Resumo:

Por facilitar a procura por informação no mar de dados da Internet, Sistemas de Recomendação (SR) são extremamente populares na Web. Usualmente SR tentam prever avaliações de usuários sobre um item desconhecido, para gerar recomendações personalizadas. Nesses sistemas, em especial os baseados em conteúdo, características de itens são processadas para identificar outros itens relacionados, mas é comum que sejam negligenciadas relações semânticas entre eles. Focar unicamente em dados sintáticos, favorece o efeito bolha, o que é caracterizado por usuários sendo menos expostos a itens relevantes e inesperados, algo desejável num SR. Encontrar itens com similaridades semânticas pode minimizar esse efeito, já que provê uma ainda relevante, porém mais abrangente similaridade. Nesse sentido, este trabalho propõe um Sistema de Recomendação, baseado em conteúdo, com Similaridade Semântica Ponderada por Links de Recursos (RLWS) na \textit{DBPedia}. O objetivo é verificar que resultados são obtidos pela comparação de termos da sinopse dos filmes, onde RLWS analisa relações semânticas diretas e indiretas entre eles, usando o DBPedia. Sendo assim, foi conduzido um experimento comparando RLWS com a conhecida similaridade do cosseno. Considerando um conjunto de cinco itens ($k=5$), o sistema proposto melhorou a precisão média (MAP) em 51\%, quando privilegiado relacionamentos indiretos, e 27\% para os diretos. Além disso, a proposta também melhorou o desempenho da métrica MRR em 26\% privilegiando relacionamentos indiretos, e de 11\% para diretos.

Abstract:

With the goal to facilitate the efforts when searching information on the Web, Recommender Systems (RS) have become extremely popular in recent years on the Web. Usually, RS try to predict the user’s evaluation over an unknown item to generate personalized recommendations. Those systems, especially those content-based, process syntactic data (e.g., item features) to identify new related items, but often neglect the semantic similarities between them. Focusing only on syntactic data favors the “bubble filter effect” - an effect characterized by the user not being exposed to unexpected and relevant items, a desired feature for RS. Finding items with semantic similarities minimizes the “bubble filter effect” since it can provide a broader and more relevant similarity. In this sense, this work proposes a Recommender System (content-based) with a Resource Link-Weighted Similarity (RLWS), using \textit{DBPedia}. The proposed system verifies which results are obtainable by comparing terms from film synopses, and then RLWS analyses the direct and indirect semantic relations between them, using the DBPedia. We conduct an experimental evaluation comparing the RLWS with the well-known cosine similarity. Considering a result set of five items ($k=5$), the proposed system improves the MAP performance by 51\% when weighting more indirect relationships between terms, and for the direct relationships by 27\%. In addition, the proposal improves the MRR performance in 26\% weighting more indirect relationships, and 11\% using the direct ones.

Introdução:

A expansão dos meios de comunicação através da Internet possibilitou o rápido acesso a todo tipo de informação de diversas áreas do mundo a todo lugar. Consumir conteúdo digital tornou-se atividade comum no dia das pessoas. Conforme mais se expande o acesso as mídias digitais mais conteúdo é gerado e mais está disponível para ler, ver, ouvir e interagir. Segundo Saint John Walker (2014) chegamos a uma era em que trafegamos uma quantidade enorme de dados que rapidamente perde-se a escala e cognição para o humano. Qual o significado de 400 milhões de tweets por dia? Usar o pensamento empírico de grandes matemáticos como “medir é conhecer” (William Thomson) torna-se especialmente difícil com o volume de informações produzidas neste século. Com a quantidade dados disponíveis não é irônico ouvir “não sei qual livro ler”, pois apesar do fácil acesso existe uma grande sobrecarga a qual expõe o usuário a um mar de dados (David Wellman, 2013), dificultando o acesso ao conteúdo que seja mais relevante.

O volume de informações apresenta-se como um obstáculo ao usuário que deseja consumir algum tipo conteúdo. Compras online possuem milhares de opções e nem todos estão dispostos a passar um grande tempo olhando o catálogo disponível, ainda mais que uma das razões pela preferência de compra pela Internet é justamente a “falta de tempo”, conforme revela análise de Mykolas. Dessa forma é natural que o usuário recorra a alternativas para se guiar pelas informações e encontrar mais facilmente aquilo que lhe é mais útil. Para minimizar o obstáculo que o volume de informações se opõem, é comum que se apele da ajuda de conhecidos, amigos, como apontado pela análise de Mykolas, onde um dos fatores relacionados ao consumidor que influenciam a opção pela compra pela Internet são as recomendações de outros usuários.

Também a larga difusão da Internet, principalmente pela Web, cria um desafio da busca de informação. Sistemas populares de recuperação de informação, como Google, amenizam o problema (Egyptian), mas são carentes quanto personalização e priorização da informação em relação as preferências e interesses do usuário. Essa é uma das razões pelo grande aumento do desenvolvimento e procura por sistemas de recomendação. Sistemas de recomendação são sistemas de filtragem de informação que possuem objetivos de prever a avaliação e preferência (Ricci2011). Tais soluções contribuem ainda mais com a experiência do usuário no que diz ao conceito do “need for cognition” (NFC) que reflete na tendência de indivíduos em se engajar e aproveitar numa atividade (Mykolas:2015a). Esses sistemas filtram os dados para reduzir o problema da sobrecarga de informação (Konstan2012), podendo ser utilizados em diversas domínios como livros, filmes, músicas até para construir experiências em jogos online (Activision:Glixel).

Os sistemas de recomendação tipicamente possuem três tipos de abordagens para as sugestões: filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida que leva em consideração as duas anteriores. Filtragem baseada em conteúdo são fundamentadas na descrição dos dados e nas preferências dos usuários \citep{Aggarwal2016}. Desse modo, o objetivo deste trabalho é modelar um sistema de recomendação com métricas que realizem a filtragem baseada em conteúdo de itens, além de incorporar dados de serviços da web semântica, focado no uso para o domínio de livros.

Motivação:

Com a crescente popularização do acesso e uso da Web no mundo, cada vez é mais comum que pessoas escolham este ambiente para fazer compras, o comércio eletrônico. No Brasil, em 2015, movimentou R\$ 41,3 bilhões com o e-commerce\footnote{Modalidade de comércio que realiza suas transações financeiras por meio de dispositivos e plataformas eletrônicas (https://ecommercenews.com.br/o-que-e-e-commerce/)} segundo estudos da E-bit\footnote{https://www.ebit.com.br/} como aponta o \cite{Sebrae:2016}. O estudo também levanta que livros e revistas estão em 5º lugar como o tipo de item mais procurado. O crescimento do uso de dispositivos eletrônicos para realizar compras online, mostra que cada vez mais pessoas utilizam a internet, especialmente para as redes sociais. Somente o Facebook\footnote{https://www.facebook.com} já registrou em 2017 2 bilhões de usuários ativos \citep{Statista:2017}. O tamanho da plataforma mostra que existe uma quantidade enorme de dados sobre usuários da Internet de todo o mundo, podendo ser fácil de encontrar relações de amizade, assim como preferências de livros. Esses dados servem como uma excelente fonte de busca para montar um perfil.

A grande quantidade de informação sobre os usuários presentes nessas redes sociais, é de amplo valor para construção de sistemas de recomendação. Em muitas dessas plataformas, é disponibilizado para terceiros uma API para que por exemplo o usuário possa acessar em uma aplicação utilizando as credenciais dessa rede, o que pode facilitar a adesão de novos serviços. Assim, é possível construir um sistema de recomendação baseado em conteúdo já com uma infraestrutura de dados conhecida e amplamente difundida e aceita pelos usuários. A utilização do SR com filtragem baseada em conteúdo, aprende e recomenda itens que sejam similares aos que o usuário já demonstrou interesse \citep{Ricci2011}.

Na similaridade em termos associados aos itens em comparação, é comum no domínio de livros seja comparado termos como gênero e autor. Nesse caso, é analisando se já foi demonstrado interesse em livros com esses termos, para que assim o sistema aprenda e recomende novos livros com esses mesmos termos. Entretanto, pode ser interessante para o usuário encontrar livros que não sejam necessariamente do mesmo gênero ou similar, mas que possuam narrativas mais similares. Nesse sentido, analisar a similaridade de conteúdo da descrição de um livro que contenha um trecho da sua narrativa, possa levar ao usuário a sair do seu círculo tradicional de preferência, podendo aumentar o seu nível de NFC. Uma das propostas desse trabalho é explorar os resultados analisando esse termo.

Além de analisar a similaridade de dois livros também observando para a descrição da narrativa, será utilizado o serviço da web semântica DBPedia\footnote{ http://wiki.dbpedia.org}, para obter mais informações das descrições dos livros extraindo relações semânticas de entidades presentes nos textos. Para o SR prover as informações personalizadas para usuário é necessário criar um perfil do usuário para indicar o tipo de conteúdo, baseando-se em itens que sejam similares que aos que usuário gostou no passado. Expandindo o alcance do SR será proposto e avaliado um modelo de usuário para o domínio de livros que irá além da similaridade baseada no gênero e autor, mas que também leve em consideração a relação semântica das entidades presentes nas descrições das narrativas dos livros.

Problema:

No início do desenvolvimento de SR um problema muito comum ao montar o perfil do usuário, é a falta de informação sobre suas preferências, também conhecido como o problema da partida a frio. O sistema ainda não obteve interações suficientes para montar um perfil, afetando diretamente a qualidade das recomendações. Com o serviço do Facebook \footnote{https://www.facebook.com} existe a possibilidade de extrair dados das preferências de forma automática e transparente, uma vez que já é amplamente aceito pelos usuários. Dessa forma, além de facilitar a montagem do perfil do usuário, de imediato diminui a sobrecarga de informação que possivelmente ainda passaria para poder usufruir de um SR.

