seja preenchida. O algoritmo MaxDiversity é obtido a partir da implementação do MMR, onde o fator de diversificação λ é sempre 1, ou seja, a lista retornada apresenta a máxima diversificação possível para o algoritmo MMR.

5.1.2 Ameaças à Validade

A exposição completa e detalhada das ameaças à validade interna, externa e de construção é um fator de qualidade importante para descrever a metodologia de um avaliação experimental, avaliar as suas falhas e possíveis melhorias. A seguir, elencamos e descrevemos algumas ameaças a validade identificadas no processo de avaliação desenvolvido neste trabalho.

- É possível que os níveis dos fatores selecionados (quantidade mínima de usuários e porcentagem dos dados usada na base de treino) não sejam suficientes para observar diferenças significativas de eficácia e/ou desempenho entre os algoritmos. Desse modo, há uma ameaça à validade de construção causada pela confusão entre constructos e seus níveis;

- Não foram detectadas ameaças à validade externa, já que as bases utilizadas aparentam ser representativas para o escopo do trabalho, dada a diversidade de gêneros e tipos de artistas musicais ouvidos pelos usuários;

- A avaliação da utilidade da diversidade para os SR é difícil de se realizar de maneira *offline*, pois, após um usuário receber uma recomendação diversa, a recomendação pode influenciar o seu gosto. Sendo assim, a avaliação *offline* da diversidade de SR seria uma ameaça a sua validade;

- Deve-se garantir a heterogeneidade randômica das unidades experimentais, pois estas podem interferir mais nos resultados finais do que os próprios algoritmos alvo de estudo.

5.1.3 Ferramental de Experimentação

Para facilitar o desenvolvimento desse experimento foram utilizados *frameworks* de código aberto, são eles:

- **MyMediaLite**: o MyMediaLite [16] é um framework e uma biblioteca pública com algoritmos de SR, desenvolvida em diversas linguagens de programação, para a plataforma .NET. A biblioteca fornece implementações do estado-da-arte nos dois cenários mais comuns de recomendação na literatura, predição de notas e recomendação de itens. O MyMediaLite contém rotinas que computam métricas de avaliação para a predição de notas (*ratings*) em SR, como o *Root Mean Square Error* (RMSE) e o *Mean Average Error* (MAE) e na tarefa de recomendação de itens implementa a *Area Under the ROC Curve* (AUC), *Mean Average Precision* (MAP) e *Normalized Discounted Cumulative Gain* (NDCG). Possui também ferramentas que facilitam o desenvolvimento, implantação e operação de sistemas de recomendação reais. Neste trabalho, o framework foi utilizado para o treinamento e geração de listas de recomendações para os usuários;

- **RankSys**: RankSys [1] é um framework desenvolvido na linguagem Java 8 e possui implementações de várias técnicas e métricas de avaliação de SR. O framework inclui

[1] https://github.com/RankSys/RankSys

um foco na avaliação e melhoria da novidade e diversidade. O seu framework RankSys foi utilizado para avaliar algumas métricas de diversidade como $\alpha ndcg$ e EILD.

- **R, versão 3.1.0**: R é uma linguagem de programação amplamente utilizado para análises estatísticas, mineração e visualização de dados. Oferece diversos pacotes que são bibliotecas para funções variadas em diversas áreas de estudo.

5.2 Resultados

Esta seção é dedicada à apresentação e a discussão de resultados experimentais de modo a permitir que se possa avaliar a contribuição do trabalho, o alcance dos seus objetivos e a verificação das hipóteses de pesquisa. Primeiramente iremos discutir os resultados e a influência dos dois parâmetros do algoritmo LOD-Diversification, a profundidade máxima ($depth$) atingida no grafo de atributos e o fator de diversificação (λ) com e sem a adição da propensão do usuário. Logo após é realizada a comparação do algoritmo proposto com os demais algoritmos estado-da-arte, seus resultados são discutidos e analisados. Por fim listamos as limitações do trabalho e as considerações finais.

5.2.1 LOD-Diversification

A profundidade máxima (Seção 3.2.1) funciona como condição de parada na busca e na distribuição de pesos para os novos atributos extraídos da base de dados de LOD. Na Figura 5.1(a) podemos perceber a influência do $depth$ no desempenho da diversidade e acurácia do algoritmo, onde o melhor valor para a métrica de EILD é quando o $depth = 3$. Quando $depth >= 6$ o algoritmo começa a adicionar atributos muito distantes dos originais onde há uma perda significativa na acurácia e aumento da diversidade.

A Figura 5.1(b) demonstra a avaliação do LOD-Diversification variando o seu parâmetro (λ), para as métricas de ILD e EILD. Podemos perceber que o algoritmo o ILD aumentando, onde o seu valor máximo é atingido quando o $\lambda = 1.0$, porém diversidade e acurácia apresentam um melhor balanceamento entre si quando o $\lambda = 0.5$, onde o algoritmo apresenta o seu melhor resultado para a métrica de EILD.

