Utilizando entidades na criação de descritores para avaliar a qualidade de recomendações de novos itens no problema de Cold Start

Andrey Leonardo de Paula[1], **Wladmir Cardoso Brandão**[1], **Rodrigo Richard Gomes**[1]

[1]Pontifícia Universidade Católica de Minas Gerais

andrey.leonardo@gmail.com,{wladmir, richard}@pucminas.br

Abstract. *The e-commerce market has growth in the recent years and every day hundred of new products are placed to sell worldwide and is necessary to find some way to offer these products to the users. To do this work we use Recommender Systems. In some cases the new items cannot be recommended to the users because there is not enough information about the item or the user. This paper will talk about the New Items Cold Start problem and the objective is to use entities to create high quality describers for these items aiming raise the quality and precision of recommendation.*

Resumo. *O mercado de comércio online tem crescido nos últimos anos e diariamente uma enorme quantidade de novos produtos são colocados à venda em todo o mundo e de algum modo esses produtos precisam ser ofertados aos usuários. Para isso são utilizados os Sistemas de Recomendação. Entretanto, em alguns casos os novos itens não conseguem chegar aos usuários pois não há informações suficientes sobre o item ou sobre o usuário. Esse trabalho tratará o problema de Cold Start de novos itens e tem como objetivo utilizar entidades para criar descritores de maior qualidade para os itens, visando aumentar a qualidade e a precisão das recomendações.*

1. Introdução

Hoje a internet não é apenas uma fonte de consultas onde o usuário busca por algo e tem apenas o retorno daquilo que ele procura. Ao procurar algo em um mecanismo de busca, recebem-se centenas de resultados relacionados à consulta realizada e diversas sugestões de itens, produtos e conteúdos, que possam interessar ao usuário, baseados no que está sendo procurado. Ao, por exemplo, realizar uma consulta por ”Brasil”, em um mecanismo de busca qualquer, para descobrir informações sobre o país (por exemplo: qual é a capital, quem é o presidente, quantos habitantes e extensão territorial), podem também aparecer recomendações de hotéis, passagens aéreas, aluguéis de carros e passeios turísticos. Os responsáveis por criar essas recomendações para o usuário são os Sistemas de Recomendação. Eles são responsáveis por recomendar produtos, itens, serviços e conteúdos baseando-se em vários aspectos sobre o usuário. Para citar alguns desses aspectos, eles podem, por exemplo, recomendar baseando-se nas preferências de navegação, no gosto, nas atividades online e nos históricos de buscas do usuário. Os dois tipos mais comuns de recomendação são a baseada em conteúdo e a baseada em filtragem colaborativa [J. Leskovec e Ullman 2014]. Um problema que pode acontecer é o de as recomendações não serem precisas e estarem de acordo com o gosto e as necessidades

do usuário. Assim o usuário pode tratá-las como *spam* e as recomendações acabam se tornando irrelevantes. Quando esse problema ocorre, porque o sistema de recomendação não tem conhecimento sobre o usuário ou sobre os itens ao qual se está recomendando, ele é chamado de problema de *Cold Start*. Este problema se divide em três categorias: nova comunidade, novo usuário e novo item, sendo que esta última categoria é o objeto de estudo neste trabalho.

O problema de *cold start* de um novo item é visto como um estudo de grande importância porque hoje plataformas online publicam centenas de novos itens todos os dias e é necessária uma recomendação efetiva para esses itens, pois isso ajuda a manter o interesse dos usuários. Atualmente a abordagem mais utilizada para recomendação de itens é a filtragem colaborativa, mas nessa abordagem um item precisa ser avaliado pelo maior número possível de usuários para que a recomendação tenha uma precisão satisfatória e seja de fato efetiva [Saveski e Mantrach 2014]. Algumas outras abordagens são propostas para esse problema como o uso de identificação de classificadores representativos [Choi e Han 2013], análise semântica de dados textuais externos [C. Krauss e Arbanowski 2014] e *Local Collective Embeddings (tradução livre: incorporação coletiva local)* [Saveski e Mantrach 2014].