Outra questão trata-se de como esses algoritmos de filtragem e personalização afetam as pessoas. O livro “The Filter Bubble” \cite{Pariser:2011} levanta preocupações sobre tais sistemas, onde o usuário fica fortemente sujeito a apenas ao mesmo tipo de conteúdo, ou informação que não venha criar conflitos de ponto de visão, o efeito bolha. Assim, utilizando um SR que apenas analisasse termos de gênero e título poderia deixar o usuário “preso” no círculo tradicional de preferência. Essa preocupação pode também ter um impacto negativo no sistema, já que é possível que os usuários venham a encontrar outros conteúdos que poderiam ter interesse, mas são apenas encorajados a aqueles mais tradicionais.

Propondo um sistema em que também seja possível extrair similaridade da descrição das narrativas dos livros, analisando e buscando outras relações semânticas com as entidades presentes, pode trazer resultados que amenizem esse efeito. Esse trabalho tem um dos objetivos de explorar que resultados podem ser obtidos levando em consideração essa abordagem.

Objetivos da solução da proposta:

Este trabalho propõem a criação de um SR baseado em conteúdo que também utilize uma análise da semelhança semântica (ver capítulo \ref{cap:semantic\_web} entre os itens envolvidos. Para isso será proposto um modelo de usuário que leve em consideração a descrição da narrativa do item. O objetivo é explorar que resultados podem ser obtidos realizando consultas ao serviço DBPedia \footnote{http://wiki.dbpedia.org}. Para a construção do SR foi escolhido o domínio de filmes, como motivador e exemplo de aplicação que tire proveito desse sistema. Com o acesso a esse serviço da web semântica, serão analisadas entidades procurando ontologias e relações presentes nas sinopses dos filmes, através dos dados ligados na DBPedia. Assim, pode ser comparada à similaridade de dois filmes através da presença ou relação de ontologias presentes na descrição. Como exemplo, caso um filme possua na sinopse o termo \textit{Morfeu} e o outro não, mas possua outras entidades sobre deuses mitológicos, como \texit{Zeus}, poderá ser criado um nível de relevância com o novo filme.

Inicialmente o usuário se registrará na aplicação desenvolvida por este trabalho com sua conta do Facebook. Em seguida o sistema irá coletar as informações do perfil do usuário referentes a livros que ele esteja lendo ou tenha marcado. Após coletar esses dados será necessário complementar as informações dos livros, já que o Facebook não possui informações satisfatórias e consistentes sobre o domínio. Para isso, será usado a plataforma do Google Books que oferece uma quantidade grande de informações do livro, inclusive autor, gênero e descrição da narrativa. Também será o serviço DBPedia para buscar informações de entidades presentes na descrição no livro para compor a classificação da recomendação.

Estrutura:

Neste capítulo foi motivado e introduzido o problema deste trabalho. Os próximos capítulos estão organizados da seguinte maneira: O Capítulo \ref{cap:recsys} apresenta os conceitos teóricos usados neste trabalho referentes a SR. O Capítulo \ref{cap:semantic\_web} apresenta conceitos sobre a web semântica. O Capítulo \ref{cap:proposal} apresenta a proposta do SR com a resolução de um modelo de usuário para o domínio de livros que leve em consideração a descrição da narrativa, discutindo a implementação. O Capítulo \ref{cap:evaluation} apresenta a avaliação do sistema, conclusões e considerações finais.

Sistemas de Recomendação:

Com a larga difusão da Internet é disponibilizado um enorme volume de informação para o usuário o que cria um desafio pela busca de informação. Por esse problema, empresas cresceram baseadas em sistemas de recuperação e filtragem, para personalizar experiência do usuário, como é o caso do Google. Neste capítulo será apresentado um panorama sobre SR, introduzindo os principais conceitos, tarefas e processos que o caracterizam.

Histórico:

Em razão da crescente dificuldade de usuários administrar a quantidade de informação, é comum decidir baseado em opiniões e recomendações de outros, especialmente quando há pouca experiência no assunto \citep{Resnick:1997:RS:245108.245121}. Conforme mais se expandia a tendência do uso de meios digitais de comunicação, mais rapidamente pessoas migraram de cartas para e-mails. A grande quantidade de e-mails acabava deixando o usuário imerso em documentos, dificultando o consumo do conteúdo. Em 1992, Xerox Palo Alto Research Center apresentou o sistema Tapestry\citep{Goldberg:1992:UCF:138859.138867} na revista mensal ACM Communications\footnote{https://cacm.acm.org/}, como proposta para lidar com o problema quantidade de e-mails.

O objetivo do sistema era prover listas de e-mails permitindo a inscrição dos usuários naquelas que fossem mais importantes. Alguns sistemas daquela época suportavam filtragem de e-mails baseado no seu conteúdo, mas os autores acreditavam que uma maneira mais eficiente seria com ajuda da avaliação de outros usuários. Interessante ressaltar que o termo “filtragem colaborativa” apresentado no artigo tornou-se comum, e só alguns anos depois surgiu a defesa do termo sistemas de recomendação, mais genérico, como defende \cite{Resnick:1997:RS:245108.245121} em seu artigo.

O sistema do Tapestry foi concebido para a filtragem colaborativa, onde colaborações de outras pessoas auxiliam a outros filtrarem gravando suas avaliações dos itens. Uma das vantagens da aplicação da filtragem colaborativa é que não depende da análise do conteúdo o que é especialmente útil para a análise itens complexos como vídeos, amplamente usado em serviços como o YouTube\footnote{https://www.youtube.com}. Um exemplo das recomendações no YouTube é na página “em alta” que mostra os vídeos em alta tendência baseada no feedback e visualizações. Em geral, as recomendações personalizadas são dispostas como uma lista de itens ranqueados. O termo “item” é o mais comum a ser denotado por SR para usuários, o que pode designar para diversos tipos, como filmes, livros, músicas etc.

Para o construir o ranque os SR tentam predizer qual é o item mais adequado àquele usuário \citep{Ricci2011}. Para realizar a tarefa o SR coleta dos usuários suas preferências que podem ser informadas de forma explícita, como avaliação de produtos, ou implícita interpretando suas ações como o histórico de navegação. O princípio dos SR é da dependência existente entre o usuário e sua atividade em torno dos itens \citep{Aggarwal:Intro:2016}. Como exemplo, se um usuário comprou um livro de ficção científica, é mais provável que também tenha interesse em outro livro de ficção científica. Dessa forma, o sistema lida com o problema da sobrecarga por filtrar itens que sejam menos prováveis do usuário gostar, baseando-se nas demonstrações do interesse prévio em outros itens, seja por outros usuários ou não.

O aumento da importância da Web como meio eletrônico, especialmente para o negócio eletrônico, também se mostrou como força para o desenvolvimento de sistemas de recomendação. Na Web o usuário pode informar o seu feedback sobre o que gostou ou não sobre determinado produto. Nesse contexto, a aplicação do SR não somente beneficiam o usuário, mas também para aqueles que o provem \citep{ ISINKAYE2015261}. Estudos \citep{Mykolas:2015a} demonstram que usuários optam por realizar compras online para poupar tempo. Contudo, com a explosão da variedade de informação disponível, em vez de agir em benefício começa a denigrir a experiência diminuindo a percepção do NFC. É bem aceito que ter escolha é bom, mas ter mais nem sempre é melhor \citep{Ricci2011}.

É importante ressaltar que por fornecer uma informação individualizada, que esteja mais alinhada com o perfil do usuário é o que diferencia os sistemas de recomendação de sistemas de recuperação de informação. Tradicionalmente o motor de buscas deve retornar tudo correspondente a um termo de pesquisa, porém cada vez mais o usuário entre no fator desses sistemas \citep{Burke:2002:HRS:586321.586352}. Sistemas como o Google\footnote{https://www.google.com}, vão além de retornar termos que batem com a consulta, mas também com a quantidade de outras páginas referentes, histórico de buscas, localização, compatibilidade com dispositivos móveis, além de introduzir informações extra a busca, com os quadros do knowledge graph\footnote{ https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html}.

Conceitos:

Sistemas de recomendação são sistemas de processamento de informação que lidam com diversos tipos de dados para construir recomendações que tentam prever a preferência do usuário \citep{Ricci2011}. Os dados tratam-se de basicamente de itens que serão apresentados a usuários na forma de recomendações. Técnicas de recomendação variam com dependência do tipo de conhecimento que pode ser extraído de um dado \citep{Ricci2011}. Dados de avaliações possuem pouca informação, o que resulta em técnicas diferentes em relação daquelas que dependem mais da descrição de um item ou relações com as atividades do usuário. Generalizando, SRs referem-se a três tipos de objetos: itens, usuários e transações que são as relações entre usuários e itens.

* Itens: Objetos que são recomendados. Podem ser caracterizados pela complexidade valor ou utilidade. O valor de um item pode ser positivo se é útil para o usuário, ou negativo se não é apropriado ou foi uma decisão errada de seleção por parte do mesmo. O usuário pode ser modelado e representado de diferentes formas, variando bastante em relação do domínio operado pelo SR. Toda vez que um usuário interage com um item constrói-se um custo cognitivo, o que pode entrar na relevância na construção do sistema, mesmo se o usuário não chega a adquirir o item interagido. Alguns exemplos de itens são: livros, notícias (baixa complexidade), computadores, viagens, vagas de trabalho (alta complexidade).
* Usuários: Usuários de um SR, podendo ter uma variedade de objetivos e características. São explorados uma série de informações variadas para personalizar as recomendações. A informação pode ser estrutura de diversas formas de acordo com o seu tipo, e a seleção de um modelo depende das técnicas a serem utilizadas. Modelos para sistemas de filtragem colaborativa pode usar apenas listas de avaliações de itens por usuários. O modelo de usuário cria o seu perfil, ou seja, armazena suas preferências e necessidades. Usuários também podem ser descritos baseados num padrão de comportamento, como o histórico de navegação na Web sua ou localização.
* Transações: Genericamente refere-se a transações gravadas das interações entre usuários e o SR. Transações podem ser vistas como um histórico de registros, um log de dados que armazena importantes informações geradas das interações com o sistema. Um registro pode conter a descrição do que foi consultado para uma recomendação particular de um item.