Por fim, na Figura 5.2 podemos ver a distribuição da diversidade apresentada nos per-

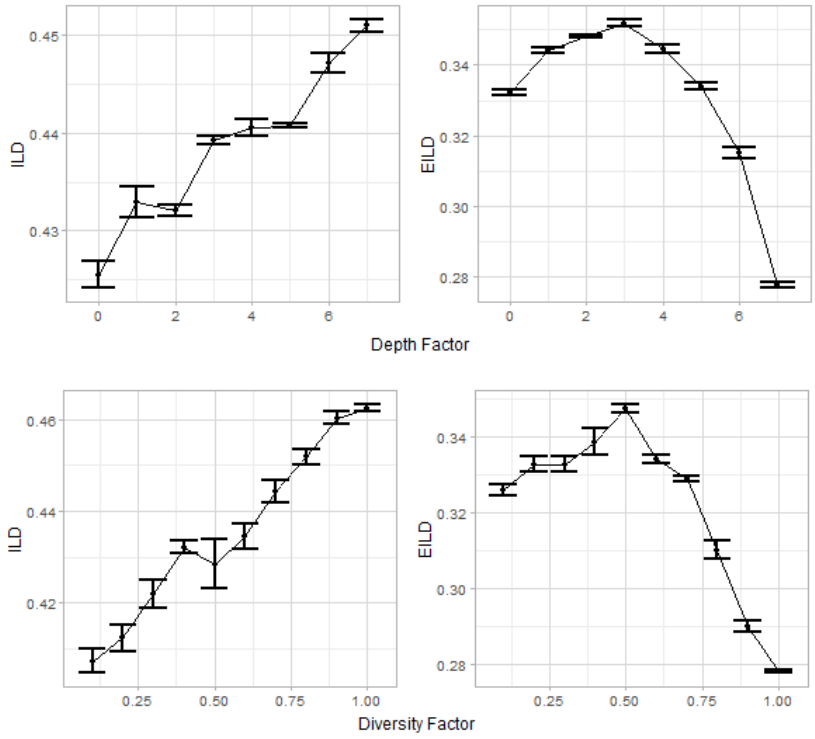

Figura 5.1: a) Fator de Depth para o LOD-Diversification; b) Fator de diversificação do algoritmo LOD-Diversification.

fis dos usuários da nossa base de dados do LastFM para gêneros musicais, esse é o valor utilizado na Equação 3.4 para inferir propensão à diversidade de cada usuário e por fim calcular a partir dele o fator de diversificação utilizado. Como podemos perceber o fator de diversificação melhora o trade-off entre a diversidade e a acurácia, o ganho da função após o algoritmo selecionar o melhor fator diversificação para cada usuário em comparação aos resultados apresentados a manter o fator de diversificação fixo em 0,5.

O fator adaptativo do algoritmo LOD-Diversificaion consegue melhorar o desempenho nos casos onde a propensão do usuário está longe da mediana, como nos casos onde a diversificação apresentada pelo perfil do usuário um ou dois desvios padrões da mediana. Nos casos em que as listas recomendadas são muito diversificadas acabam compensando as listas pouco diversificadas geradas pelo algoritmo e em contraposição a acurácia do algoritmo aumenta, como podemos perceber no EILD.

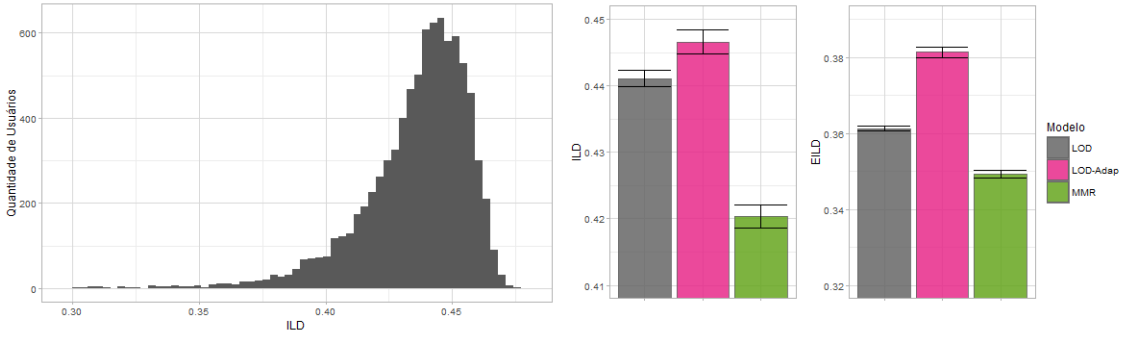

Figura 5.2: Trade-off entre Diversidade e Acurácia para o LOD-Diversification.

5.2.2 Apresentação e Análise dos Resultados

Como mencionado na seção 5.1.1, inicialmente o Grid Search foi executado para os três algoritmos estado-da-arte de SR e dois baselines para a base de dados do Last.FM. Os melhores resultados dos algoritmos em termos do nDCG, S-Recall e ILD, os Top 50 artistas recomendados estão sumarizados na Figura 5.3. Os valores de ILD que avalia a diversidade das listas de recomendação base foram avaliado a partir das informações de gêneros, tópicos e origem dos artistas musicais recomendados.

