

Enriquecendo o *dataset* MovieLens com SPARQL*

*Nota: Esse artigo foi escrito para a matéria MATE75, semestre 2019.1

1st Amanda Chagas de Oliveira
Aluna do PGCOMP
Instituto de Matemática
Universidade Federal da Bahia
Salvador, Bahia, Brasil
oliveira.amanda@ufba.br

2nd Frederico Araújo Durão
Prof. do PGCOMP
Instituto de Matemática
Universidade Federal da Bahia
Salvador, Bahia, Brasil
freddurao@dcc.ufba.br

3rd Cássio Vinícius Serafim Prazeres
Prof. do PGCOMP
Instituto de Matemática
Universidade Federal da Bahia
Salvador, Bahia, Brasil
cprazerres@gmail.com

Resumo—Sistemas de Recomendação são utilizados largamente hoje em dia, principalmente nas áreas de *e-commerce* e de entretenimento (distribuição de conteúdo). Em tais sistemas, conhecer bem os itens que serão recomendados é fundamental para realizar uma recomendação precisa e relevante. Para tal, podemos assumir que quanto mais informações sobre os itens soubermos, melhor poderemos classificá-los no futuro. Nesse artigo iremos focar no cenário de filmes, realizaremos o enriquecimento do *dataset* MovieLens¹ utilizando a tecnologia SPARQL.

Index Terms—Sistemas de Recomendação, enriquecimento de *dataset*, SPARQL

I. INTRODUÇÃO

Com tanta informação disponível, é natural que as pessoas sintam dificuldades em encontrar aquela que lhe é útil. Para minimizar a sobrecarga, elas podem obter avaliações de seus pares (amigos, conhecidos, etc) ou fazer pesquisas na Internet, o que já caracteriza um conjunto de informações muito grande e disperso, dificultando a descoberta. Por esse motivo, a informação não é entregue de forma personalizada ao usuário e encontrá-la pode ser uma tarefa bem complexa.

Uma saída são os Sistemas de Recomendação, que filtram a informação para fornecer sugestões para um usuário, como assistir a um filme, ouvir uma música, ler um livro e assim por diante. Neste sentido, um Sistema de Recomendação irá atuar a favor das preferências do usuário. As preferências do usuário são determinadas com base nos itens que o mesmo consumiu no passado. Em geral, esses sistemas retornam ao usuário itens similares aos itens que agradaram esse usuário no passado. Dessa forma, ao enriquecermos o *dataset* com mais informações sobre os itens em si, podemos contribuir nesse cálculo de similaridade entre itens.

II. CONCEITOS

A. Sistemas de Recomendação

Os Sistemas de Recomendação, ao menos aqueles iniciais, eram sistemas bem rudimentares de filtragem e recuperação das informações, podendo apenas recomendar alguns resultados, de acordo com a pesquisa. O primeiro deles foi um sistema experimental de filtragem de imagem, o Tapestry [1], criado por pesquisadores da Xerox Palo Alto Research

Center. O Tapestry foi desenvolvido para atender uma simples motivação: lidar com o aumento do número de e-mails que o centro de pesquisa recebia [2]. A Tapestry apareceu para lidar com filtragem colaborativa (eles foram os pioneiros no uso desse termo), além da filtragem de conteúdo. A primeira, a colaborativa, filtra e arquiva os e-mails a partir do modo como as pessoas, que já os tinham lido, reagiam a tudo isso.

Normalmente, as pessoas acabam sempre dependendo do que é recomendado por outros, seja uma recomendação feita por amigos, reviews de livros e filmes, ou guias de restaurantes. Nesse sentido, o Sistema de Recomendação apoia e intensifica esse processo natural [2]. Os Sistemas de Recomendação, portanto, são técnicas e ferramentas de softwares que oferecem opções de itens úteis para algum tipo de usuário. Nesses mesmos sistemas, o termo “item” é ainda bem genérico, denotando apenas o que é recomendado pelo sistema [3].