O objetivo desse trabalho é investigar se há melhora na qualidade das recomendações com a inserção de descritores aos novos itens. Foram extraídos descritores para os novos itens a partir de entidades e então foi avaliada a precisão e a qualidade das recomendações, comparando os resultados obtidos antes e após a inserção dos descritores extraídos das entidades. As entidades são páginas na Web que possuem informações adicionais sobre os itens ao qual se está sendo feita a consulta e é de onde serão extraídos os descritores. Descritores são características extraídas de um conjunto de informações que melhor descrevem um item.

Este estudo propõe uma abordagem que utiliza entidades acessadas após consultas feitas em um mecanismo de buscas ou acessadas através de APIs, para criar ou adicionar descritores complementares a novos itens para melhorar o significado destes itens. API é a abreviação para *Application Programming Interface* ou Interface de Programação de Aplicações no português, que representam funções ou métodos de aplicações que são disponibilizados para que outros desenvolvedores utilizem de forma transparente, sem ter acesso direto ao código fonte. Utilizando a recomendação baseada em conteúdo para fazer recomendações de novos itens aos usuários, basea-se nos novos descritores gerados para avaliar o aumento da qualidade das recomendações de novos itens. Foram gerados dados baseados em avaliações feitas antes e após a adição dos novos descritores e esses dados foram analisados para dar uma conclusão sobre os resultados obtidos na recomendações de itens em *cold start* utilizando essa abordagem.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta o referencial teórico usado neste trabalho, a Seção 3 descreve a metodologia aplicada, a Sessão 4 descreve os resultados e a Sessão 5 descreve a conclusão e trabalhos futuros.

2. Referencial Teórico

C. Anderson (2006) em *The Long Tail* diz: ”estamos deixando a era da informação e entrando na era da recomendação”.

Com o crescimento constante da Internet o usuário passou a ter um papel diferente no

seu uso, deixando de ser apenas um consumidor e tornando-se um gerador de informação. Informação que é gerada em uma escala que não pode ser consumida totalmente pelos usuários. Porém várias dessas informações estão dispersas e são perdidas por uma série de fatores: falta de organização, falta de identificação adequada, descrições que não representam totalmente o conteúdo, dentre outros fatores. Existe uma imensa quantidade de informações úteis e inúteis misturadas e é necessário separá-las para um melhor aproveitamento delas. Visando essa necessidade surgiram os Sistemas de Recomendação.

2.1. Sistema de Recomendação

Um Sistema de Recomendação (SR) consiste em um sistema que utiliza diversas tecnologias e pode ser classificado basicamente em dois grupos: Recomendação Baseada em Conteúdo e Recomendação por Filtragem Colaborativa [J. Leskovec e Ullman 2014]. Na recomendação baseada em conteúdo, recomendam-se itens baseado no perfil de compras, de navegação, de busca e também em avaliações de itens feitas pelo o usuário no passado [Saveski e Mantrach 2014]. Essa abordagem pode ser vista, por exemplo, na plataforma *Netflix*.Na recomendação por filtragem colaborativa recomendam-se itens baseado no que outros usuários com gostos similares compraram ou avaliaram [Saveski e Mantrach 2014]. Esse tipo de abordagem pode ser vista principalmente em lojas virtuais como, por exemplo, o *Amazon*.

A origem dos SRs pode ser rastreada nos trabalhos extensivos de ciência cognitiva, teoria da aproximação, recuperação de informação, teoria de previsão e em modelos de escolhas do consumidor em marketing [M. Albanese 2013]. Algumas dessas áreas estão diretamente relacionadas ao comportamento humano. Visto que a principal funcionalidade de um SR é reconhecer padrões e hábitos do usuário e tentar prever o que o ele quer, fica clara a relação entre essas áreas. Na Web, há uma quantidade enorme de serviços que fazem uso de aplicações que necessitam predizer as necessidades e escolhas do usuário e os SRs são utilizados para recomendar produtos, serviços e informações para prover uma experiência personalizada aos usuários. Para prover essa experiência personalizada e a melhoria na qualidade da recomendação, exige-se cada vez mais informações pessoais dos usuários. [Resnick e Varian 1997] levantam um problema relacionado à privacidade em seu trabalho. Tal fato nos leva a questão: será que as pessoas estão dispostas a dar mais informações sobre elas? Se estiverem, como esses dados serão armazenados de maneira que possamos garantir a integridade e confidencialidade dos mesmos? Isto pode ser observado como um problema de segurança de informação e é algo a ser considerado mas não será tratado neste trabalho.