Tarefas de um Sistema de Recomendação:

Sistemas de recomendação são vistos como mais do que uma ferramenta para prover sugestões para itens que o usuário possa desejar. \citep{Ricci2011} em seu artigo introduziu uma série de funções que podem aplicadas em SRs.

* Aumento do número de itens vendidos: Uma das funções mais importantes para aplicações comerciais. O objetivo é ser capaz de vender outros itens comparados a aqueles que são vendidos sem qualquer tipo de recomendação. O objetivo é geralmente alcançado devido a itens que são prováveis de serem úteis a necessidade do usuário.
* Vender itens mais diversos: Também outra função de alta importância, na qual permite o usuário a selecionar itens que podem ser difíceis de encontrar. Num serviço de recomendações de filmes, como o Netflix\footnote{https://www.netflix.com}, o provedor estará interessado que os usuários encontrem conteúdos diversos, não somente os mais populares.
* Aumentar a satisfação do usuário: Quando um usuário encontra recomendações que sejam de seu interesse, eleva seu nível de NFC. Um SR bem desenvolvido permite uma combinação precisa de recomendações que juntos a uma interface com boa operabilidade, pode aumentar a noção subjetiva da avaliação de um sistema.
* Aumentar a fidelidade: Um usuário costuma ser leal a um site que, quando visitado, reconhece como um consumidor reincidente e o trata como um visitante de valor. É muito como para um SR levar em consideração as informações obtidas em prévias interações com o usuário. Consequentemente, por quanto mais tempo o usuário interage com o site, mais refinado seu modelo torna, ficando cada vez mais efetivo e customizado o resultado da recomendação.
* Melhor entendimento do que o usuário quer: Outra função importante, na qual pode ser influenciada por outras aplicações, é a descrição das preferências do usuário, seja coletada de forma explícita ou prevista pelo sistema. Um serviço pode decidir reutilizar esses dados do usuário para anunciar um produto em específico, derivado da coleta das informações de transações do SR.

Usuários também podem desejar um SR quando oferece suporte a suas tarefas ou objetivos. \cite{Herlocker:2004:ECF:963770.963772} é uma clássica referência no assunto, e define onze tarefas comuns que SR podem ajudar a implementar.

Encontrar bons itens: Recomendar a usuários alguns itens em ranque junto a uma predição de o quão o usuário possa gostar deles. Também comum no uso em sistema comerciais.

Encontrar todo os bons itens: Recomendar todos os itens que satisfazem as preferências do usuário. Neste caso é insuficiente apenas encontrar alguns bons itens. Esta função torna-se útil quando existe um número reduzido de itens, ou quando há uma razão crítica para fornecer informação, como em contextos de uso médico ou financeiro.

* Anotações em contexto: Dado um contexto, enfatizar alguns itens de uma lista a depender das preferências do usuário.
* Recomendar uma sequência: Recomendar uma sequência de itens invés de gerar uma única recomendação.
* Recomendar um grupo: Sugerir grupos de itens bem relacionados que possam ser da preferência do usuário.
* Apenas navegando: Mesmo que o usuário não possua a intenção de comprar um item, o SR deverá ajuda-lo a navegar pelos itens de maneira que que caixa no escopo de interesse do usuário.
* Encontrar um sistema de recomendação confiável: Nem todos os usuários podem confiar no sistema, dessa forma o mesmo pode oferecer teste de suas funcionalidades.
* Melhorar o perfil: Relativo a capacidade de o usuário prover dados ao SR sobre suas preferências. Tarefa fundamental para personalizar o sistema, caso contrário apenas seria possível oferecer recomendações que fosse relativa ao usuário comum.
* Expressar-se: Usuários podem não se importar com as recomendações, mas o sistema pode permiti-lo a contribuir com as avaliações e expressão de suas opiniões.
* Ajudar outros: Para alguns é importante contribuir com informações de suas opiniões e avaliações, pois compartilhando sua experiência pode ajudar a outros a decidirem.
* Influenciar outros: Alguns usuários podem ter apenas o objetivo de influenciar outros, ou até usar o SR para denegrir a imagem de alguns itens.

Técnicas de Recomendação:

As recomendações utilizadas no sistema são alcançadas através de algumas técnicas que possuem o objetivo de prever informações sobre itens e preferências de usuários. O SR irá produzir recomendações individualizadas como saída, ou será capaz de guiar o indivíduo de forma personalizada a modo de encontrar itens úteis \citep{ Burke:2002:HRS:586321.586352}. Apresentadas não somente como técnicas de filtragem colaborativa, \cite{Resnick:1997:RS:245108.245121}, introduz o termo mais genérico de sistema de recomendação, uma vez que tais sistemas podem explicitamente não utilizar recipientes que talvez sejam desconhecidos uns aos outros.

Para alcançar as principais funções de um SR, é necessário que o sistema seja capaz de identificar que itens possuem alguma utilidade para o usuário \citep{Ricci2011}. O sistema deve prever ou comparar a utilidade de itens, para decidir como recomendá-los. Dessa forma, as recomendações podem variar conforme os dados conhecidos de usuários e itens, podendo ter maior ou menor influência em uma função específica. Como exemplo, durante a etapa da predição pode ser considerado uma informação que não seja necessariamente personalizada, como apenas recomendar itens mais populares. De posse de poucas informações, ou não conclusivas, a premissa é basear-se num item que tem boa aceitação, ou seja, que é útil para muitos, com uma recomendação provável ao usuário genérico.

Ampliando ao já apresentado Tapestry\citep{Goldberg:1992:UCF:138859.138867}, nem todas as técnicas precisam ser baseadas nas informações de preferências de outros usuários. Na literatura já foram discutidos diversas técnicas, como as apresentadas nos trabalhos de \citep{Ricci2011} e \citep{ Burke:2002:HRS:586321.586352}. Dentre essas abordagens estão:

Filtragem Colaborativa: O sistema agrega avaliações ou recomendações, reconhecendo características comuns entre usuários baseando-se nos itens de suas avaliações.

Baseada em conteúdo: Objetos de interesse são definidos pela associação de suas características. O sistema aprende e recomenda itens similares ao que usuário demonstrou interesse no passado.

Demográfico: Objetivam categorizar o usuário baseado nas informações pessoais dos usuários. Recomendações são baseadas nas classes demográficas dos usuários.

Baseada em conhecimento: Realizam sugestões de itens baseadas em inferências das preferências do usuário.

Abaixo será apresentado em maiores detalhes o funcionamento das técnicas de filtragem colaborativa e baseada em conteúdo

Filtragem Colaborativa:

Recomendação com filtragem colaborativa (FC) é uma das técnicas mais familiares e já implementadas \citep{Ricci2011}. A similaridade das preferências e desejos de dois usuários é calculada baseada na similaridade do histórico de avaliações dos usuários. A premissa do método é de que a opinião de outros usuários pode ser selecionada e agregada de forma a prover predições razoáveis ao usuário alvo \citep{HCI-009}. Como exemplo, intuitivamente assume-se que se usuários que concordam sobre a qualidade de um filme que João gosta, então João provavelmente gostará de outros filmes que outros usuários avaliaram, mas não assistiu.

O perfil de um usuário na FC pode ser continuamente aprimorado conforme o usuário interage com sistema, podendo levar o tempo de uso como fator de avaliação. Em alguns casos a avaliação pode ser apenas binária (gostei ou não), ou então de valor real que determina um grau de utilidade. Nesse caso nas avaliações do usuário, o sistema deverá modelar uma função R(u,i) representado o grau de utilidade do item i para o usuário u. Basicamente, a tarefa do sistema é estimar um valor de R baseado nos pares de usuário e item. Dessa forma, avaliando os dados dessas predições de R para o usuário alvo, o sistema recomendará uma quantidade os itens com as maiores utilidades previstas.

Tipicamente, conforme apresentado por \cite{Burke:2002:HRS:586321.586352}, FC divide-se em dois método principais: vizinhança e baseados em modelo. No método da vizinhança o foco é no relacionamento entre itens ou usuários, utilizando informações armazenadas com o tempo. O método aborda modelos através da análise da preferência armazenada das classificações de usuário-item, pela avaliação de outros itens similares. Já o método baseado em modelo é criado diretamente do histórico das avaliações para aprender as preferências do usuário, podendo-se usar uma quantidade diversa de técnicas para o aprendizado, como redes neurais. O objetivo é compreender e extrair das interações usuário-item características de destaque para o sistema, podendo criar classes de preferências dos itens.

Filtragem baseada em conteúdo:

Ao contrário da filtragem colaborativa, sistemas de recomendação baseados em CBF (Filtragem Baseada em Conteúdo), seleciona itens baseados entre as relações de seus conteúdos e as preferências do usuário. A CBF é uma continuação natural das pesquisas nos sistemas de filtragem de informação, \cite{Burke:2002:HRS:586321.586352}. O método utiliza-se da intuição de que se o usuário demonstrou interesse em certos itens com determinados atributos, é provável de também ter interesse em outros itens de mesmo atributo ou semelhante. Como exemplo, se João gostou dos filmes com o ator Tom Cruise, é provável que vá gostar de outros filmes com o mesmo ator. Os sistemas de CBF foram desenhados para explorar cenários com itens que podem ser descritos com um conjunto de propriedades ou atributos \citep{Aggarwal2016}.

Nessa abordagem o sistema deverá aprender do perfil do usuário seus interesses baseados na combinação das características presentes nos objetos que ele avaliou ou marcou. O tipo do perfil utilizado no sistema dependerá do método utilizado aplicado. A informação das preferências do usuário pode manifestar-se de forma explícita, onde existem avaliações ou indicações dos itens favoritos, ou de forma implícita como itens que o usuário comprou. Nos métodos aplicados na CBF, as descrições dos itens avaliados são usadas como dados de treinamento para criar uma classificação específica para o usuário \citep{Aggarwal2016}. Os perfis da filtragem baseada no conteúdo são modelos de longo prazo, onde mais dados são atualizados conforme mais evidências do usuário são observadas, \cite{Burke:2002:HRS:586321.586352}.