Os Sistemas de Recomendação estão por toda parte, sendo muito utilizados na indústria de serviço, no consumo de conteúdo, no comércio eletrônico etc. Em muitas aplicações modernas, esses sistemas também são encontrados, principalmente ao entregar uma coleção enorme de itens, recomendando opções ao usuário final, ao mesmo tempo que utiliza uma combinação de abordagens baseada em conhecimento, conteúdo e filtragem colaborativa [3]. Por exemplo, a Netflix², serviço de streaming de filmes, mostra predições de classificação para cada filme assistido, auxiliando o usuário a escolher qual deles deve ser visto ou não. A Amazon³, uma loja virtual, oferece, na página de um produto, informações que envolvem outras compras feitas pelos usuários que adquiriram aquele mesmo produto [4].

A partir da predição das informações dos itens e usuários, para a realização das recomendações, são utilizadas algumas técnicas de recomendação. Algumas técnicas como a colaborativa e a baseada em conteúdo foram criadas para serem suportes para um Sistema de Recomendação.

1) *Filtragem colaborativa*: A partir do gosto de outros usuários no passado, o sistema, na filtragem colaborativa,

¹MovieLens: <https://grouplens.org/datasets/movielens/>

²Netflix: <https://www.netflix.com>

³Amazon: <http://www.amazon.com/>

recomenda itens ao usuário ativo. A semelhança das preferências entre os usuários é calculada a partir da equivalência do histórico de classificações dos usuários. Essa técnica, a de filtragem colaborativa, é considerada uma das técnicas mais populares nos Sistemas de Recomendação e aquela mais amplamente implementada [3].

2) *Filtragem baseado em conteúdo*: Sistemas que implementam uma estrutura de recomendação, ao menos aquelas baseadas em conteúdo, analisam uma série de documentos e/ou descrições de itens previamente classificados por um usuário, construindo um modelo ou perfil de interesses, fundamentado nas características dos objetos classificados [5]. O perfil é um tipo de representação estruturada das preferências do usuário, adaptado para sugerir itens inéditos e interessantes. O processo de recomendar itens nada mais é do que uma combinação dos atributos do perfil do usuário contra os atributos de um objeto de conteúdo. Como consequência disso, ocorre um julgamento de relevância que representa os níveis de interesse do usuário naquele objeto. Se um perfil descreve, com precisão, as escolhas do usuário, isso é uma grande vantagem para que o acesso à informação seja eficaz. Por exemplo, poderia ser usado para filtrar os resultados de pesquisa, na tentativa de decidir se um usuário tem interesse ou não em uma particular página Web, e, em caso negativo, impedir que seja mostrada [6].

B. DBpedia

O projeto *Linked Open Data*⁴(LOD) surgiu da necessidade de conectar informações que são relacionadas e que já estavam disponibilizadas na Web através do uso de alguns padrões. Foi justamente esse pensamento que deu origem a DBpedia, que é o resultado de um esforço colaborativo da extração de informações e da estrutura dessas informações de página disponibilizadas na Wikipedia⁵. Seu principal objetivo é tornar a informação na Web acessível seguindo padrões propostos pela Web Semântica e pelo LOD, como a estruturação RDF e os *endpoints* de consulta SPARQL.

Esse sistema já transformou milhares de informações da Wikipedia em triplas RDF sobre uma variedade enorme de assuntos, como pessoas, lugares, filmes, livros e muito outros. A DBpedia nos permite fazer perguntas semânticas sofisticadas a Wikipedia, e traz como opção ainda o uso de *links* a outras bases de dados conectadas na Web [7].

III. TRABALHOS RELACIONADOS

Foi realizado um levantamento de trabalhos na área de Sistema de Recomendação que utilizassem algum enriquecimento de dados. Alguns desses trabalhos são:

- **RDF Graph Embeddings for Content-based Recommender Systems [8]**

Nesse trabalho, a autora propõe transformar informações provenientes da linguagem natural, extraídas de um domínio web, em triplas RDF que serão utilizadas em

um Sistema de Recomendação baseado em conteúdo. Nesse artigo, não é comentado sobre o uso de consultas SPARQL, no entanto, existem citações de outros trabalhos relacionados onde o foco é a extração de informações sobre filmes da DBpedia [9]. Foram realizados experimentos de teor avaliativo tanto no contexto de filmes, com o *dataset* MovieLens 1M, como no contexto de livros, com o *dataset* LibraryThing⁶.