2.2. Trabalhos Relacionados

Existem abordagens que visam melhorar a qualidade das recomendações na presença do problema de *cold start* como a abordagem híbrida proposta por [B. Scholz 2012] usando a Filtragem Colaborativa e a Recomendação Baseada em Conteúdo [Saveski e Mantrach 2014]. Essa abordagem leva em consideração a ideia de que pessoas com um perfil específico terão interesse sempre nos mesmos tipos de itens. Com essa abordagem foram obtidos bons resultados que resolveram parcialmente o problema de *cold start* de novos usuários. Esse método só foi efetivo em uma base de dados menor que 10 mil usuários. Ao expandir para uma base de dados próxima a 1 milhão de usuários o método não foi efetivo como esperado. Apesar dos resultados serem baseados em novos usuários, o método pode ser aplicado para novos itens. Outra abordagem

apresentada por [C. Krauss e Arbanowski 2014] é a análise de opiniões e sentimentos dos usuários em redes sociais. Nesta abordagem são extraídas palavras chaves que subsequentemente são mapeadas com as propriedades dos itens e então calcula-se um valor de associação entre os usuários e o item para fazer a recomendação. A abordagem principal é fazer uma análise automática da semântica na qual, independentemente da complexibilidade da sentença, elas são tratadas como sentenças estruturadas de palavras. Através disso é possível utilizar os dados extraídos das redes sociais para recomendar itens para o usuário assim que ele se conecta pela primeira vez em um serviço, desde que ele utilize uma contas de rede social. Nesse estudo a qualidade da recomendação depende da atividade do usuário na rede social a qual ele participa, quando mais curtidas, postagens e comentários, maior será a qualidade das recomendações feitas à ele.

O foco deste trabalho é tratar do problema de *cold start* para novos itens usando uma abordagem que consiste em alterar ou inserir descritores em novos itens utilizando entidades para enriquecer as informações. Utilizando a Recomendação Baseada em Conteúdo [Saveski e Mantrach 2014], os itens com descritores de maior qualidade poderão garantir uma precisão maior na recomendação dos novos itens.

3. Metodologia

3.1. Implementação

Para esse estudo foi implementado um SR utilizando a linguagem de programação *Ruby* (Versão 2.2.2) [Flanagan e Matsumoto 2008] que é uma linguagem baseada em script e a biblioteca *Predictor* (Versão 2.0) [Pathgather 2015]. A biblioteca armazenar em uma matriz a quantidade de vezes que a relação entre dois itens ocorre, por exemplo,**usuário x item**. Após armazenar a quantidade de vezes que as relações ocorrem calcula-se a distância entre eles através do Coeficiente de Similaridade de Jaccard para encontrar similaridades entre os itens, por exemplo, quais quais filmes possuem os mesmos gêneros. O coeficiente de similaridade de Jaccard, também conhecido como índice de Jaccard, é um modelo estatístico criado por Paul Jaccard (1868-1944) e publicado em 1901, que é usado para comparar similaridades e diversidade em grupos amostrais. Para isso é necessário determinar quais pares serão armazenados na matriz e o peso de ativação de cada relacionamento. O peso é utilizado para definir o nível de importância de cada item no cálculo da similaridade. Quanto maior o peso maior será a importância daquele item na matriz. Os pares definidos nesse trabalho foram **usuários:filme**, **avaliações:filme**, **gêneros:filme**, **classificações:filme** e **descritores:filme**. A biblioteca armazena as similaridades no *Redis* [Redislabs 2016b] que é um servidor de estrutura de dados em memória RAM, o que resulta em melhor desempenho ao realizar os cálculos e as buscas das similaridades.