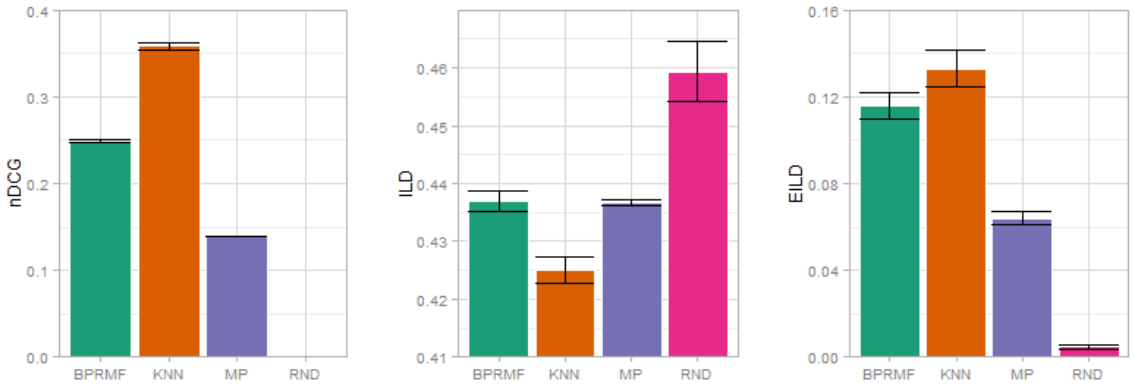

Figura 5.3: Apresentação dos resultados dos recomendadores para as métricas de Diversidade e Acurácia aplicadas à base do Last.FM.

Analisando cada algoritmo separadamente vê-se inicialmente que o algoritmo *Random* obteve o resultado esperado, de pior desempenho entre os algoritmos avaliados. Esse pode ser considerado o pior baseline em relação à acurácia, porém a sua diversidade Intra-List

(ILD) é a maior entre os algoritmos candidatos, pois existe uma quantidade razoável de atributos entre os artistas musicais, esses randomicamente escolhidos apresentam um alta probabilidade de retornar uma lista altamente diversa.

Em seguida, o algoritmo *MostPopular* consegue obter um melhor desempenho na métrica de nDCG, em relação ao baseline anterior. Por conta do viés da popularidade dos artistas na base de dados do Last.FM, o algoritmo consegue acertar artistas musicais simplesmente recomendando os mais populares. Porém, esse não consegue competir em acurácia com os algoritmos personalizados. Na diversidade, o ILD retornado na avaliação do algoritmo *MostPopular* apresenta uma interseção entre o seu intervalo de confiança com o do algoritmo BPRMF. Realizando o teste estatístico (T-Student pareado) com 95% de confiança, percebeu-se que *MostPopular* supera BPRMF nas métricas de diversidade. O representante da fatoração de matrizes supera o kNN em diversidade mas apresenta um desempenho inferior nas métricas de acurácia e S-Recall.

Em seguida, o algoritmo kNN foi selecionado como o melhor indicado para a aplicação da diversificação da lista da recomendação, pois esse apresenta o melhor resultado em acurácia e o pior representante em diversidade. Foram aplicados os algoritmos de diversificação baselines e estado-da-arte para vários atributos extraídos dos artistas musicais através da base de LOD da DBpedia: Gênero Musical, Tópicos e Origem dos artistas. Para cada um deles serão analisadas as métricas de acurácia, diversidade e métricas que avaliam as duas dimensões em conjunto.

Utilizando os dados para o cálculo da diversidade e as relações de subgênero, fusão de gênero, origem estilística e gênero derivado para o LOD-Diversification, determinaram-se os pesos k e o valor de profundidade no grafo, de depth através do Grid search. Como dito anteriormente, os algoritmos comparados são: MMR adaptative, IA-Select, ESWC14, além dos baselines MaxDiversity e Random. A lista de recomendações original também foi considerada como baseline para uma melhor comparação. A Figura 5.4 mostra o resultado da diversificação de acordo com as métricas de diversidade ILD e SRecall.

O algoritmo *MaxDiversity* consegue atingir na mediana um valor de 0.47 de ILD, o melhor desempenho em relação aos seus concorrentes. Contudo, ainda analisando a diversidade do *MaxDiversity* percebe-se que ele, apesar de recomendar uma lista diversificada em função da distância e dissimilaridade entre os itens da lista apresenta um baixa cobertura, represen-