- **Towards Semantics-Aware Recommender System: A LOD-Based Approach [10]**

O objetivo desse trabalho é mostrar que, ao adicionar informações semânticas, disponibilizadas na *Linked Open Data*(LOD), aos itens a serem recomendados aos usuários, há uma melhora na eficiência de Sistemas de Recomendação tradicionais de filtragem colaborativa. A autora realiza o enriquecimento dos dados de forma offline e divide o processo em três partes: Enriquecimento, onde são utilizados endpoints SPARQL, *clustering* e geração do modelo de dados. Em termos avaliativos, a acurácia e a qualidade das recomendações foram analisadas. Por fim, nesse artigo foi descrito um experimento realizado com o *dataset* MovieLens enriquecido com dados provenientes da DBpedia.

- **Content Recommendation through Semantic Annotation of User Reviews and Linked Data [11]**

O foco desse trabalho é extrair informações não triviais sobre itens de avaliações(*reviews*) feitas por usuários disponibilizadas na Web. Além disso, são utilizadas informações presentes na DBpedia e na Wikidata para o enriquecimento de dados. Há menção ao uso de consultas SPARQL pelo autor, no entanto, esse ponto não é desenvolvido ou exemplificado. Na seção de avaliação, o trabalho buscou trazer recomendações mais diversificadas e ainda assim relevantes em relação à precisão para os usuários. Foram realizados experimentos no contexto de filmes, *dataset* MovieLens, de livros, *dataset* LibraryThing e de músicas, *dataset* LastFM⁷.

- **Similarity-based Matrix Factorization for Item Cold-Start in Recommender Systems [12]**

Esse trabalho propõe uma nova técnica de fatorização de matriz, uma abordagem utilizada em Sistemas de Recomendação, que incorpora o cálculo da similaridade entre itens utilizando seus metadados, para assim melhor prever os *ratings* dos usuários sobre aqueles itens em questão. O foco principal deste trabalho é atacar o problema do *Cold Start*, que é um dos problemas clássicos em Sistemas de Recomendação. Foram exploradas descrições semânticas dos itens disponibilizadas online, especificamente na DBpedia e por meio de consultas SPARQL. A pesquisa foi avaliada no domínio de filmes, *dataset* MovieLens 100k, e de músicas, *dataset* R1-Yahoo! Music⁸.

- **Extending SPARQL for Recommendations [13]**

⁴LOD: <https://lod-cloud.net/>

⁵Wikipedia: <https://www.wikipedia.org/>

⁶LibraryThing: <https://www.librarything.com/>

⁷LastFM: <http://millionsongdataset.com/lastfm/>

⁸R1-Yahoo! Music: <https://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

Nesse trabalho, é exposto o RecSPARQL, que consiste num Sistema de Recomendação que faça uso do SPARQL em seu processo de recomendação de forma flexível sobre grafos RDF tanto em sistemas baseados em conteúdo quanto em sistemas que utilizam filtragem colaborativa. A Figura 1 é um exemplo de uma consulta retirada desse trabalho e exposto aqui para melhor ilustração.

```

1  RECOMMEND ?user ?user.REC ?movie.REC
2  ?SIMscore ?RATING USING CF
3  WHERE {
4    ?user ml:rates ?personalRating .
5    ?personalRating ml:ratedMovie ?movie .
6    ?personalRating ml:hasRating ?uRating }
7  BASED ON {
8    ?user rdf:type recsparql:User .
9    ?movie rdf:type recsparql:Item .
10   ?uRating rdf:type recsparql:UserRating .
11   ?user ml:rates ?personalRating .
12   ?personalRating ml:ratedMovie ?movie }

```

Figura 1. Exemplo retirado do trabalho [13].

IV. ENRIQUECENDO O *dataset* MOVIELENS COM SPARQL

O enriquecimento de informação do *dataset* MovieLens [14] será utilizado em um Sistema de Recomendação, em um futuro próximo, na tentativa de melhorar o cálculo de similaridade entre os itens do *dataset*. Nesse trabalho, o foco será realizar tal enriquecimento utilizando o SPARQL, será detalhado como o enriquecimento foi feito e quais passos foram necessários para alcançar tal objetivo.