Apesar da descrição do conteúdo, ou seja, atributos particulares dos itens, sejam o centro da análise da utilidade de novos itens para recomendação, a avaliação de outros usuários tem significativo impacto no sistema \citep{Aggarwal2016}. Essa característica apresenta tanto vantagens como desvantagens. Por um lado, num contexto da *partida a frio,* onde há pouca informação disponível sobre as avaliações dos usuários, há margem de utilização enquanto houver outras suficientes informações das preferências do usuário. Mesmo quando um item é novo ou desconhecido, o sistema ainda pode aproveitar suas características para recomendar novos itens, algo que não é possível apenas baseando-se nas avaliações de outros usuários.

Assim, sistemas de CBF são tipicamente utilizados quando há suficiente informação das preferências do usuário disponíveis. Particularmente, são de maior fácil utilização quando usados em domínios com dados não estruturados e ricos em textos, como páginas da Web.

Comparação das técnicas de recomendação:

Todas as abordagens dos SR possuem vantagens e desvantagens, dependendo de questões como novos itens, usuários, e quantidade de informação disponível sobre os dois. Em relação a novos usuários, como recomendações partem da comparação de informações do usuário alvo e outros usuários, quanto menos avaliações o sistema possuir, mais difícil será a classificação. Já para novos itens, o problema surge em domínios em constante atualização e novas informações e onde cada usuário pouco avalia. Também pode ser visto como o problema do early rater, uma vez que a pessoa que avalia primeiro, pouco se beneficia.

\cite{Burke:2002:HRS:586321.586352} apresentou alguns pontos comuns das diferenças desses sistemas:

Sistemas baseados em filtragem colaborativa: Dependem da sobreposição de avaliações através dos usuários e possuem dificuldades quando há escassez dessas avaliações dos itens. O problema ressalta que as técnicas colaborativas melhor servem quando a densidade de interesses de usuários é alta através de um universo de itens que não mudam rapidamente.

Sistemas baseados em conteúdo: Possuem o problema da partida a frio, onde o sistema não acumulou dados suficientes para construir uma recomendação confiável. Também são limitados pela quantidade de informações disponíveis e associadas aos itens. Isto acaba colocando a técnica muito dependente da descrição dos dados. Uma grande desvantagem em relação a abordagem colaborativa é que a abrangência de gêneros, onde deixa o usuário sujeito ao mesmo tipo de conteúdo. A depender da CF, pela a avaliação de outros usuários é possível recomendar itens “fora da caixa”.

Aplicações de Sistemas de Recomendação:

O sistema Tapestry foi um marco inicial no desenvolvimento de aplicações, introduzindo a filtragem colaborativa. Hoje, SR são quase que obrigatórios para muitas lojas online e serviços de entretenimento, tornou-se algo comum e já disseminado entre usuários. A seguir será apresentado algumas aplicações em destaque que usam sistemas de recomendação.

Netflix:

Com a evolução da internet, as mídias físicas para consumo de entretenimento começaram a decair, especialmente para filmes. O avanço na conexão da banda larga trouxe o modelo do \texit{streaming}\footnote{Transmissão contínua de mídia, (<https://directradios.com/streaming/)>} que possibilita o usuário a assistir o conteúdo a qualquer momento, lugar, sem ter que necessariamente sair de sua residência para ir à uma locadora, por exemplo. Embora o Netflix \footnote{https://www.netflix.com}, tenha iniciado no ramo de aluguel de DVDs\citep{keating2012netflixed}, a companhia rapidamente abandonou este modelo e partindo para a transmissão de filmes e em seguida para produção de seus próprios filmes e séries. Dessa forma, o serviço de filmes e séries, cresceu ocupou espaço das televisões, cinemas e alcançou diversos países.

Com a crescente quantidade de títulos disponíveis na plataforma e também de usuários, logo o serviço desenvolveu seu próprio sistema de recomendações vídeos, baseado nas avaliações de usuários. Em outubro de 2006 a companhia publicou um concurso pelo melhor sistema de filtragem colaborativa que poderia superar a precisão de seu SR, o Cinematch \citep{bennett2007netflix}. Neste ponto o serviço já tinha lançado um banco de dados contendo 100 milhões de avalições de usuários e 18 mil títulos. O Cinematch analisava as avaliações acumuladas dos usuários semanalmente usando uma variante da correlação de Pearson, com todos os outros filmes para determinar uma lista de filmes similares. Sendo assim, conforme o usuário provia avaliações, o sistema computava uma regressão baseada nessa correlação para determinar uma predição única personalizada. Caso não houvesse nenhuma predição personalizada a média de todas as avalições é usada. As predições eram apresentadas como conjunto de 5 estrelas.

O desempenho do Cinematch é medido principalmente pelo cálculo da raiz do erro quadrático médio, RMSE \citep{Herlocker:2004:ECF:963770.963772}, das predições do sistema contra as avaliações que os usuários informam. Com os sistemas propostos no concurso, a companhia propôs um prêmio para aqueles que conseguissem melhorar a precisão em 10%. Nesse ano de 2017, a companhia migrou seu sistema de avaliação das tradicionais 5 estrelas para uma avaliação binária, o \textit{Like} e \textit{Dislike} \citep{ VarietyNetflix:2017}. Segundo a companhia, os usuários confundiam a avaliação de 5 estrelas, pois na verdade eram sempre as predições avaliadas para o filme, assim agora as predições aparecem no formato de porcentagem de relevância e a avaliação do usuário é indicada pelos símbolos do gostei ou não gostei. Também as predições passaram a serem baseadas apenas no histórico e comportamento do usuário e não mais na média em relação às outras pessoas

Skoob:

Em janeiro de 2009, o analista de sistemas Lindeberg Moreira realizou sua ideia de criar uma plataforma em que pessoas socializassem o ato da leitura \citep{SkoobSocializando:2009}, o Skoob\footnote{ <https://www.skoob.com.br>}. O sistema criado trata-se de uma rede social para leitores no Brasil. Na plataforma, o usuário montará uma estante virtual realizando buscas pelos livros e em seguida indicar o que já fez com o livro, se já leu, se lerá ou está relendo. Após a seleção dos livros os usuários poderão avaliar seus livros, podendo até escrever resenhas completas ou de capítulos dos livros, compartilhando com outras pessoas na plataforma.

A rede social, conta com algumas mecânicas para ajudar usuários a encontrar livros, com um sistema busca de livros, recomendação com filtragem colaborativa baseada nas avaliações de usuários, dos marcados como mais lidos, lendo, quero ler entre outros. A plataforma também conta com um sistema que indica livros similares. Todos esses processos levam não somente levam a questão da socialização da leitura e escrita entre indivíduos que compartilham interesses, surgidas a partir da aplicação, mas passam a influenciar a forma como usuários passam a tratar a leitura fora do ambiente da comunidade virtual, é o que aponta \cite{SkoobUFPE:2010}.

Sumário

Neste capítulo, foi apresentado um panorama geral sobre os sistemas de recomendação. Incialmente abordando o histórico envolvido e motivações na criação dos conceitos envolvidos do tema. Em sequência foi aprofundado e explicado os conceitos utilizados nesses sistemas. Então, foi apresentado as tarefas e técnicas utilizadas. Também foi aprofundado algumas diferenças e dificuldades entre as principais técnicas de recomendação. Por fim, foi mostrado- exemplos de sistemas que utilizam esses sistemas de recomendação. No capítulo 3 será discutido sobre os conceitos envolvidos na Web Semântica, bem como os princípio dos dados ligados e o serviço da DBPedia \footnote{http://wiki.dbpedia.org/}.

Web Semântica:

A introdução e expansão da \textit{World Wide Web} possibilitou acessar e publicar uma grande variedade de conteúdo, seja para o consumo de entretenimento, exposição de opiniões, compras online. O crescimento da rede tornou-se tão grande que é latente a necessidade para seus usuários de encontrar informações. Para os usuários foram criados e desenvolvidos os indexadores de páginas, como o Google\footnote{https://www.google.com}, Yahoo\footnote{https://www.yahoo.com}, Bing\footnote{https://www.bing.com}. Tais sistemas facilitam encontrar informações em serviços populares na internet. Entretanto, e se quiséssemos encontrar algum médico de confiança para marcar uma consulta, levando em consideração minha agenda de compromissos? Ou então se estamos realizando um trabalho escolar e queremos encontrar os reis do século XV? Essas pesquisas, certamente são mais complicadas, e resultados de buscas tradicionais levam a informações fragmentadas com uma série de outras buscas separadas para alinhar todo o conhecimento e semântica envolvidos nessas tarefas. É nesse ponto que entra o conceito da Web Semântica, como uma extensão da já existente.

O conteúdo da Web tradicional é fundamentalmente desenvolvido para humanos lerem, não para máquinas manipularem de forma produtiva, significante \citep{bernerslee2001semantic}. Originalmente desenvolvida para compartilhar e apresentar conteúdo de forma que fosse possível interagir e navegar entre hipertextos e hipermídia, a WWW torna fácil a apresentação de layouts. É possível estruturar um documento com um cabeçalho, um link para outra página, entretanto, dificilmente as máquinas poderão processar semanticamente que informações estão disponíveis e podem ser organizadas naquela página ou site, como exemplo esta é a página de João com link para seu currículo informando que possui especialização em cardiologia. Todas essas informações podem até serem compreendidas por humanos ao associar a semântica das entidades presentes numa página, analisando com links e seus relacionados, mas para a máquina não há uma estrutura comum e eficiente que leve a essas mesmas conclusões.