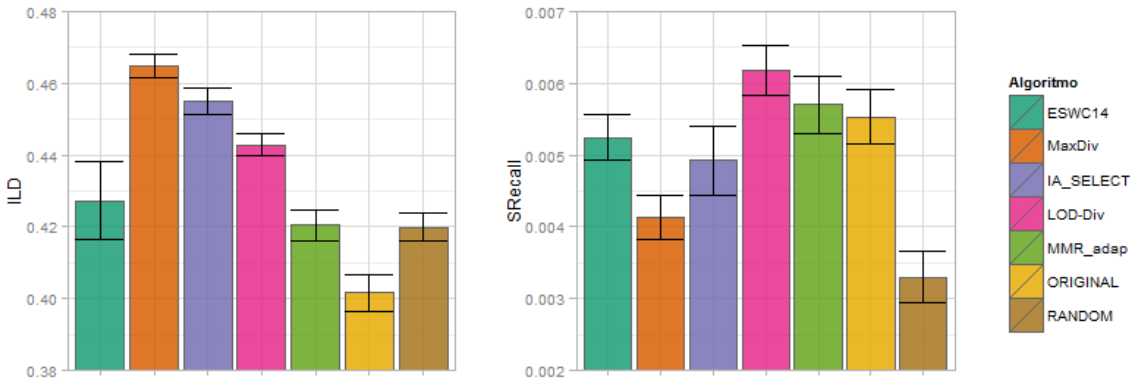

Figura 5.4: Apresentação dos resultados dos diversificadores para as métricas de Diversidade.

tada pelo SRecall. Em contrapartida, o algoritmo adaptativo MMR apresenta um desempenho inverso, com um bom desempenho na cobertura SRecall e um baixo desempenho na métrica ILD, estando acima somente do Random e da lista de recomendações original. Os algoritmos IA-select e LOD-Diversification apresentaram bom desempenho em ambas as métricas de diversidade, porém, como será visto adiante, o algoritmo IA-Select não possui um parâmetro que regule o *trade-off* entre a acurácia e diversidade e acaba por perder em acurácia. O algoritmo LOD-Diversification se comporta de forma mais dirigida aos dados dos usuários, apresentando um ILD mais próximo ao encontrado na base de teste, ou seja, diversificando a lista de recomendações de maneira diferenciada para cada usuário dependendo da propensão apresentada na base de treino, tal como ocorre com a adaptação do MMR de Castells et al. [37]. Percebe-se em ambas as métricas de diversidade que o algoritmo randômico apresenta o pior desempenho entre as outras métricas, só superando a lista original em ILD. Isso ocorre pois na maioria das listas há uma predominância por alguns certos tipos de atributos.

Com o enfoque na acurácia, na Figura 5.5, vê-se que os baselines *MaxDiversity* e *Random* apresentam uma degradação muito grande de desempenho nas métricas avaliadas em comparação com a lista original. Com a abordagem de diversificar ao máximo a lista em função da dissimilaridade, o *MaxDiversity* acaba por selecionar itens que não são de muito interesse para o usuário e prejudicando a ordenação gerada pelo algoritmo de recomendação

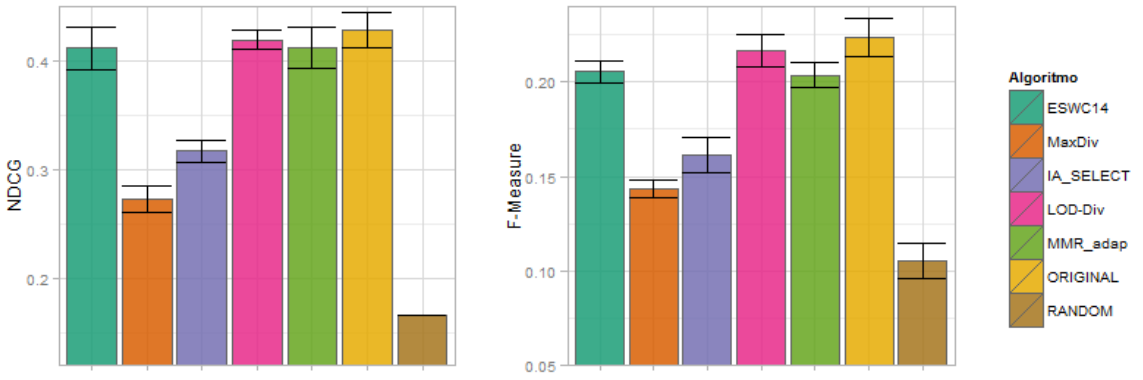

Figura 5.5: Apresentação dos resultados dos diversificadores para as métricas de acurácia.

original, o mesmo ocorre com algoritmo randômico. Como dito anteriormente o IA-Select diversifica a lista sem se importar em ajustar o *trade-off* entre diversidade e acurácia de modo que ele diversifica excessivamente a lista de recomendações. O algoritmo vencedor do ESWC 2014 mantém os *n* primeiros itens da lista e diversifica os demais. Essa abordagem pode ser percebida na diferença entre o seu nDCG e F-Measure, pois como o algoritmo não modifica a posição dos primeiros itens, consegue conservar um nDCG alto enquanto que no F-Measure podemos perceber a degradação da acurácia. Um ponto negativo para essa abordagem é o fato dos itens do topo da lista apresentarem uma baixa diversidade, que seria igual a da lista original, porém esse quesito não foi avaliado neste trabalho. Para os outros algoritmos de diversificação pode-se perceber uma diminuição suave da acurácia em relação à lista original. Esse *trade-off* pode ser melhor percebido utilizando as métricas que combinam a diversidade e a acurácia (EILD e α-nDCG). Pode-se ver o LOD-Diversification competindo com o MMR adaptative e o ESWC 14. Ao utilizar o T-teste pareado para os três algoritmos de diversificação com 95% de confiança, não é possível afirmar que o desempenho do MMR é maior ou menor do que o ESWC14 em ambas as métricas de acurácia. Em comparação com o LOD-Diversification, o algoritmo proposto supera ambos com o p-valor de 0.04563%.

