Universidade Federal da Bahia
Departamento de Ciência da Computação
Doutorado em Ciência da Computação

Enriquecendo o dataset MovieLens com SPARQL

Amanda Chagas de Oliveira
MATE75 - Sistemas Web, Prof. Cássio Prazeres, 2019.1

Agenda

- **Introdução**
- **Conceitos**
  - Sistemas de Recomendação
  - DBpedia
- **Trabalhos relacionados**
- **Enriquecendo o dataset MovieLens com SPARQL**
  - Pré-processamento
  - Implementação da consulta
  - Adição da informação e persistência dos dados
- **Resultados**
  - Avaliação
  - Trabalhos futuros

Introdução

Atualmente, somos expostos a uma enorme quantidade de informações diariamente, e é natural que sintamos dificuldades em encontrar aquela que nos é útil.

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.

Nesse artigo iremos focar no cénario de filmes, realizaremos o enriquecimento do *dataset* MovieLens[1] utilizando a tecnologia SPARQL.

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

Sistemas de Recomendação

Normalmente, as pessoas acabam sempre dependendo do que é recomendado por outros, seja uma recomendação feita por amigos, reviews de livros e filmes, etc. Sistemas de Recomendação propõem automatizar esse processo através de técnicas e ferramentas de softwares que oferecem opções de itens úteis para algum tipo de usuário. [1]

- Filtragem colaborativa
  - A partir do gosto de outros usuários no passado, o sistema, na filtragem colaborativa, recomenda itens ao usuário ativo.
- Filtragem baseado 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. [1]

DBpedia

A DBpedia é o resultado de um esforço colaborativo da extração de informações e da estrutura dessas informações de página disponibilizadas na Wikipedia[2].

A ideia desse projeto surgiu da necessidade de conectar informações que são relacionadas e que já estavam disponibilizadas na Web através do uso de alguns padrões, de acordo com o *Linked Open Data*(LOD).

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

Trabalhos Relacionados

- [2] RDF Graph Embeddings for Content-based Recommender Systems, **2016**
- [3] Towards Semantics-Aware Recommender System: A LOD-Based Approach, **2017**
- [4] Content Recommendation through Semantic Annotation of User Reviews and Linked, **2017**
- [5] Similarity-Based Matrix Factorization for Item Cold-Start in Recommender Systems, **2018**
- [6] Extending SPARQL for Recommendations, **2014**

Enriquecendo o dataset MovieLens com SPARQL - Pré-processamento

O enriquecimento de informação do *dataset* MovieLens será utilizado em um Sistema de Recomendação, em um futuro próximo, na tentativa de melhorar o cálculo de similaridade entre os itens do *dataset*.

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Dataset sem tratamento

	movieId	title	genres	year
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995
1	2	Jumanji	[Adventure, Children, Fantasy]	1995
2	3	Grumpier Old Men	[Comedy, Romance]	1995
3	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995
4	5	Father of the Bride Part II	[Comedy]	1995

Dataset após pré-processamento

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 Python[3] e as bibliotecas Pandas, SPARQLWrapper e JSON dessa linguagem.

```
sparql.setQuery("""

    PREFIX foaf: <http://xmlns.com/foaf/0.1/>
    PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
    PREFIX dbpprop: <http://dbpedia.org/property/>

    SELECT DISTINCT
        ?name
        ?abstract
        ?alternateTitle

    WHERE {
        ?instance a <http://dbpedia.org/ontology/Film>.
        ?instance foaf:name ?name .
        FILTER REGEX (?name, '^""" + search + """$', 'i').
        OPTIONAL {
            ?instance dbpedia-owl:abstract ?abstract .
            FILTER (LANG(?abstract) = 'en').
        }
        OPTIONAL {
        ?instance dbpprop:alternateTitle ?alternateTitle
        }
    }
""")
```

[3] : https://www.python.org/

Adição e persistência das informações

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. Caso a consulta não retorne resultados, é uma adicionada uma mensagem de erro na coluna.