A. Metodologia

Observando o *dataset*, foi identificada a necessidade de realizar um pré-processamento das informações antes mesmo de começar o processo de enriquecimento. O passo-a-passo realizado foi o seguinte:

- 1) Aplicar pré-processamento no *dataset*;
- 2) Implementar a consulta SPARQL para buscar informações dos filmes na DBpedia;
- 3) Adicionar nova coluna com informações no *dataset*;
- 4) Persistir os dados enriquecidos da memória pro disco;
- 5) Avaliar novo *dataset* enriquecido;

movieId	title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
2	Jumanji (1995)	Adventure Children Fantasy
3	Grumpier Old Men (1995)	Comedy Romance
4	Waiting to Exhale (1995)	Comedy Drama Romance
5	Father of the Bride Part II (1995)	Comedy

Figura 2. Representação original do *dataset* MovieLens.

1) **Pré-tratamento do *dataset*:** A Figura 2 representa fielmente como as informações sobre os filmes estão disponibilizadas no *dataset* MovieLens. São três colunas, *movieId*, *title* e *genres*, as quais representam respectivamente o identificador

do filme, o título e os gêneros do mesmo. Observe que junto ao título do filme está sempre presente a informação do ano de lançamento do mesmo. Por tanto, uma melhoria trazida com o pré-processamento foi a realocação dessa informação para uma nova coluna chamada de *year*. Além disso, a coluna referente aos gêneros dos filmes está preenchida originalmente com informações do tipo *string*, o que foi corrigido no pré-processamento e tais informações agora são representadas na estrutura de lista, o que facilita o manuseio computacional dessa informação. A Figura 3 representa o *dataset* após o pré-processamento.

movieId	title	genres	year
1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995
2	Jumanji	[Adventure, Children, Fantasy]	1995
3	Grumpier Old Men	[Comedy, Romance]	1995
4	Waiting to Exhale	[Comedy, Drama, Romance]	1995
5	Father of the Bride Part II	[Comedy]	1995

Figura 3. Representação *dataset* MovieLens após pré-processamento.

2) **Implementação da consulta SPARQL:** O foco da consulta SPARQL implementada nesse trabalho foi buscar na DBpedia o *abstract*(resumo) dos filmes presentes no *dataset*. Essa consulta foi implementada utilizando a linguagem de programação Python⁹ e as bibliotecas Pandas, SPARQLWrapper e JSON dessa linguagem. O Código 1 é referente à consulta implementada. A variável *search* recebe no *script* Python o valor referente aos títulos dos filmes disponíveis no *dataset*.

```

1  sparql.setQuery("""
2
3
4  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5  PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
6  PREFIX dbpprop: <http://dbpedia.org/property/>
7
8  SELECT DISTINCT
9    ?name
10   ?abstract
11   ?alternateTitle
12
13  WHERE {
14    ?instance a <http://dbpedia.org/ontology/Film>.
15    ?instance foaf:name ?name .
16    FILTER REGEX (?name, '^"' + search + '"$', 'i').
17    OPTIONAL {
18      ?instance dbpedia-owl:abstract ?abstract .
19      FILTER (LANG(?abstract) = 'en').
20    }
21    OPTIONAL {
22      ?instance dbpprop:alternateTitle ?alternateTitle
23    }
24  }
25  """)

```

Listing 1. Consulta SPARQL definida com o método *setQuery()*

Todo código fonte e demais materiais produzidos por esse trabalho estão disponibilizados online em: <https://github.com/amandachagas/sparqlmovielens>, para uso aberto e colaborativo a quem possa interessar.

⁹Python: <https://www.python.org/>

3) **Adição da coluna abstract no dataset:** Após buscar a informação na DBpedia sobre o resumo do filme, essa informação precisa ser adicionada ao *dataset*. Para tal, criamos a coluna **abstract** para conter tal informação. Em caso de sucesso na consulta SPARQL, o resultado dessa consulta é adicionado na coluna **abstract** do filme. Além disso, foi realizado um tratamento a falha nos seguintes casos: A consulta realizada não obteve resultado, e o filme não possui resumo na DBpedia.

a) *A consulta sobre o filme não obteve resultado:* Nesse caso, é adicionada na coluna **abstract** do filme a mensagem 'Movie not found on DBpedia by query title.'

b) *O filme não possui resumo na DBpedia:* Nesse caso, é adicionada na coluna **abstract** do filme a mensagem 'Found movie, but there is no ABSTRACT on DBpedia.'