Por fim, tem-se a avaliação em fator das métricas de avaliação que consideram a acurácia, através do ranking e a diversidade da recomendação. Novamente, os algoritmos baselines apresentam os piores valores de EILD e α-nDCG. A alta diversidade nas listas geradas pelo MaxDiversity não compensa a baixa acurácia apresentada. O IA-Select não consegue tratar

o *trade-off* de uma maneira otimizada. O quadro da Figura 5.6 se repete nas métricas de diversidade e acurácia. Quando analisamos ambas as dimensões das listas de recomendação vemos que o IA-Select empatou com a lista original em ambas as métricas de avaliação e os três melhores candidatos são o MMR adaptative, ESWC14 e o LOD-Diversification. Novamente realizamos testes estatísticos para avaliar se há diferenças entre os resultados de forma pareada. Vê-se que os algoritmos MMR e Lod-Diversification superam o ESWC14. Além disso, o LOD-Diversification consegue ultrapassar o MMR adaptative pela diferença de 0.07 entre suas medianas.

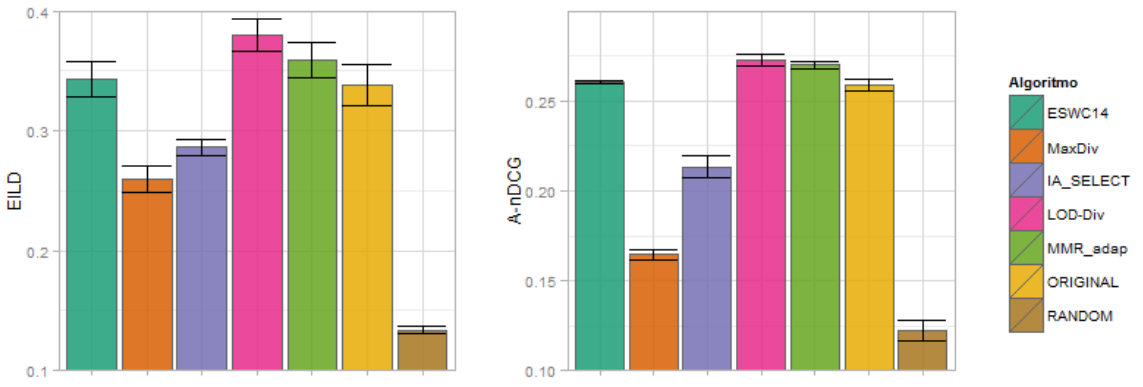

Figura 5.6: Apresentação dos resultados dos diversificadores para as métricas de Diversidade e Acurácia.

5.2.3 Limitações

Como o algoritmo proposto é principalmente baseado em conteúdo e nas relações existentes entre os atributos disponíveis para os itens, são elencadas algumas limitações e problemas existentes na aplicação e avaliação do LOD-Diversification:

- Falta de dados: um dos motivos da escolha de artistas musicais como itens ao invés de faixas de músicas na avaliação experimental deste trabalho foi o fato das bases LOD terem mais de 200 mil objetos do tipo *MusicalArtist* e o fato do mapeamento entre o Last.FM e a DBpedia ser relativamente simples com um mapeamento entre o seus labels;

- Falta de relações relevantes: relações que não são relevantes para o domínio como, por exemplo, na recomendação de artistas musicais, utilizar relações de atributos físicos como altura e cor dos olhos não ajudam ao algoritmo;

- Homogeneidade de atributos: a utilização de atributos muito homogêneo e enviesado não agrega nenhuma função para o algoritmo LOD-Diversification e fazem com que o diversificador recomende itens que o usuário provavelmente não irá consumir, prejudicando assim a acurácia da recomendação;

- Falta de validação experimental: uma validação rigorosa do algoritmo proposto só é possível através de avaliação online de usuários reais e a qualidade das notas obtidas a partir desses experimentos. Tal validação irá permitir avaliar a qualidade real do LOD-Diversification, identificando e validando os tipos de usuários que gostam de recomendações diversas ou não, a utilidade da lista em descobrir novos itens que os usuários não descobririam por si só, o que é a principal função de um recomendador. Esse tipo de análise não pode ser realizada utilizando avaliação offline.