Foi utilizada uma base de dados de filmes do *GroupLens* [GroupLens 2016] com 10.329 filmes, 105.339 avaliações feitas por 668 usuários. A versão utilizada é *"Small: 100,000 ratings and 6,100 tag applications applied to 10,000 movies by 700 users. Last updated 1/2016"*. As bases de dados do GroupLens são utilizadas em diversos trabalhos relacionados a sistemas de recomendação e *cold start* como pode ser visto nos trabalhos de [M. Albanese 2013, B. Scholz 2012, Choi e Han 2013, Xuan Nhat Lam 2008]. Para este estudo foram ignoradas as *tags* presentes na base. A base de dados do GroupLens não possui dados sobre o perfil do usário e esse perfil foi criado artificialmente pela ferramenta. Foram utilizadas três bases de conhecimentos para acessar informações sobre os

filmes e extrair os descritores: *MoviesApi.com*, *OMDBApi.com* e *TheMovieDB.org*. Todas foram acessadas diretamente por APIs públicas que retornam as informações dos filmes que estão disponíveis também nos respectivos Websites. Essas entidades externas proveram informações extras sobre os filmes de onde foram extraídos os descritores utilizados nesse estudo. Vale citar que a API do MoviesApi é uma API não oficial do IMDB e todas as informações extraídas dessa API são as mesmas informações públicas encontradas nas páginas dos filmes do IMDB. A base de dados do GroupLens possui uma tabela com identificadores dos filmes no IMDB e todas essas APIs recebem esses IDs como entrada para busca de informações sobre os filmes e foram esses identificadores que foram utilizados para buscar as informações sobre os filmes de onde foram extraídos os descritores.

Foram analisados dois cenários com os seguintes pesos de ativação:

Cenário 1:

1. Usuários: peso 3
2. Avaliações: peso 2
3. Gêneros: peso 1
4. Classificação etária: peso 1
5. Descritores: peso 1

Cenário 2:

1. Usuários: peso 1
2. Avaliações: peso 1
3. Gêneros: peso 1
4. Classificação etária: peso 1
5. Descritores: peso 1

Os pesos do **Cenário 1** foram definidos baseando-se na importância dos relacionamentos no contexto da recomendação. Foram testadas algumas variações nos pesos, como por exemplo dando mais peso para os gêneros, porém a recomendação se tornou tendenciosa para filmes que continham apenas os mesmos gêneros e tornou os resultados das recomendações imprecisos em várias ocasiões. Essa distribuição de peso adotada levou a recomendações mais balanceadas, pois basea-se principalmente no perfil de filmes que o usuário tem relação e em filmes que possuem avaliações parecidas com as que o usuário submeteu. Ao mesmo tempo dá ênfase aos relacionamentos com pesos menores. O **Cenário 2** é um cenário neutro onde todos os relacionamentos possuem a mesma importância, foi avaliado esse cenário para analisar o impacto dos pesos dos relacionamentos no problema de *cold start* e o quanto pode influenciar os resultados.

Para extrair os descritores foi escrito um script em Ruby que processa as informações coletadas das entidades externas e extrai dessas informações todos os descritores que possam dar um melhor sentido ao item excluindo as *stopwords*. *Stopwords* são as palavras mais comuns em um texto e que aparecem repetidas vezes. Elas são filtradas para não serem processadas durante a extração dos descritores. Foi utilizada uma lista do *Ranks NL* [Doyle 2015] que possui 665 termos em inglês considerados *stopwords*. Após a extração foi feita a classificação dos descritores de acordo com a frequência do termo (TF - *term frequency*) onde considera-se a quantidade de vezes que um descritor apareceu em cada entidade individualmente. Nessa classificação os descritores que mais aparecem ficam nas primeiras posições. Após feita a classificação por TF é necessário combinar os descritores de cada entidade e gerar uma classificação única de onde serão extraídos os 10 melhores descritores. Para realizar a combinação é considerada a posição em que cada descritor aparece na classificação por TF em sua respectiva entidade e segue-se a lógica exibida na Figura 1:

1. Soma-se as posições em que cada descritor aparece nas classificações.

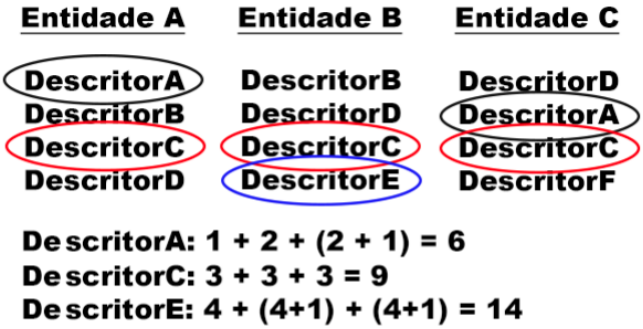

Figura 1. Método de combinação dos descritores

2. Se o descritor não aparece em uma classificação é somado o valor da maior posição em que ele apareceu mais 1.

Dessa maneira os descritores com menor valor após a combinação são os descritores que possuem maior relevância e serão os primeiros na nova classificação combinada.

Após a combinação dos descritores foram associados aos filmes os 10 melhores descritores. Há uma tabela no banco de dados onde são salvos os descritores e feita uma relação NxN através de uma tabela de ligação. Quando filmes possuem o mesmo descritor, adiciona-se o descritor uma única vez a tabela e esse descritor é relacionado a todos os filmes a qual ele pertence. Com todos os descritores criados eles são adicionados à matriz de similaridade e onde é feito o cálculo baseando-se no peso de ativação definido para o relacionamento dos descritores.

A implementação do SR foi realizada inicialmente utilizando uma configuração padrão da biblioteca o que ocasionou em um longo tempo para processamento para a geração da matriz. Foram necessárias mais de 30 horas para gerar as similaridades com os relacionamentos de cada cenário. Após alguns ajustes indicados na documentação da biblioteca e melhorias no algoritmo, esse tempo foi reduzido para pouco mais de 2 horas. Essa melhoria no desempenho se deu também à utilização do *Hiredis* [Redislabs 2016a], como indicado na documentação do Predictor, que é uma biblioteca em **C** para o Redis que ajuda linguagens de alto nível, como Ruby, a enviar e receber comandos ao Redis de maneira mais rápida e eficiente, pois, desacopla os comandos enviados e recebidos da camada de entrada e saída. Outra alteração realizada foi a utilização de *Lua script* [PUC-Rio 2015] ao invés do script padrão em Ruby da biblioteca para realizar o processamento dos dados na geração da matriz. Lua é uma linguagem de script para programação procedural desenvolvida no Brasil pela PUC-Rio [R. Ierusalimschy e Celes 2007].

3.2. Métricas

Para a aplicação da metodologia proposta foram selecionados um total de 63 usuários aleatoriamente pelo algoritmo criado para essa finalidade. O algoritmo utiliza o método *sample(:int)* do Ruby que retorna uma quantidade de itens de um array **(arr1)** aleatoriamente de acordo com o valor passado por parâmetro criando um novo array **(arr2)** com os valores retornados. Foi realizada uma iteração no **arr2** e foi escolhida uma avaliação feita por cada usuário retirando de um grupo chamado de 'Melhores Avaliações', que consiste nas melhores avaliações dadas pelo usuário aos filmes. As notas das avaliações vão de

0 a 5 com intervalos de 0.5 pontos. Para selecionar a melhor avaliação de cada usuário foi levado em conta, por exemplo, se o usuário possuir 10 avaliações com nota 5 apenas uma das 10 avaliações de 5 pontos será removida de maneira aleatória. Para definir quais eram as melhores avaliações realizadas pelo usuário foi calculada uma média das notas dadas aos filmes pelo usuário e somente as avaliações que estavam acima dessa média foram utilizadas. Do filme selecionado foram removidas todas as avaliações recebidas e todos os seus descritores da base de dados e da matriz para criar um caso de *cold start* do item sem descritores. Em um arquivo de texto foram salvos os IDs dos usuários que foram selecionados junto com os IDs dos respectivos filmes que foram removidos de cada usuário e desses dados foram feitas as análises iniciais.