4) **Persistência do dataset enriquecido em disco:** A fim de persistir os dados enriquecidos em disco, utilizamos um método da biblioteca Pandas chamado `to_csv()`. Esse método é utilizado no objeto *dataframe* da biblioteca que utilizamos para manipular os dados. Tal método recebe como parâmetro o nome do arquivo .csv onde o *dataframe* em questão deve ser salvo.

5) **Avaliação do dataset enriquecido:** Para fins avaliativos, optamos por utilizar a métrica de cobertura. Avaliaremos quanta informação foi enriquecida e se essa informação é consistente.

V. RESULTADOS OBTIDOS

O *dataset* utilizado nesse trabalho foi o MovieLens Latest Small, o qual possui por volta de 100.000 avaliações e 3.600 *tags* aplicadas por cerca de 600 usuários reais sobre 9742 filmes, onde cada usuário avaliou pelo menos 20 filmes. Esse *dataset* foi atualizado pela última vez em Setembro do ano 2018.

A fim de evitar a perda das informações coletadas devido a falhas físicas, como queda de energia, falha na conexão de internet ou até mesmo falha no servidor de consulta, o *dataframe*, objeto de manipulação de dados utilizado, foi particionado em 10 outros objetos cada um contendo cerca de 1000 filmes. Dessa forma, o experimento foi conduzido de forma incremental, realizando a busca, enriquecimento e persistência dos objetos de forma sequencial. O tempo para realização das consultas SPARQL dos *dataframes* durou entre 1 hora e 30 minutos à cerca de 3 horas por *dataframe*.

Após finalizar o enriquecimento de todos os *dataframes*, esses dados foram combinados novamente em um único *dataframe* e as informações enriquecidas foram unificadas e persistidas em um único arquivo em disco.

A Figura 4 mostra que 17,3%(1685) dos filmes tiveram seus resumos obtidos com sucesso, e que 82,7%(8057) não foram encontrados na DBpedia pela consulta SPARQL. Dos 1685 filmes encontrados pela consulta, todos eles possuíam a propriedade **abstract**, o que indica que os dados presentes na DBpedia são consistentes em termos da propriedade em questão.

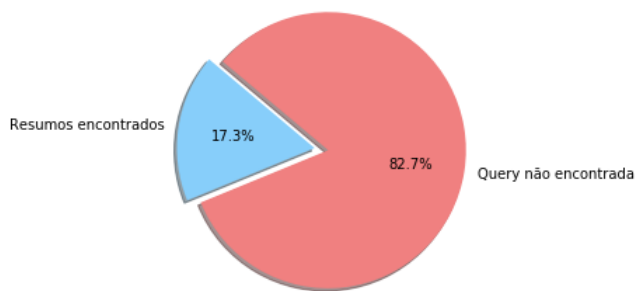


Figura 4. Cobertura obtida com o experimento.

Acreditamos que alguns fatores relacionados a sintaxe dos títulos presente no *dataset* podem ter levado a essa cobertura tão baixa da busca SPARQL. Por exemplo, o filme O Rei Leão possui título original *The Lion King*, e esse título está presente no *dataset* como "**Lion King, The**". Observamos que os filmes que possuam algum artigo no título estão registrados seguindo o padrão de posicionar esse artigo no final do título, após uma vírgula, como os filmes "**Misérables, Les**", "**Bamba, La**", "**Star Is Born, A**", "**Innocent Man, An**". Outro ponto observado ainda na sintaxe dos títulos foi que alguns desses têm a sua tradução para outra língua embutida no título, como os filmes "**And God Created Woman (Et Dieu... crée la femme)**", "**Shanghai Triad (Yao a yao dao waipo qiao)**". Existe ainda a combinação desses dois pontos anteriores, como no filme "**Affair of Love, An (Liaison pornographique, Une)**".