O objetivo da Web Semântica é de estender a WWW, aproveitando a enorme variedade de dados já existente, mas agregando uma nova camada de metadados que possibilitem o processamento pela máquina e agentes de forma a compreender a semântica das informações ali apresentadas. Assim, a Web Semântica trata-se de prover formatos para integração de dados de diferentes fontes \citep{SemanticWebW3C}, onde a Web tradicional mantém-se como o meio de publicação e interconexão de documentos, e na contraparte semântica, armazena-se como os dados se relacionam com objetos e coisas do mundo real. Um agente pode se deparar com uma página de clínica na Web e não apenas compreenderá que possui palavras como “tratamento, terapia, remédios, médicos”, como tipicamente é encontrado na Web tradicional, mas também saber o Dr João trabalha nessa clínica nas segundas e quartas com horários no formato \textit{dd/mm/YYYY}.

Arquitetura e formato de dados

O funcionamento da Web Semântica depende da capacidade de máquinas acessar coleções estruturadas de informações e dados e regras de inferência para executar raciocínio automatizado \citep{bernerslee2001semantic}. O desafio é de como representar conhecimento. Inicialmente o desenvolvimento desses sistemas utilizaram uma abordagem centralizadora, requerendo que as partes envolvidas compartilhem exatamente as mesmas definições de conceitos comuns ou hierárquicos. Entretanto, com a quantidade de conteúdo existente hoje em diferentes línguas, controle centralizado é inviável. Contrastando essa visão inicial, na Web Semântica cria-se linguagens para regras as quão expressivas for necessário para que a Web seja tão ampla como desejado. Com um sistema que não seja centralizado é possível que não se responda todas as perguntas ou seja encontrado todas as informações, mas permite que regras sejam usadas para criar inferências e escolher o curso de ações para responder tais perguntas.

Com esses fundamentos os pesquisadores da Web Semantic, em especial o \textit{World Wide Web Consortium}, desenvolveram uma série de padrões e formatos de dados para o uso na Web. O intuito é possibilitará máquinas compreender documentos com dados semânticos e não discursos e textos criados pelo homem. Uma tecnologia muito importante para o desenvolvimento da representação do conhecimento e protocolo de comunicação entre máquinas, foi o XML. Com o XML é possível que qualquer um sejam capazes de criar suas próprias \textit{tags} e estrutura de um documento com definição de cada termo presente de forma arbitrária. Desse ponto de vista o XML é fundamental como um padrão de comunicação entre máquinas. Anos seguintes, a W3C foram introduziu três importantes tecnologias presentes no cenário atual da Web Semântica: RDF, SPARQL, OWL.

RDF

Resource Descripton Framework é um modelo de dado para a Web que facilita a junção de dados mesmo que seu \textit{schema} \footnote{https://en.wikipedia.org/wiki/XML\_schema} difiram, além de permitir a sua evolução sem requerer que seus consumidores tenham que se adaptar \citep{ W3CRDF}. No RDF a estrutura da Web de links é estendida para usar as URIs para nomear a relação entre qualquer coisa, com ambas as pontas do link, formando o que é conhecido como a tripla.

O uso da URI é especialmente notável para o uso na Web, uma vez que não é possível apenas se basear em valores literais, mesmo para representar um atributo de algo, já que é desejado ter a definição e estrutura do mesmo podendo considerar um domínio em específico. Como exemplo com uma URI é possível identificar de forma única o predicado “título” que se refere ao título da função em uma empresa, e não um título de filme. Então, a tripla forma um grupo de três entidades que expressam uma declaração sobre o dado semântico sob a forma de “sujeito, predicado, objeto”. Com essa estrutura de links é formado um grafo direcionado, com \textit{labels}, aonde suas arestas representam o link nomeado entre dois recursos representados pelos seus nós análogos.

OWL

Ontology Web Language é uma linguagem para definir e instanciar ontologias na Web \citep{OWLW3C}. Um programa que deseja comparar ou combinar informações entre dois bancos de dados com URIs distintas, deve saber se termos podem ser usados para descrever o significado da mesma coisa \citep{bernerslee2001semantic}. O objetivo é que um programa descubra o significado comum seja para o que for encontrado entre os conjuntos de dados. A solução proposta na Web Semântica para esse problema é a utilização de uma coleção de informações denominada de ontologias. Na filosofia uma ontologia tem por objeto o estudo das propriedades, tratando da natureza da existência. Entretanto, no campo da Inteligência artificial e Web, define os termos básicos e relações que compreendem um vocabulário de um domínio, bem como regras para combinar termos e as relações para definir extensões desse vocabulário \citep{Patil:1992:DKS:3087223.3087302}.

Em essência a ontologia é um documento que define formalmente as relações entre termos. As ontologias podem ser vistas de forma semelhante à hierarquia de classes na programação orientada a objetos. Tipicamente uma ontologia para a Web possui uma taxonomia e um conjunto de regras de inferência. A taxonomia define classes (ou conceitos) de objetos e suas relações, sendo assim, um endereço pode ser definido como um tipo de localidade e o código de uma cidade pode ser definido para ser aplicado apenas a localizações, entre outros exemplos.

A linguagem OWL provê três sub-linguagens, OWL Lite, OWL DL, OWL Full como apresentado pela \cite{OWLW3C}.

OWL Lite: Para primariamente a criação hierárquica e simples de limitações de \textit{features}. Como exemplo, é possível oferecer suporte a limitações de cardinalidade que só permitam valores de 0 ou 1. É mais simples de prover suporte.

OWL DL (descrição lógica): Oferece suporte a uma expressividade máxima sem perder a completude computacional (todas as implicações são garantidas para serem computadas), decidibilidade (todos os cálculos finalizaram em um tempo finito). Inclui todas as construções com restrições e separação de tipos (uma classe também não pode ser indivíduo ou propriedade, uma propriedade também não pode ser um indivíduo ou uma classe).

OWL Full: Oferece o máximo de expressividade e é sintaticamente livre do RDF sem garantias computacionais. Nessa linguagem uma classe pode ser tratada simultaneamente como uma coleção de indivíduos ou indivíduo como todo. Então, a OWL Full permite uma ontologia ter seu significado ampliado ao pré-definido (RDF ou OWL) vocabulário.

Todas as sub-linguagens são extensões de sua predecessora, sendo assim cada ontologia válida em OWL Lite é uma ontologia válida em OWL DL que por sua vez é uma ontologia válida em OWL Full \citep{ OWLW3C}. É notável destacar que o inverso das relações não é verdadeiro. Completando, todo documento OWL é um documento em XML construído com o RDF.

Estrutura de um documento

Com a OWL é possível descrever de forma natural classes e relacionamentos entre documentos e aplicações na Web \citep{OWLReport:2005}. Os termos descritos devem estar dispostos de tal maneira que não cause ambiguidade, assim é necessário que seja informado quais vocabulários serão empregados. Para o uso de vocabulários a \citep{OWLW3C} deve-se definir no cabeçalho do documento os \textit{xml namespaces} \footnote{No XML, os namespaces são nomes únicos para elementos e atributos no documento. Para resolver as ambiguidades e facilitar as referências antes dos nomes são utilizados prefixos}, conforme mostrado no código fonte \ref{cod:owl\_head}.

Acrescentando, a W3C recomenda incluir um cabeçalho XML no documento que preceda as definições das ontologias como apresentado no código fonte \ref{cod:owl-head}

Por último será informado o cabeçalho da ontologia junto a suas propriedades. Nesse cabeçalho é importante fornecer informações sobre ela própria. Para descrevê-las utiliza-se as propriedades do OWL, uma vez que a ontologia é um recurso, assim demonstrado no código fonte \ref{cod:owl-ontology-head}

Dentro da definição da ontologia poderão ser informados as classes e indivíduos relacionados como as propriedades e suas relações. As propriedades podem ser descritas como transitivas, simétricas, funcionais ou inversamente funcional. Como exemplo, numa propriedade transitiva de subordinado, sem é dito que João é subordinado de Pedro e Pedro é subordinado de Maria, portanto João é subordinado de Maria. No código fonte \ref{owl-props} é demonstrado a declaração desse tipo de propriedade.

Estrutura na rede semântica

A introdução das tecnologias para alcançar os princípios idealizados na Web Semântica são implantados em camadas. De acordo com \cite{bernerslee2001semantic} é possível dividir esses serviços em três grandes camadas, como demonstrado na figura \ref{fig:sw-layers}. Na camada de estrutura os dados são organizados e é definido seus significados, na qual utiliza-se as triplas do RDF. A camada com os esquemas estão as ontologias, utilizando-se o OWL para a representação de conceitos, inferências através das taxonomias e conjunto de regras. Por último na camada lógica são definidos para se fazer inferência sobre os dados. Dessa forma, o desenvolvimento dessas tecnologias e padronização dos formatos foi formulado pela W3C uma pilha das camadas da Web Semântica confiável, conforme mostrado na figura \ref{fig:sw-w3c-stack}

Dados ligados

A evolução da WWW tornou cada vez mais acessível a publicação e acesso a documentos pela navegação no espaço global, através links dos hipertextos \citep{Bizer2009}. Com os navegadores da Web pode-se passear pelos links nesse espaço e em especial com o uso dos buscadores, que indexam páginas para facilitar a recuperação. Tais mecanismos já estão amplamente difundidos na publicação de documentos, mas quando comparados aos dados em si, esses princípios ainda foram timidamente aplicados. Assim, com o crescimento da Web Semântica trouxe a ênfase em criar uma Web para os dados, capaz descrever entidades individuais presentes nos documentos, conectando-se por links categorizados para relacionar tais entidades. O objetivo não é somente colocar dados na Web, mas utilizar links que ambas máquinas (principalmente) e humanos possam navegar.

Suportando essa evolução da Web \citep{LinkedData:2006} introduziu um conjunto de melhores práticas para a publicação e conexão de dados estruturados na Web, denominado de \textit{Linked Data} (dados ligados). A adoção dessas práticas permite a extensão da Web com um espaço de dados global conectado de diversos domínios, desde pessoas, livros, publicações até dados governamentais dos mais variados assuntos. Com essa Web de dados surge a oportunidade para novos tipos de aplicações \citep{Bizer2009}, como navegadores customizados para um determinado domínio podendo saltar entre diferentes fontes de dados.