Foi verificado se o filme removido está sendo recomendado ao usuário ao qual o filme estava vinculado e qual a posição inicial desse filme na lista de recomendações na condição de *cold start*. A posição inicial do filme foi registrada em um arquivo de resultados e em seguida foram gerados e adicionados descritores ao filme e à matriz. Com os descritores gerados e adicionados à matriz foram verificadas novamente as recomendações para cada usuário e registradas as posições em que o filme relacionado a cada um deles apareceu com os descritores. Essa posição foi registrada no arquivo de resultados para posteriormente serem realizadas as análises.

Para a análise dos resultados foram utlizados percentis para normalizar o tamanho das listas de recomendação de cada usuário para 100. Os itens terão posição relativa nesta lista de recomendação. As posições que os itens aparecem no arquivo de resultados gerado nos testes são as posições absolutas.

4. Resultados

Os resultados estão demonstrados na Figura 2 e Figura 3. A Figura 2 representa os resultados do Cenário 1 e a Figura 3 representa os resultados do Cenário 2. Nos gráficos são demonstradas as mudanças de posições relativas obtidas pelos filmes após inserir os descritores. O eixo Y são as mudanças nas posições relativas do item em uma escala de 0 a 100 e o eixo X são a quantidade de itens. Os experimentos realizados demonstraram que essa abordagem foi capaz de gerar mundanças nas posições na recomendação de novos itens no problema de *cold start*.

No Cenário 1 pode-se observar que mesmo o relacionamento com os usuários e com as avaliações possuindo um peso maior, há mundanças consideráveis nas posições dos itens recomendados. Em alguns casos o Cenário 1 apresentou resultados superiores aos resultados do Cenário 2, cenário que todos os relacionamentos possuíam o mesmo peso. Porém no Cenário 2 é possível notar uma dispersão maior das mudanças e o número médio fica em 25,33 posições alteradas por item enquanto no Cenário 1 as posições alteradas ficaram mais agrupadas com uma mudança média de 23,73 posições por item. Analisando os resultados através da mediana no Cenário 2 o resultado obtido nota-se uma mudança de 25 posições por item enquanto no Cenário 1 a mundança obtida são de 22 posições por item.

A amostra total são de 63 usuários mas no gráfico do Cenário 1 estão representados 57 usuários e no gráfico do Cenário 2 estão representados 56 usuários. Isso ocorre devido a um caso especial que merece ser citado e estes resultados estão representados na Tabela 1 e Tabela 2. Deste caso especial são 6 usuários do Cenário 1 representados na Tabela 1 e 7