Por fim, observamos ainda que existem filmes que são baseados em obras literárias, e o seu título está registrado na DBpedia como tal obra literária. Sendo assim, ao buscarmos por tal título na ontologia de filmes, a consulta retorna vazia.

VI. CONCLUSÃO

Esse trabalho foi o começo de uma possível vertente de aprimoramento para Sistemas de Recomendação. Ao enriquecermos a base de dados com mais informações sobre os itens a serem recomendados, possibilitamos melhor comparação entre itens, além de podermos inferir mais informações sobre os mesmos. No cenário de filmes, mostramos que a propriedade **abstract** se mostra consistente na DBpedia, então essa é uma informação considerável para utilizarmos no enriquecimento de dados.

Existe espaço para muitas melhorias em trabalhos futuros, dentre elas:

- **Melhorar os resultados da consulta SPARQL**

Acreditamos que pode ser realizado um estudo sobre como tratar melhor as informações presentes nos títulos dos filmes, para assim aumentar a cobertura da consulta SPARQL.

- **Retornar mais propriedades dos filmes**

Nesse trabalho focamos apenas na propriedade **abstract** e mostramos que a mesma é consistente na DBpedia. Porém, existem mais propriedades na DBpedia sobre

filmes que podem ser bastante relevantes, como *director*(diretor), *starring*(elenco), *imdbId*(identificador do imdb), *musicComposer*(compositor musical), que podem ser consideradas no enriquecimento e que devem ser avaliadas também em termos de consistência.

- **Avaliar a acurácia das informações da DBpedia**

É possível também tentar buscar na DBpedia informações que já temos no *dataset*, como os gêneros dos filmes, e comparar tais informações a fim de avaliar quão precisas são tais informações da DBpedia.

- **Aplicar o enriquecimento em diferentes cenários**

Existem diversos outros *datasets* voltados para a área de Sistemas de Recomendação, e também em outros cenários, como voltados para livros, músicas ou até mesmo turismo. Seria interessante aplicar o enriquecimentos nesses outros cenários e avaliar também seu desempenho.

REFERÊNCIAS

- [1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992. [Online]. Available: <http://doi.acm.org/10.1145/138859.138867>
- [2] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997. [Online]. Available: <http://doi.acm.org/10.1145/245108.245121>
- [3] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Springer, 2011. [Online]. Available: <http://www.springerlink.com/content/978-0-387-85819-7>
- [4] G. Shani and A. Gunawardana, "Evaluating recommender systems," Tech. Rep. MSR-TR-2009-159, November 2009. [Online]. Available: <http://research.microsoft.com/apps/pubs/default.aspx?id=115396>
- [5] D. Mladenic, "Text-learning and related intelligent agents: A survey," *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 44–54, Jul. 1999. [Online]. Available: <http://dx.doi.org/10.1109/5254.784084>
- [6] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 73–105. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-85820-3_3
- [7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.
- [8] J. Rosati, P. Ristoski, T. Di Noia, R. d. Leone, and H. Paulheim, "Rdf graph embeddings for content-based recommender systems," in *CEUR workshop proceedings*, vol. 1673. RWTH, 2016, pp. 23–30.
- [9] O. Hassanzadeh and M. P. Consens, "Linked movie data base," in *LDOW*, 2009.
- [10] A. Fridi and S. M. Benslimane, "Towards semantics-aware recommender system: A lod-based approach," *International Journal of Modern Education and Computer Science*, vol. 9, no. 2, p. 55, 2017.
- [11] I. Vagliano, D. Monti, A. Scherp, and M. Morisio, "Content recommendation through semantic annotation of user reviews and linked data," in *Proceedings of the Knowledge Capture Conference*. ACM, 2017, p. 32.
- [12] E. P. Fressato, A. F. da Costa, and M. G. Manzato, "Similarity-based matrix factorization for item cold-start in recommender systems," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2018, pp. 342–347.
- [13] V. A. A. Ayala, M. Przyjaciół-Zablocki, T. Hornung, A. Schätzle, and G. Lausen, "Extending sparql for recommendations," in *Proceedings of Semantic Web Information Management on Semantic Web Information Management*. ACM, 2014, pp. 1–8.
- [14] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2827872>