Resumidamente, a \cite{LinkedDataW3C} define que para a Web dados ser uma realidade é necessário que estejam disponíveis em padrões de formatos que sejam buscáveis e manipuláveis pelas ferramentas e tecnologias da Web Semântica. Complementando, é preciso também ter acesso ao relacionamento de dados. O conjunto de \textit{datasets} inter-relacionados na Web, para criar links tipificados entre dados de diferentes fontes é o que se denomina de dados ligados.

Ao contrário dos documentos HTML na Web dos hipertextos, os dados ligados se baseiam-se nos documentos contendo dados em RDF. Assim são construídos links que são tipificados para realizar declarações sobre coisas arbitrárias no mundo. \cite{LinkedData:2006} enumerou um conjunto das regras para a publicação dos dados, de tal modo a atender os princípios dos dados ligados:

* Usar URIs para nomear coisas.
* Usar HTTP URIs para que pessoas possam procurar seus nomes.
* Quando alguém procura uma URI, forneça informação útil, utilizando os padrões como RDF e SPARQL.
* Inclua links para outras URIs, para que assim eles possam descobrir mais coisas.

Um exemplo notável do uso das dados ligados, é o projeto da DBPedia\footnote{ http://wiki.dbpedia.org} que essencialmente torna o conteúdo da Wikipedia\footnote{https://www.wikipedia.org} disponível em RDF.

Similaridade Semântica

A similaridade semântica entre dois termos, recursos, itens ou documentos é uma métrica para medir a distância de seus significados ou semântica, dado uma ontologia \citep{Slimani2013}. O objetivo é estabelecer características em comum entre dois conceitos. A distância entre dois conceitos para humanos pode não ter uma definição formal, já que se pode criar juízo de valor diferentes no relacionamento entre eles. Como exemplo, para uma pessoa a maçã e a banana podem estar mais relacionadas do que a maçã e a pera para outra. A similaridade e relação semântica podem por vezes determinadas como a mesma coisa, ambas como métricas de distâncias entre termos, contudo a similaridade semântica é mais específica \citep{Slimani2013}. A relação semântica é calculada usando um modelo de espaço vetorial e uma métrica de similaridade, como a similaridade do cosseno \ref{eq:cosine\_sim}.

Entretanto, para a similaridade semântica é levado em consideração relações léxicas de sinonímia e hiperonímia onde o significado é abrangido pelo outro termo mais geral (como carro e veículo) \citep{Gracia2008}. Na prática, a similaridade semântica pode ser medida pelo menor caminho entre dois termos utilizando suas ontologias associadas. Para realizar a similaridade podem ser usadas diversos tipos de ontologias. \cite{Slimani2013} descreve dois principais tipos de ontologias usadas para medir similaridade.

* Propósito genérico: \textit{Wordnet}\footnote{https://wordnet.princeton.edu} é um banco de dados que modela o conhecimento léxico da língua inglesa. Nomes, verbos, adjetivos e advérbios são agrupados em conjuntos sinônimos, onde cada um expressa um conceito distinto. Essa ontologia pode ser utilizada para criar um \textit{score} de similaridade. Pode ser considerada um ontologia para termos de linguagem natural.
* Domínio específico: \texit{ULMS}\footnote{https://www.nlm.nih.gov/research/umls} é um sistema de linguagem médica com uma rede semântica de ontologias de multiuso, multilíngue para biomedicina, conceitos e assuntos relacionados à saúde. O banco de dados do sistema possui uma coleção de vocabulários de conceitos e termos e seus relacionamentos que são denominados de \texit{Metathesaurus}. Cada Metathesaurus é classificado como pelo menos uma categoria semântica.

Medidas de Similaridade Semântica

Na literatura já foram apresentadas algumas medidas de similaridade semântica, mas comumente existem três fatores principais \citep{Slimani2013} que podem ser associados na topologia (i.e. nós do grafo direcionado) das ontologias: \textit{path length}, \textit{depth}, \textit{density}. Todos esses fatores afetam a medida da distância semântica, assim como as características entre dois termos, que podem aumentar ou diminuir as medidas de acordo com suas semelhanças. Quanto a densidade entre dois termos trata-se do número de filhos dos quais pertencem ao menor caminho (\textit{path}) da raiz ao mais específico conceito entre esses termos. Os fatores que influenciam nas medidas levam a definição de uma classificação que podem ser divididas em quatro principais \citep{Slimani2013}: baseadas em estrutura, conteúdo, recursos ou características e as híbridas que combinam as características estruturais (\textit{path length}, \textit{depth}, \textit{density}) e alguma outra abordagem.

Baseadas em Estrutura:

As medidas baseadas em estrutura (\textit{Structured-baser ou Edge-counting}, utilizam funções que computam a similaridade baseada na hierarquia e estrutura da ontologia, ou seja, onde um conceito é definido como “é parte de”, “é um” etc. A função calcula o tamanho do caminho que liga os termos e seus posicionamentos no grafo direcionado da ontologia. Quanto mais dois conceitos são, mais \textit{links} existem entre eles. Dentre as medidas baseadas em estrutura se destacam:

* \textit{Shortest Path}: A medida do menor caminho é um tipo de medida de distância que é primariamente voltada para lidar com hierarquias em redes semânticas. A função da similaridade entre conceitos $C\_1$ e $C\_2$ é definida como:

Baseadas em Conteúdo:

As medidas baseadas no conteúdo, são aquelas que utilizam a informação do conteúdo para medir similaridade. O conteúdo de um conceito é definido pela frequência de termos dado uma coleção de documentos. Grande parte das medidas deste tipo utilizam a informação compartilhada de dois conceitos pais $C\_1$ e $C\_2$, dos quais $S(C\_1; C\_2)$ é o conjunto de conceitos que os engloba. O menor $p(C)$ é utilizado quando há mais de pai em comum que $C$ é o \textit{most informative subsume} (MIS), ou seja, o conceito mais informacional que os engloba.

Baseadas em características ou recursos:

Baseia-se em características ou recursos (\textit{Featured-based}), que partem do princípio de valorizar informações importantes em relação ao conhecimento sobre um termo. A medida assume que os conceitos são descritos por termos indicando suas propriedades ou \textit{features}. A similaridade entre dois conceitos é definida por uma função que relaciona suas propriedades ou relacionamentos a outros termos similares na hierarquia da ontologia. \cite{Tversky:1977} apresenta uma medida \textit{Feature-based} de termos para calcular a similaridade entre diferentes conceitos, contudo o posicionamento desses termos na taxonomia e a informação do conteúdo não são levadas em consideração. A proposta é de que com termos descritos por um conjunto de palavras como propriedades do conceito, então as que são em comum tendem a aumentar a similaridade, enquanto as que não são em tendem a diminuí-la. Dessa forma, é definida uma equação onde $C\_1$ e $C\_2$ representam o conjunto de descrições dos termos e $\alpha \in [0,1]$ é a relação de relevância das características que não são em comum. O valor de \alpha aumenta o quão mais em comum dois conceitos são, e decresce com suas diferenças, e não é necessariamente uma relação de simetria, mas mais baseada na similaridade \citep{Slimani2013}.

DBPedia

A DBPedia (DB para \citep{database}) é um esforço colaborativo para a extração de dados do Wikipedia para publicação de dados essencialmente em RDF \citep{Auer:2007:DNW:1785162.1785216}. Um dos objetivos é possibilitar que outros explorem a criar uma experiência da enciclopédia mais abrangente, utilizando serviços e aplicações na Web Semântica. O projeto é um dos mais famosos que aplica os conceitos de dados ligados, onde sua importância não somente é dada pela publicação dos dados da Wikipedia, mas também da incorporação de links de outros \texit{datasets}. De fato, o DBpedia, por muitas vezes é considerado um núcleo dentro da iniciativa do LOD.

O projeto tem o foco em converter o conteúdo presente do Wikipedia em conhecimento estruturado utilizando as tecnologias da Web Semântica para que outros agentes possam explorar realizando consultas e ligando a outros conjuntos de dados \citep{ Auer:2007:DNW:1785162.1785216}. Assim, o projeto cobre uma das limitações da Wikipedia que é a dependência de apenas ter busca em texto livre. Desse papel, o projeto promove três importantes contribuições:

Desenvolvimento de um \textit{framework} para extração de informação, o qual converte o conteúdo da Wikipedia em RDF.

Prover o conteúdo da Wikipedia como um largo, multi-domínio \textit{dataset} de RDF. São mais de 100 milhões de triplas já mapeadas.

Interligar o DBpedia com outros conjuntos de dados abertos, o que expande a contagem das triplas RDF para mais bilhão.

Desenvolvimento de uma série de interfaces é módulos de acesso para que tal \textit{dataset} possa ser acessado por serviços da Web ligado a outros sites.

|  |  |
| --- | --- |
| ★ | Disponível na Web (qualquer formato), mas com licença aberta para ser *Open Data*. |
| ★★ | Disponível como dado estruturado legível por máquinas (exemplo, excel invés de imagem capturada de uma tabela) |
| ★★★ | Como o (2), além do formato não proprietário (exemplo, CSV invés de excel). |
| ★★★★ | Todos os anteriores, além do uso de padrões abertos da W3C (RDF e SPARQL) para identificar coisas, para que assim pessoas possam apontar para seus dados |
| ★★★★★ | Todos os anteriores, além de ligar seus dados à outros dados de pessoas para prover um contexto |

O LOD (Linked Open Data) tornou-se o projeto de maior adoção dos princípios dos dados ligados \citep{Bizer2009}, sendo um esforço colaborativo iniciado em 2007 para suportar as definições suportadas pela W3C. O motim para o início da colaboração era mapear os dados da Web identificando os conjuntos que já estavam disponíveis sob licença aberta. O projeto inclui dados de várias fontes, como a Wikipedia\footnote{https://www.wikipedia.org}, Geonames\footnote{ <http://www.geonames.org>}, Wordnet{https://wordnet.princeton.edu} entre diversos outros de múltiplos domínios, alcançando um impressionante grafo como mostrado na figura \ref{fig:lod-graph}.