5.2.4 Considerações Finais

O design utilizado na experimentação, o ferramental utilizado, suas ameaças à validade e limitações foram apresentadas e discutidas neste capítulo. Foram mostrados ainda os resultados da avaliação experimental realizada, nos quais o algoritmo LOD-Diversification se mostrou competitivo em todas as métricas de avaliação, tanto na diversidade quando acurácia e nas métricas híbridas (ac/div). O algoritmo apresenta um tipo de abordagem capaz de ser utilizada em diversos domínios e possui um fator de cobertura e exploração que demostraram ser promissores. O *trade-off* entre diversidade e acurácia foi tratado de modo personalizado, tal qual o MMR adaptative faz com entropia e tamanho do perfil. Nesse quesito, podemos ver uma diversidade e cobertura que supera os algoritmos avaliados e um pequeno decaimento na acurácia em comparação com a lista original, tando na métrica de ranking quanto na de F-Measure.

Capítulo 6

Conclusão

Este capítulo resume as contribuições deste trabalho de pesquisa e introduz algumas oportunidades para futuras pesquisas relacionadas à diversidade em SR.

6.1 Resumo

Neste trabalho, um novo algoritmo para diversificação de listas de recomendação para SR de re-ranqueamento foi desenvolvido utilizando uma abordagem baseada em grafos. A abordagem explora as relações semânticas entre os itens disponíveis em bases de dados LOD em diversas dimensões, com o objetivo de melhorar o *trade-off* entre diversidade e acurácia. O algoritmo desenvolvido tenta aprender um melhor ajuste do fator diversidade baseado no perfil do usuário de forma personalizada.

Foi proposto um novo algoritmo de diversificação de listas de recomendação baseado em relações semânticas para diversificação que explora as relações semânticas entre os itens extraídos da LOD em diversas dimensões. O algoritmo conta também com um melhor ajuste entre o perfil do usuário e a diversidade da lista de recomendações identificando o seu nível de ecleticidade ao diversificar suas recomendações, criando assim um melhor ajuste ao *trade-off* entre diversidade e acurácia.

O algoritmo proposto oferece novas possibilidades para explorar as relações entre as diferentes dimensões de atributos relevantes do domínio de dados. Foi realizada uma avaliação com o objetivo de recomendar artistas musicais para os usuários. Para tal, foram coletados históricos de consumo de usuários reais na base de dados do Last.FM. Para o mapeamento

entre artistas musicais na base do Last.FM e do DBpedia foram utilizadas consultas em SPARQL. Os resultados demostram:

- A eficiência do algoritmo proposto comparado com vários algoritmos de estado-da-arte e *baselines* em várias dimensões de atributos dos artistas musicais.
- A avaliação demostrou que o algoritmo LOD-Diversification superou os algoritmos estado-da-arte em métricas que consideram a diversidade, a acurácia e ambas combinadas, como é o caso das métricas de EILD e $\alpha NDCG$.
- O ajuste do nível de diversificação por usuário se demonstrou importante para que o LOD-Diversification mantivesse o nível da acurácia;

6.2 Trabalhos Futuros

Um possível trabalho futuro seria uma análise mais aprofundada sobre os vários tipos de relações semânticas entre os atributos, entre elas a generalização, onde os atributos são organizados em uma árvore taxonômica e os atributos perdem especificidade a medida que se aproximam do nó raiz, estratégia abordada por Ziegler em [42], a especialização onde os atributos, pelo contrário, ganham especificidade e a associação (composição, agregação) onde vários atributos se unem para compor outro, como é o caso do fusão de gêneros musicais ou partes dos atributos formam novos atributos, como é o caso dos relacionamentos de gêneros derivados, discutidos na seção 4.2. Todas elas foram abordadas de um modo único na elaboração deste trabalho. Um algoritmo poderia aprender e recomendar baseado em cada tipo de relacionamento, dependendo do perfil que o usuário apresenta, automatizando a busca dos pesos para as relações entre os atributos.

A exploração e mapeamento de novas bases de dados LOD, com o intuito de enriquecer a base de treinamento do SR e melhor diversificar a lista de recomendações, assim como a avaliação do algoritmo proposto com novas bases de dados de diferentes domínios como, por exemplo, livros, filmes e eventos.

Uma avaliação online poderia ser realizada abordando inclusive métricas de serendipidade e novidade, pois o algoritmo LOD-Diversification descobre novas relações implícitas entre os itens, o que favorece o usuário a receber recomendações com novidade e surpresa.

Além de poder relatar descritivamente porque esse item foi recomendado baseado nas arestas e nos caminhos possíveis entre o item consumido e o item recomendado.

Outro aspecto a ser estudado é o uso do LOD para criação de *playlists* diversificadas através de uma ou diferentes dimensões de categorias musicais. Essa possibilidade permitiria ao algoritmo acrescentar novas maneiras de abordar a conexão entre as músicas em sequência, como é feito hoje na rádio de *streaming* online Pandora[1] na qual os usuários recebem recomendações de músicas e, opcionalmente, oferecem o seu *feedback* explícito (positivo ou negativo) acerca daquela música recomendada.