A fim de presistir os dados enriquecidos em disco, utilizamos um método da biblioteca Pandas chamado ***to_csv()***.

Experimento

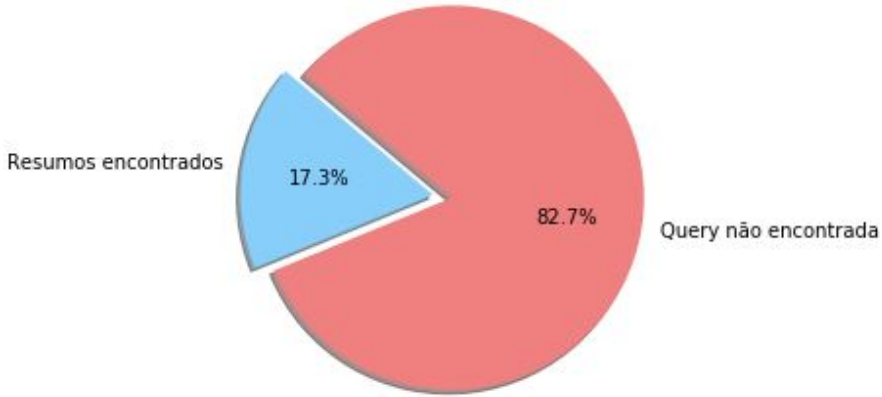
Experimento com o *dataset* MovieLens Latest Small, o qual possui por volta de **100.000 avaliações** e **3.600 tags** aplicadas por cerca de **600 usuários** reais sobre **9742 filmes**, onde cada usuário avaliou pelo menos 20 filmes. Esse *dataset* foi atualizado pela última vez em Setembro do ano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

A fim de evitar a perda das informações coletas devido a falhas físicas, o *dataframe*, objeto de manipulação de dados utilizado, foi particionado em 10 outros objetos cada um contendo cerca de 1000 filmes.

O tempo para realização das consultas SPARQL dos *dataframes* durou entre 1 hora e 30 minutos à cerca de 3 horas por *dataframe*.

Resultados

A Figura ao lado 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. 100% dos filmes encontrados possuem ***abstract***.



<u>O que pode ter levado a essa cobertura?</u>

- “Misérables, Les", “Star Is Born, A"
- “And God Created Woman (Et Dieu... créa la femme)"
- “Affair of Love, An (Liaison pornographique, Une)"

Trabalhos futuros

- Melhorar a consulta SPARQL
  - Estudar um tratamento de dado nos títulos do dataset

- Retornar mais “propriedades” dos filmes
  - Diretor, elenco, “imdbId”, “musicComposer”

- Avaliar acurácia através de comparação com informações já presentes
  - Comparar por exemplo quais gêneros são retornados da consulta

- Aplicar enriquecimento em diferentes cenários
  - Outros *datasets* voltados para livros, músicas ou até mesmo turismo

Obrigada!

Repositório:

- https://github.com/amandachagas/sparqlmovielens

Referências:

- [1] 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
- [2] Rosati, Jessica, et al. "RDF graph embeddings for content-based recommender systems." *CEUR workshop proceedings*. Vol. 1673. RWTH, 2016.
- [3] Fridi, Asmaa, and Sidi Mohamed Benslimane. "Towards Semantics-Aware Recommender System: A LOD-Based Approach." *International Journal of Modern Education and Computer Science* 9.2 (2017): 55.
- [4] Vagliano, Iacopo, et al. "Content recommendation through semantic annotation of user reviews and linked data." *Proceedings of the Knowledge Capture Conference*. ACM, 2017.
- [5] Fressato, Eduardo Pereira, Arthur Fortes da Costa, and Marcelo Garcia Manzato. "Similarity-Based Matrix Factorization for Item Cold-Start in Recommender Systems." *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2018.
- [6] Ayala, Victor Anthony Arrascue, et al. "Extending sparql for recommendations." *Proceedings of Semantic Web Information Management on Semantic Web Information Management*. ACM, 2014.