Um Sistema de recomendação semântico baseado em conteúdo:

Desde de tempos o homem busca construir ferramentas e máquinas que facilitem, ampliem, sustem sua capacidade de trabalho e produção. Com o advento dos computadores e dos programas de máquina, o \textit{software} tornou-se essencial para a contínua demanda de problemas e desafios da crescente população global. Como avaliado por \cite{Sommerville2010}, o software não se restringe a propriedades materiais das leis da física ou por processos de manufatura. Por um lado, este fato simplifica a engenharia de software devido falta de restrições físicas, mas o torna complexo e de alto custo na realização de mudanças. Dessa forma, com a crescente quantidade de computadores e a diversidade de dispositivos, é cada vez mais relevante a qualidade de software.

O tema não é novo e já é levantado desde a década de sessenta, como na conferência NATO \citep{NR68} sobre problemas e desafios no desenvolvimento de software. A qualidade de software não somente aborda problemas do ponto de vista da coordenação do desenvolvimento, viabilizando a execução pelas máquinas, mas também estuda a importância da legibilidade a fim facilitar a manutenção e compreensão por humanos. Assim, é de suma importância documentar funcionalidades, decisões técnicas a serem utilizadas no processo do desenvolvimento de software, para que outros possam entender o trabalho que está sendo construído \citep{Pressman2009}.

Neste capítulo será apresentado os requisitos funcionais e não funcionais para um sistema de de recomendação semântico baseado em conteúdo. Será discutida as tecnologias, comportamentos, modelos e arquiteturas utilizadas. Por fim, será demonstrado o funcionamento de um protótipo de um site para recomendações de filmes, implementando esse sistema.

Requisitos

Os requisitos de um sistema são descrições do que deve fazer, suas funcionalidades e serviços que restringem sua operação \citep{Sommerville2010}. Tais requisitos são uma reflexão das necessidades dos consumidores do sistema e definem um propósito específico, como cadastrar um usuário, encontrar produtos etc. Os requisitos de software, então, tratam-se de descobrir, analisar e documentar tais serviços e restrições para a operação do produto final. A descrição desses requisitos deve ser clara e objetiva, para apenas descrever o objetivo final da funcionalidade a ser desenvolvida.

Os requisitos de software são tradicionalmente classificados entre funcionais e não funcionais, a fim de manter diferentes níveis de detalhamento para diferentes leitores.

Requisitos funcionais: Descrevem o funcionamento do sistema, e para isso devem prover como o sistema deve reagir à entrada/saída assim como seus comportamentos em diferentes situações.

Requisitos não funcionais: Devem estabelecer as restrições das funcionalidades e serviços oferecidos pelo sistema. São descritas caraterísticas gerais do sistema, como a usabilidade que não se referem a termos específicos como os requisitos funcionais. Comumente também são descritas questões que devem ser atendidas para a segurança e confiabilidade do sistema.

Para cada requisito é utilizado um código para identificar a funcionalidade, assim facilitando referenciá-la durante o desenvolvimento. Nesta seção serão apresentados os requisitos funcionais e não funcionais para o desenvolvimento deste projeto. Para a descrição das funcionalidades optou-se por usar códigos com a sintaxe [RF0X] para requisitos funcionais e [RNF0X] para requisitos não funcionais. Junto ao código e a descrição do requisito foi adicionada a sua prioridade. As prioridades são classificadas em três categorias: a) Essencial para os que precisam ser implementados indispensavelmente, ou seja, são estritamente necessários para o funcionamento do sistema; b) Importante para os que são importantes para o funcionamento, mas não são cruciais; c) Desejável para os que não interferem diretamente nas funcionalidades básicas do sistema embora relevantes, mas que podem ser deixados para ser implementados posteriormente.

Modelo de Dados

Os usuários inicialmente são estruturados como um conjunto de filmes, sendo extraídos aqueles bem avaliados, à partir de um valor de relevância $r$ baseado num modelo de cinco estrelas, onde nenhuma estrela é totalmente irrelevante e cinco totalmente relevante. Os filmes bem avaliados serão considerados aqueles com relevância $r \geq 5$, portanto tornando-se a representação de suas preferências. Contudo, desta forma ainda não é possível realizar a comparação com o modelo de filmes utilizando a métrica de similaridade proposta (ver \ref{ssec:sim\_rec}), devido a comparação ser de termo a termo. Uma opção inicial seria simplesmente calcular todos os termos dos filmes de preferência do usuário, utilizado no processo de \ac{NLP}, e uni-los num grande conjunto, porém isto tornaria o modelo de usuário muito custoso para ser utilizado, além de não escalar bem conforme as preferência do usuário aumentam. Sendo assim, optou-se por calcular os \enquote{melhores termos únicos} que representam o usuário, como sendo um conjunto de termos de um tamanho definido. Para calcular esses \enquote{melhores termos} é aplicado um modelo de frequência, o \ac{TFIDF} (ver Equação \ref{eq:tfidf}), que busca criar um ranking de termos determinando o quão importante são numa coleção, no caso a união de todos os conjuntos filmes e seus termos. Com isso é definido $Y$ como sendo o conjunto de todos os usuários $U$, e este por sua vez a união de todos os filmes $F\_u$ de sua preferência, contendo todos os termos. Esses termos são denominados de termos do usuário, representado pelo elemento $t\_u$. Assim como o modelo de filmes, o conjunto $U$ de termos do usuário também representa seus metadados. Posteriormente define-se o conjunto $M\_u$ como sendo o \textbf{modelo do usuário} dos melhores termos $t\_u \in U$, ao passarem pela seleção da função $M\_{tfidf}$, que um define um subconjunto de um tamanho definido pela constante $z$. Por fim, o conjunto $Z$ é a união de todos os modelos de usuário $M\_u$. O tamanho de $M\_u$ é algo que será explorado no capítulo \ref{cap:evaluation}. As equações \ref{eq:user\_model1} à \ref{eq:user\_model4} formalizam a construção do modelo do usuário.

Arquitetura

A arquitetura de software trata-se das estruturas e componentes, assim como as interações entre essas partes que irão compor o software do sistema. Para \cite{Perry1992} a arquitetura de software manifesta-se principalmente em partes do software do produto em relação a: 1) Requisitos para a determinação da informação, processamento e características que serão necessárias para o usuário e o sistema; 2) Arquitetura quando preocupa-se com a seleção de elementos, suas interações, e restrições necessárias para prover um \textit{framework} que satisfaça os requisitos; 3) Design quando está interessado na modularização e detalhamento do design dos elementos, algoritmos, procedimentos e tipos de dados que suportem a arquitetura e os requisitos; 4) Implementação quando preocupa-se com a representação de algoritmos, tipos de dados que satisfaçam a arquitetura, design e o requisitos.

Para a organização e estrutura deste projeto foi escolhida o padrão de arquitetura Model View Controller \ac{MVC}. O objetivo desse padrão é organizar o sistema em camadas em que cada uma seja responsável por funcionalidades específicas no fluxo entre o sistema e o usuário. Assim, o desenvolvimento e alterações podem ser realizadas de forma independente. No \ac{MVC} o sistema é estruturado em três camada que interagem entre si:

Model: Camada da representação ou modelo para a manipulação dos dados da aplicação, sendo usado tanto na manipulação de elementos da interface como na persistência de dados. Comumente para a persistência de dados modela-se cada entidade da base dados como entidades que manipularam os registros armazenados. Nessa camada também são estabelecidos elementos que implementam recursos de \textit{proxy} para os dados de fato, como os \textit{repositories}, assim facilitando o reuso da aplicação.

View: Camada da apresentação para o usuário, a interface. Envolve toda a parte de visualização e interação com o sistema do ponto de vista do usuário. Nessa camada serão apresentados os dados das \textit{models}.

Controller: Camada que controla o fluxo das informações e/ou comportamentos entre a \textit{view} e a \textit{model}. Pode receber dados da \textit{view} para ser passado à camada das \textit{models}, além de obter dados da \textit{model} para exibição na \textit{view}. Essa camada também pode acessar a serviços terceiros que manipulem dados vindos da \textit{view} ou da \textit{model}. Essa camada é também responsável pela validação dos dados vindos \textit{view} passados pelos usuários, além do controle do acesso entre outros comportamentos dos requisitos.

Na figura \ref{fig:mvc} é demonstrado o fluxo da comunicação entre as três camadas do \ac{MVC}. Iniciando pela requisição e interação do usuário através da interface na camada da \textit{view}. Nos \textit{controllers} são implementados os pontos de acesso, endereços para a camada da \textit{view}, assim como todas as regras e controles da entrada de dados, controle de acesso e utilização das \textit{models}. O \textit{controller} também poderá transformar os dados para suportar o formato de dados da \textit{model}. Por último na \textit{model} encontra-se as representações das entidades da base de dados, assim como dados possíveis dados para facilitar a manipulação dos dados da \textit{view}.

Para o desenvolvimento do sistema foram escolhidas algumas tecnologias para arquitetura software, como linguagens de programação, \texit{framework} \ac{MVC}, processamento e banco dedos, entre outras. A seguir serão apresentadas as tecnologias utilizadas.

JAVA

JAVA\footnote{https://www.java.com} é uma linguagem de programação de propósito genérico, desenvolvida originalmente por James Gosling na Sun Microsystems\footnote{ https://www.oracle.com/br/sun/index.html} em 1995. Atualmente a linguagem foi comprada pela Oracle Corporation\footnote{https://www.oracle.com}. As características em destaque da linguagem estão no fato de ser baseada em classes e orientada a objetos. A \ac{OOP} é um paradigma de programação que abstrai conceitos em objetos, que podem conter dados, campos e comportamentos nomeados de \textit{methods} \citep{Lewis2000}.