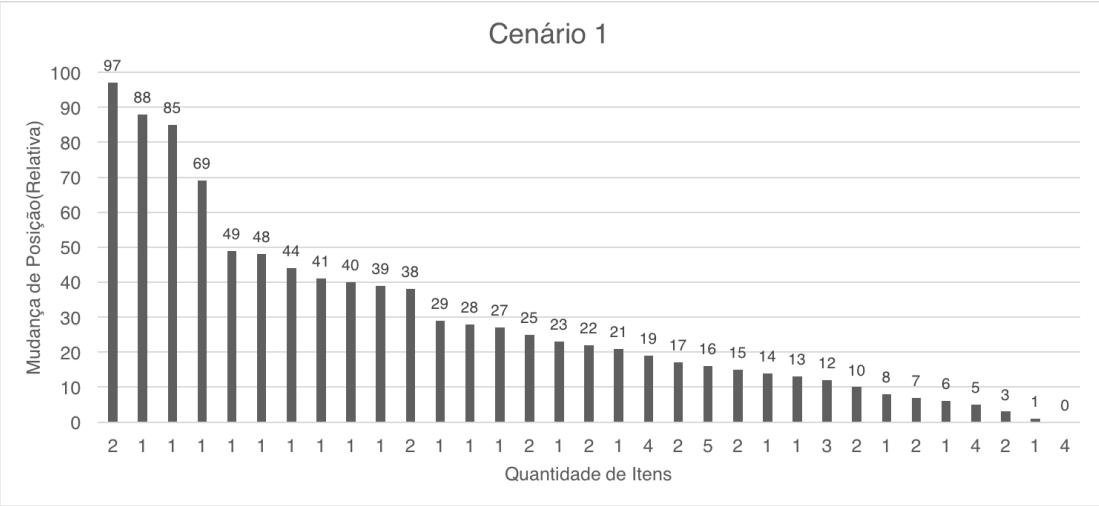

Figura 2. Resultados do Cenário 1

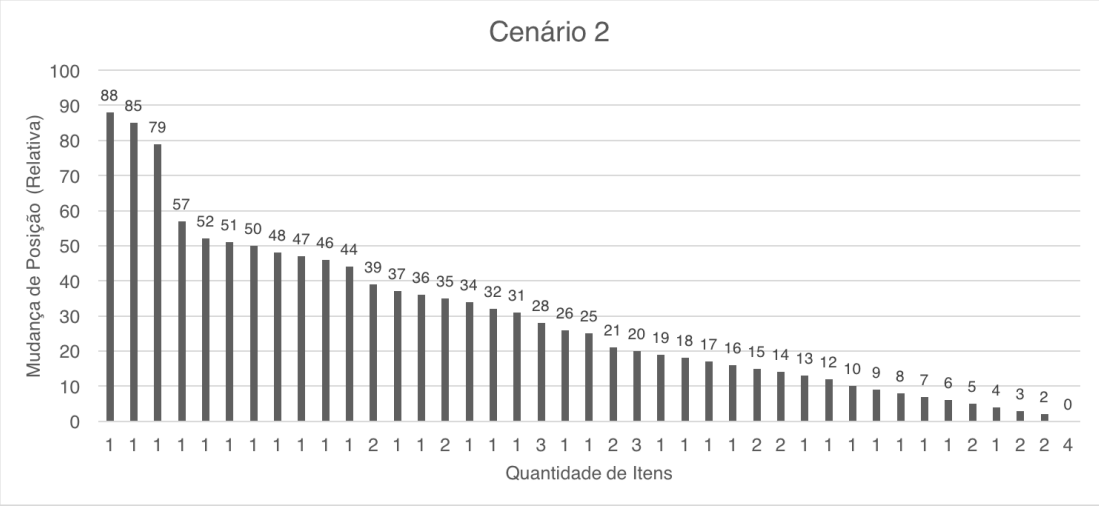

Figura 3. Resultados do Cenário 2

usuários representados do Cenário 2 representados na Tabela 2. Alguns itens ao passarem para o estado de *cold start* não foram recomendados antes da inserção dos descritores e só passaram a ser recomendados após a inserção dos descritores. Não foi possível utilizar percentis para avaliar a mudança de posições desses itens pois não havia uma referência da posição anterior à inserção dos descritores. Para esses itens foi considerada a posição absoluta para avaliar a mudança da posição da recomendação após a inserção dos descritores. Esses resultados podem ser observados na Tabela 1 e Tabela 2. É possível notar que os itens nas três últimas posições obtiveram resultados parecidos em ambos cenários, com itens que não eram recomendados em estado de *cold start* e que passaram a ser recomendados mas com uma mudança menor nas posições em relação ao outros itens na mesma situação. No Cenário 1 houve uma mudança média de 3049,16 posições por itens enquanto no Cenário 2 a mudança média foi mais que o dobro, atingindo 6730,57 posições por item.