Outro aspecto poderia ser a inclusão de conteúdo acústico das músicas escutadas pelo ouvinte relacionadas com outros aspectos contextuais, tais como o gênero musical. O *Music Project Geroma* [2] utiliza musicólogos para classificar as músicas de cada gênero e extrair 450 atributos diferentes para cada uma delas. Essas informações são utilizadas no sistema de *streaming* do Pandora para realizar comparações de similaridade entre músicas e a similaridade entre características do conteúdo do áudio, como timbre, tempo e ritmo. Assim, LOD serviria para mapear e criar uma rede de conexões semânticas entre esses diferentes atributos e tornar o cálculo de similaridade mais ajustado.

[1] www.pandora.com/

[2] https://www.pandora.com/about/mgp

Bibliografia

[1] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):896–911, 2012.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

[4] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International journal on semantic web and information systems*, 5(3):1–22, 2009.

[6] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[7] Ben Carterette. An analysis of np-completeness in novelty and diversity ranking. *Information Retrieval*, 14(1):89–106, 2011.

[8] Pablo Castells, Saúl Vargas, and Jun Wang. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. 2011.

[9] Oscar Celma. *Music recommendation*. Springer, 2010.

[10] Òscar Celma and Pedro Cano. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, page 5. ACM, 2008.

[11] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[12] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito. Exploiting the web of data in model-based recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 253–256. ACM, 2012.

[13] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8. ACM, 2012.

[14] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. An analysis of users' propensity toward diversity in recommendations. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 285–288. ACM, 2014.

[15] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[16] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 305–308. ACM, 2011.

[17] Mouzhi Ge, Fatih Gedikli, and Dietmar Jannach. Placing high-diversity items in top-n recommendation lists. In *Workshop chairs*, page 65. Citeseer, 2011.

[18] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[19] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[20] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[21] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, page 210, 2010.

[22] PM Lerman. Fitting segmented regression models by grid search. *Applied Statistics*, pages 77–84, 1980.

[23] Andryw Marques, Nazareno Andrade, and Leandro Balby. Exploring the relation between novelty aspects and preferences in music listening. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 407–412, 2013.

[24] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, New York, NY, USA, 2006. ACM.

[25] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 85–92. ACM, 2013.

[26] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[27] Dmitry Yurievich Pavlov, Alexey Gorodilov, and Cliff A Brunk. Bagboo: a scalable hybrid bagging-the-boosting model. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1897–1900. ACM, 2010.

[28] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. *Recommender systems handbook*, pages 1–35, 2011.

[29] Petar Ristoski, Eneldo Loza Mencía, and Heiko Paulheim. A hybrid multi-strategy recommender system using linked open data. In *Semantic Web Evaluation Challenge*, pages 150–156. Springer, 2014.

[30] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[31] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194, 1998.

[32] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Found. Trends Inf. Retr.*, 9(1):1–90, March 2015.

[33] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. *Search result diversification*. Now Publishers, 2015.

[34] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 595–604. ACM, 2011.

[35] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. *Recommender systems handbook*, pages 257–297, 2011.

[36] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[37] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116. ACM, 2011.

[38] Saúl Vargas and Pablo Castells. Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th Conference on Open Research Areas in*

Information Retrieval, pages 129–136. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[39] Cheng Xiang Zhai, William W Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. ACM, 2003.

[40] Mi Zhang and Neil Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 123–130, New York, NY, USA, 2008. ACM.

[41] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

[42] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

Apêndice A

Descrição da Base de Dados da DBpedia

Esta seção descreve os grafos e atributos extraídos da DBpedia, utilizados no desenvolvimento dos experimentos e validação deste trabalho.

A.1 Subgêneros Musicais

Tabela A.1: Análise do grafo de subgêneros musicais

Gêneros fortemente conectados	480
Grau médio	1,080
Diâmetro	4
Caminho médio	1.0
Número de menores caminhos	736

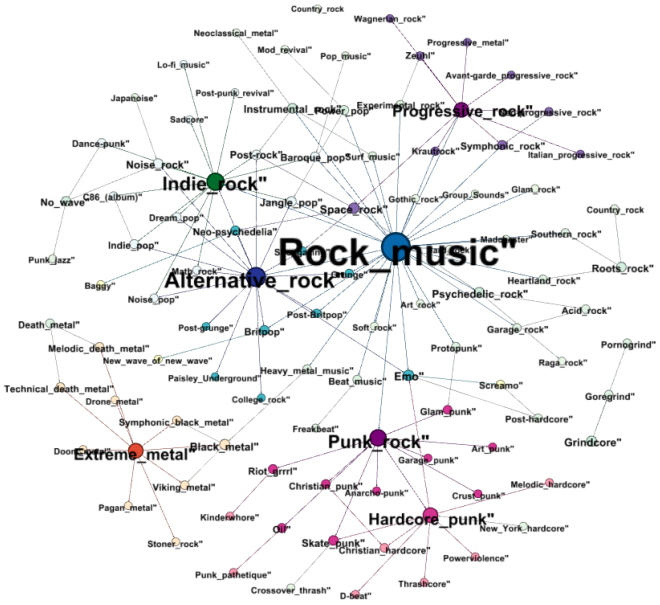

Figura A.1: Sub-árvore da ontologia da DBpedia, cada vértice representa um objeto e cada aresta uma relação de "tipo_de".