Outra caraterística importante da linguagem trata-se da filosofia apresentada pelos desenvolvedores de “escreva uma vez, rode em qualquer lugar”. A filosofia trata-se da linguagem ser compilada por uma \ac{VM} possibilitando escrever um mesmo pedaço de código que possa ser portado para outra plataforma sem necessidade de alterá-lo, uma vez que cada \ac{VM} implementa as especificidades da nova plataforma abstraindo o acesso ao \ac{SO}.

A linguagem JAVA já é usada em diversos sistemas e plataformas, com inúmeros propósitos, desde aplicações desktop, pesquisa científica, desenvolvimento web entre outros propósitos.

Spring Boot

Spring Boot\footnote{https://projects.spring.io/spring-boot/} é um projeto da Pivotal Software\footnote{https://pivotal.io} para facilitar o processo de configuração e publicação de aplicações e serviços providos pelo Spring\footnote{https://spring.io}, como baixo esforço e configuração. O \textit{Spring} é um framework \textit{open source}\footnote{Modelo de desenvolvimento que promove um licenciamento livre para o design ou esquematização de um produto} que provê um compreensivo conjunto de modelos de configuração para aplicações JAVA. O elemento principal do \textit{Spring} é prover infraestrutura para aplicações oferecendo os seguintes principais recursos:

Inversão de Controle: \ac{IOC}, também conhecido como \textit{dependency injection} é um princípio que as “dependências” devem ser supridas, injetadas por outro objeto. As dependências são objetos que serão usados como “serviços” para acessar suas funcionalidades, dentro dos \textit{containers} de \ac{IOC}. A injeção é passagem da dependência para um objeto (o cliente) \citep{DependencyInjection2006}. O termo “inversão de controle” origina-se do fato que a criação de valores de classes externas ao objeto não pode ser realizada pelo próprio objeto e sim pelos \textit{containers} de \ac{IOC}.

Acesso a dados: O framework possui diversas bibliotecas para o acesso a dados, tanto para bancos relacionais como não relacionais. Também é oferecido um sistema ORM que trata-se de uma técnica para traduzir o formato de dados de um banco relacional para \ac{OOP}, facilitando sua manipulação.

Arquitetura MVC: Fornece todo suporte para customizar e criar uma arquitetura \ac{MVC}.

HTML

O HTML\footnote{https://www.w3.org/html}, \ac{CSS}\footnote{https://www.w3.org/Style/CSS/} e JavaScript forma a principal pilha de tecnologias utilizadas na Web. O HTML é uma linguagem de marcação mantida pela \ac{W3C} para criação de páginas, originalmente desenvolvida por Tim-Berners-Lee \citep{Raggett1998}. O objetivo é a fácil construção e publicação de conteúdo no ambiente Web e consequentemente na \ac{WWW}. No \textit{Spring Boot} as páginas HTML podem ser escritas utilizando algum dos mecanismo de templates, como o \textit{thymeleaf}. Uma das vantagens porá utilização desses mecanismos é a herança de visualizações, assim como facilidade de interligar em manipular os dados passados do \textit{controller}.

O \ac{CSS} é uma linguagem para criar regras de estilização das páginas \ac{HTML}. O CSS cria ou altera um formato de apresentação (tamanho, cores, margens etc) de algum elemento do HTML, como blocos, parágrafos, imagens entre outros. Quanto ao JavaScript é uma linguagem de programação originalmente criada por Brendan Eich na Netscape Communications\footnote{ <http://isp.netscape.com>}. A linguagem é utilizada para controlar o comportamento de páginas HTML, oferecendo dinamicidade, podendo alterar elementos da página em tempo real.

MySQL

O MySQL\footnote{https://www.mysql.com} trata-se de um \ac{SGBD} que utiliza a linguagem \ac{SQL} para manipulação de dados guardados em um sistema de arquivos \citep{MySQLSGBD}. Originalmente desenvolvido por Michael Widenius em 1994, o seu foco é para o desenvolvimento de aplicações Web, embora tenha se popularizado para a maioria das plataformas existentes \citep{MySQLDevelopers}. Foi o banco de dados escolhido para a persistência de dados da aplicação e ser de fácil integração com o \textit{Spring Boot}.

Apache Jena

Apache Jena\footnote{https://jena.apache.org} é um \textit{framework} \textit{open source} para Web Semântica, escrito na linguagem Java. A biblioteca provê uma \ac{API} que facilita a extração e criação de dados nos grafos do \ac{RDF}, além de oferecer suporte para a linguagem de consulta \ac{SPARQL}. O objetivo da escolha dessa tecnologia para o projeto, é para facilitar a busca e navegação pelo grafo de entidades (\textit{resources}) no sistema da DBPedia\footnote{http://wiki.dbpedia.org} utilizando \ac{SPARQL}. Após o \ac{SR} extrair entidades das descrições do filme, essas serão buscadas no serviço da Web Semântica estendendo o conhecimento do recurso.

Apache OpenNLP

Apache OpenNLP\footnote{https://opennlp.apache.org} é um \textit{framework} de aprendizado de máquina que é usado para processamento de \ac{NLP}. A biblioteca provê uma \ac{API} com serviços para geração de \textit{tokens}, sentenças, segmentação, reconhecimento de partes da fala, extração de entidade de nome, geração de \textit{chunks} (pedaços), entre outras tarefas do \ac{NLP}. A figura \ref{fig:nlp} mostra algumas das tarefas envolvidas no processamento de linguagem natural.

No projeto essa tecnologia será utilizada para o \ac{NER} e extração de partes gramaticais presentes na descrição do filme, assim como a geração dos \textit{tokens}. O objetivo é que com essa biblioteca seja possível gerar \textit{tokens} com entidades encontradas, de nomes localizações, como também partes do texto de nomes próprios, substantivos e adjetivos.

Apache Lucene

Apache Lucene\footnote{https://lucene.apache.org} é um \textit{framework} \textit{open source} para sistemas de recuperação de informação e recomendação. O projeto oferece dois principais recursos: indexação e pesquisa de texto. Lucene é muito reconhecido por sua utilidade na implementação em mecanismos de buscas na Internet \citep{McCandless2010}. O projeto também é muito utilizado em sistemas de recomendação com implementação de diversos algoritmos para calcular a similaridade de documentos.

No projeto essa tecnologia será utilizada para tirar proveito dos algoritmos de similaridade, como o \textit{cossine similarity}\ref{eq:cosine\_sim}, podendo estender seu funcionamento e permitir integração com a biblioteca, facilitando o seu uso para outras pessoas e outros projetos.

Funcionamento

As tecnologias apresentadas anteriormente serão utilizadas para construir toda a arquitetura do sistema de recomendação. A proposta é criar uma recomendação baseada em conteúdo, e para este trabalho foi definido o domínio de filmes como exemplo de utilização. Sendo assim, o sistema possui algumas etapas de processamento para viabilizar a recomendação:

Coleta dos Filmes: Serão coletados dados dos filmes utilizando o projeto MovieLens\footnote{https://movielens.org} (ver \ref{ssec:dataModel})

Coleta das preferências do usuário: Serão coletados dados das preferências dos usuários, ou seja, os filmes de interesse. Nessa etapa poderá ser utilizada o perfil do Facebook\footnote{https://facebook.com} para obter tais dados.

Pré-processamento dos filmes: Nessa etapa após a coleta dos filmes, os dados serão previamente processados para a geração de \textit{tokens} com \ac{NLP} analisando a descrição dos itens. Os dados gerados também serão expandidos analisando as entidades extraídas no \ac{NER} da sinopse do filme, com o serviço da DBPedia\footnote{http://wiki.dbpedia.org}. Após todos os processamentos os \textit{tokens} serão persistidos no banco de dados.

Cálculo da Similaridade: Após a etapa de pré-processamento dos filmes, será realizado o cálculo da similaridade para todos os filmes da base dados utilizando uma similaridade do cosseno semanticamente estendida (ver \ref{ssec:recsysAlgo}). Os dados gerados serão persistidos no banco para viabilizar o desempenho da geração das recomendações

Geração das recomendações: Com as similaridades calculas e persistidas o sistema deverá gerar listas de filmes para recomendação (ver \ref{ssec:recsysAlgo})

Apresentação dos resultados: Por fim o sistema apresentará os resultados das recomendações para o usuário.

Cache

\begin{itemize}

\item{\textbf{Cache remoto com banco de dados}: Para calcular a similaridade entre dois tokens quaisquer, o sistema utiliza uma Equação (\ref{eq:rlws}) de similaridade que tira proveito do serviço da DBPedia. Na equação, conforme abordado em \ref{ssec:sim\_rec} realiza-se a contagem de links diretos e indiretos dos recursos. Essa contagem posteriormente é armazenada no banco de dados para uma consulta mais ágil.}

\item{\textbf{Cache local, em memória}: A cada comparação realizada é verificado inicialmente se a mesma está presente no cache remoto, evitando consultas desnecessárias ao DBPedia. Após a verificação a comparação é armazenada num cache local em memória de execução (\ac{RAM}). O objetivo é que durante o cálculo da recomendação de um usuário não seja necessário realizar a todo instante consultas ao banco, visto o volume de comparações, já que cada palavra dos termos do usuário é comparada com todas as palavras dos filmes. O processo de realizar tais consultas ao banco no momento da comparação prejudica o desempenho da aplicação. Assim, optou-se por utilizar a estrutura de \textit{cache} em memória fornecida pelo \textit{framework} Spring\footnote{https://spring.io}. Após um determinado número de comparações o sistema verifica quais delas precisam ser persistidas, salvando-as em sequência e posteriormente liberando o cache em memória.

\end{itemize}

Com esse sistema minimiza-se o tráfego de dados entre a aplicação e o banco de dados durante a comparação de filmes, acelerando o processo, além de ter um bom balanço de desempenho entre o cache remoto e cache local