Tabela 1. Cenário 1 - Itens não recomendados antes de adicionar descritores (Posição absoluta)

FilmeID	Tam. da Lista (A)	Pos. Antes	Pos. Depois (B)	Mud. de Pos. (A-B)
162	3324	N/A	2021	1303
1193	1384	N/A	308	1076
4995	1270	N/A	614	656
79132	1298	N/A	1285	13
1246	2270	N/A	2263	7
1201	1708	N/A	1708	0

Tabela 2. Cenário 2 - Itens não recomendados antes de adicionar descritores (Posição absoluta)

FilmeID	Tam. da Lista (A)	Pos. Antes	Pos. Depois (B)	Mud. de Pos. (A-B)
923	4192	N/A	2219	1973
162	4350	N/A	2577	1773
1193	1751	N/A	95	1656
253	1455	N/A	128	1327
79132	1530	N/A	1519	11
1201	1947	N/A	1947	0
1246	2328	N/A	2328	0

5. Conclusão e Trabalhos Futuros

O objetivo desse trabalho foi o de investigar o potencial de melhoria na qualidade das recomendações com a inserção de descritores extraídos de entidades em novos itens. A abordagem proposta se baseia na ideia de que novos itens possuirão entidades externas de onde serão extraídos os descritores para que ele seja eficaz. Os testes aplicados mostraram resultados positivos para serem utilizados como uma alternativa para resolver o problema de *cold start* de novos itens. Um problema nesta abordagem está relacionado à adição de novos itens à matriz de similaridade, pois conforme a base de dados cresce, o tempo para processar a similaridade de um novo item adicionado cresce exponencialmente. Existem estratégias que podem ser adotadas para contornar esse problema, como processar as similaridades dos novos itens somente uma vez por dia em horários de pouco acesso à aplicação. Os trabalhos restantes que deverão ser realizados serão o de analisar se adicionar mais descritores leva a melhores recomendações, fazer um estudo comparativo entre essa abordagem e uma abordagem estado da arte de *cold start* e por fim analisar uma abordagem híbrida da recomendação baseada em conteúdo e filtragem colaborativa se essa abordagem pode levar a melhores resultados.

Referências

B. Scholz, e. a. (2012). A recommendation system based on a subset of raters. *ICUIMC '12: Proceedings of the 6th International Conference on Ubiquitous Information*.

C. Krauss, S. B. e Arbanowski, S. (2014). Preference ontologies based on social media for compensating the cold start problem. *SNAKDD-2014*.

Choi, S.-M. e Han, Y.-S. (2013). Identifying representative ratings for a new item in recommendation system. *ICUIMC-2013*.

Doyle, D. (2015). Ranks NL Webmaster Tools. Disponível em: http://www.ranks.nl/stopwords Acesso em 16 mai. 2016.

Flanagan, D. e Matsumoto, Y. (2008). *The Ruby Programming Language*. O'Reilly Media.

GroupLens (2016). MovieLens. Disponível em: http://grouplens.org/datasets/movielens/ Acessado em 11 jan. 2016.

J. Leskovec, A. R. e Ullman, J. D. (2014). *Mining of Massive Datasets*. digital book.

M. Albanese, e. a. (2013). A multimedia recommender system. *ACM Transactions on Internet Technology*, 13(1).

Pathgather (2015). Predictor. Disponível em: https://github.com/Pathgather/predictor Acessado em 10 mar. 2016.

PUC-Rio (2015). Lua: Documentation. Disponível em: https://www.lua.org/docs.html Acessado em 16 mai. 2016.

R. Ierusalimschy, L. H. d. F. e Celes, W. (Jun.2007). The evolution of lua. *HOPL III: Proceedings of the third ACM SIGPLAN conference on History of programming languages*, pages 56–58.

Redislabs (2016a). HIREDIS. Disponível em: https://github.com/redis/hiredis Acessado em 16 mai. 2016.

Redislabs (2016b). Redis. Disponível em: http://redis.io Acessado em 10 mar. 2016.

Resnick, P. e Varian, H. R. (Mar.1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Saveski, M. e Mantrach, A. (2014). Item cold-start recommendations: Learning local collective embeddings. *RecSys '14: Proceedings of the 8th ACM Conference on Recommender systems*, pages 89–96.

Xuan Nhat Lam, e. a. (Jan.2008). Addressing cold-start problem in recommendation systems. *ICUIMC '08: Proceedings of the 2nd international conference on Ubiquitous information*, pages 208–211.