A.2 Fusão de Gêneros Musicais

Tabela A.2: Análise do grafo de Fusão de Gêneros

Vértices	390
Arestas	480
Gêneros fortemente conectados	393
Grau médio	2,427
Diâmetro	1
Caminho médio	1.0
Número de menores caminhos	954

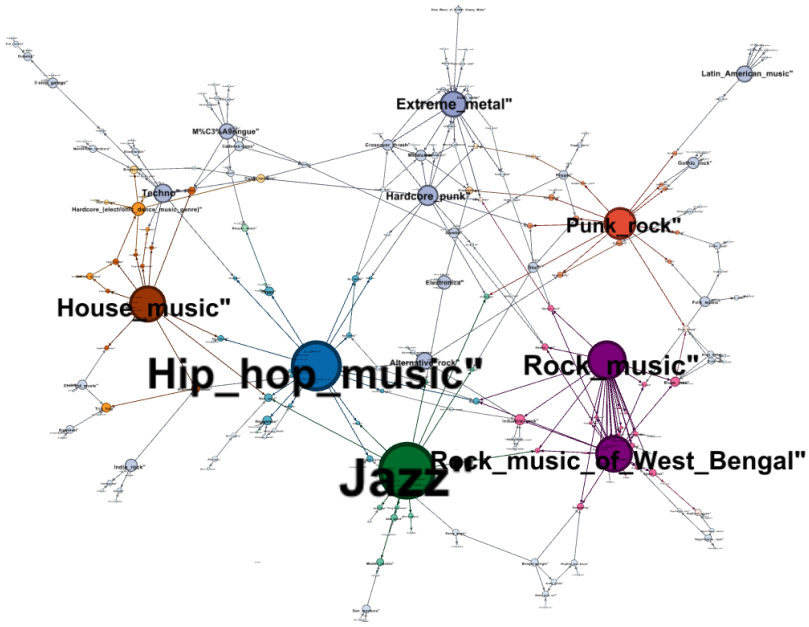

Figura A.2: Grafo da ontologia da DBpedia, cada vértice representa um objeto e cada aresta uma relação de "fusion_genre".

A.3 Gêneros Derivados

Tabela A.3: Análise do grafo de Gêneros Derivados na DBpedia.

Vértices	499
Arestas	760
Gêneros fortemente conectados	355
Diâmetro	9
Caminho médio	.011
Número de menores caminhos	9469
Densidade	0,005

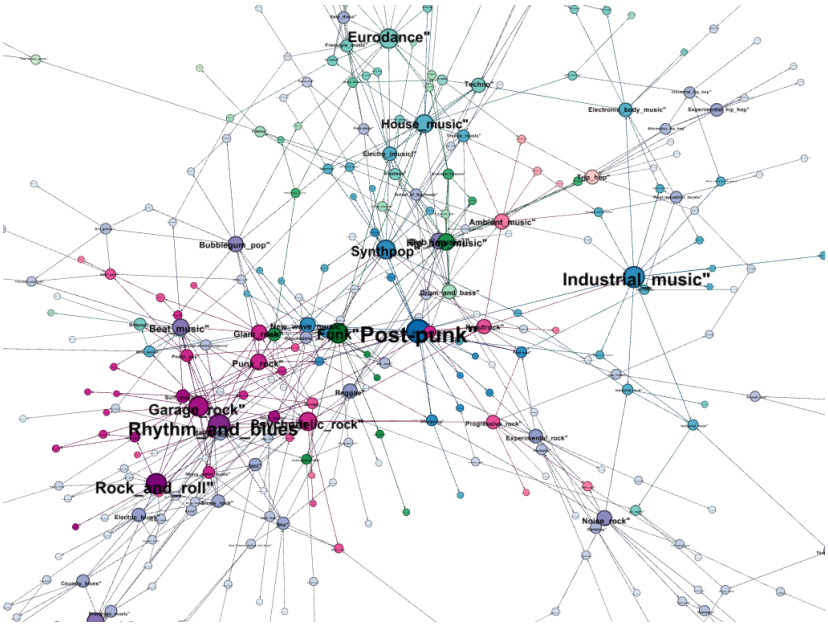

Figura A.3: Grafo da ontologia da DBpedia, cada vértice representa um objeto e cada aresta uma relação de "derivate_genre".

A.4 Grafo de Gêneros e suas origem estilísticas

Tabela A.4: Origens Estilísticas na DBpedia.

Vértices	1038
Arestas	3138
Gêneros fortemente conectados	1017
Grau médio	3,023
Diâmetro	13
Caminho médio	4.05
Número de menores caminhos	74487
Densidade	0,003

A.5 Tópicos de Artistas Musicais

Tabela A.5: Análise do grafo de Tópicos na DBpedia.

Vértices	7220
Arestas	10130
Grau médio	1,402
Diâmetro	1
Radius	0
Caminho médio	1.0
Número de menores caminhos	10130