

3

Framework para avaliação de sistemas de recomendação

A *framework* a ser apresentada baseia-se na experiência adquirida durante os experimentos realizados para a construção de recomendadores para redes sociais levando em conta a rede social dos usuários e o conteúdo visual de itens. Esta aplicação é descrita no capítulo 4.

Podemos dizer que ela fornece auxílio ao estudo de recomendadores para redes sociais nos seguintes aspectos:

1. na caracterização da rede social e relacionamentos entre seus conceitos;
2. na identificação e classificação das tarefas de recomendação que podem ser desempenhadas na rede social em análise;
3. na realização de experimentos com diferentes algoritmos recomendadores. Nestes experimentos, podem ser realizadas variações em seus parâmetros, dados e taxas de treinamento, além do uso de variadas métricas de semelhança entre conteúdos de itens, usuários etc.

Na seção 3.1.4 alguns exemplos de sua aplicabilidade são dados e no capítulo 4 uma instanciamento da *framework* proposta será apresentada em maiores detalhes, demonstrando o uso da mesma para a análise de uma rede social cujo tema central são fotografias contribuídas pelos usuários (serviço online *Flickr*).

3.1

Modelagem da rede social e tarefas de recomendação

3.1.1

Contribuições da web semântica

A existência de semânticas comuns (ontologias) para a descrição de redes sociais conforme apresentadas na seção 2.2.3 representam os seguintes benefícios ao processo de criação de sistemas recomendadores:

em tempo de modelagem e análise: uma vez que na construção de recomendadores há uma recorrente necessidade de identificar e formalizar

as principais entidades da rede social (contendo os itens alvos da recomendação, usuários finais e relacionamentos entre eles), é vantajoso reusar na modelagem os conceitos semânticos definidos nessas ontologias. Um exemplo de modelagem básica para tarefas de recomendação pode ser visto na figura 3.3. Tais ontologias servem também como auxílio na identificação das possíveis tarefas a serem desempenhadas pelos recomendadores, levando em consideração o contexto da rede social sendo analisada.

em tempo de execução: durante a realização das recomendações em si pode haver reuso de mecanismos e técnicas para apresentação dos resultados para o usuário final (padrões de IHM¹). Para sistemas recomendadores online podemos identificar padrões de IHM para tarefas recorrentes como: apresentar resultados, solicitar avaliação do desempenho, fornecer assistência durante navegação etc².

em tempo de treinamento e manutenção: a existência não só de uma ontologia para descrever relacionamentos entre usuários e conteúdo/mídia social mas também de padrões como o RDF/XML³ para a codificação e troca dessas informações entre os diversos sistemas produtores e consumidores de mídia social é vantajoso para sistemas recomendadores ao propiciar uma alimentação contínua de suas bases de dados para treinamento e calibração de parâmetros usados nas recomendações, o que ultimamente podem implicar em maior desempenho das mesmas. A figura 3.1 ilustra esse conceito, onde agregadores de conteúdo e mídia social (os próprios serviços de rede social ou *hubs*/motores de busca especializados na compilação dessas informações) agem como fontes para consumidores de dados descritos nessa ontologia comum (SIOC), que podem ser sistemas recomendadores ou outros serviços de mídia social onde existam benefícios no reaproveitamento e interligação de seu conteúdo com mídia social de terceiros.

3.1.2 Ontologias SIOC

Introduzida em 2005, a iniciativa *SIOC* tem como objetivo prover os principais conceitos e propriedades necessários para descrever na Web semântica as

¹sistemas recomendadores podem ser estudados sob a ótica de possíveis melhorias para a *interface humano máquina*.

²maiores discussões sobre interfaces para usuários finais em sistema recomendadores podem ser encontradas em [Herlocker et al., 2000] e [Mcnee et al., 2003].

³padrão de sintaxe para representação em XML do RDF [rdf, 2008], disponível em <http://www.w3.org/TR/rdf-syntax-grammar/>

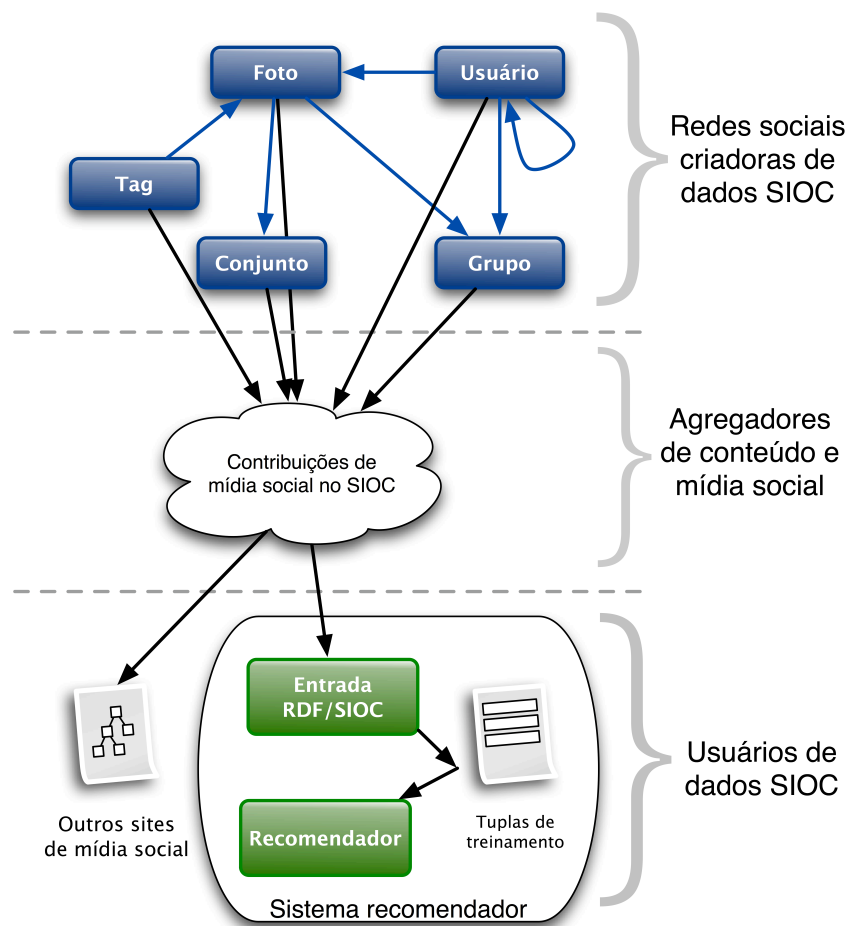


Figura 3.1: Relação entre recomendadores para redes sociais, agregadores de conteúdo social e redes sociais

informações criadas em comunidades online (fóruns de discussão, wikis, blogs, redes sociais etc).

Esta iniciativa é composta por alguns módulos, que são usados para estender os termos e relacionamentos disponíveis e assim evitar tornar a ontologia *SIOC-Core*⁴ muito complexa. Atualmente, esta iniciativa conta com três módulos além do *SIOC-Core*: *SIOC-Access*, *SIOC-Services* e *SIOC-Types*.

O módulo *SIOC-Services*⁵ pode ser usado para melhor definir os serviços de publicação de mídia social disponibilizados por repositórios dessas informações, conforme ilustrado na figura 3.1 e mencionado na seção anterior.

As principais classes da ontologia *SIOC-Core* (com destaque para os conceitos relevantes para este trabalho), seus relacionamentos e atributos são ilustrados na figura 3.2 e de forma mais completa na figura A.1.

Segue breve descrição destas classes:

⁴<http://rdfs.org/sioc/spec/>

⁵<http://rdfs.org/sioc/services>

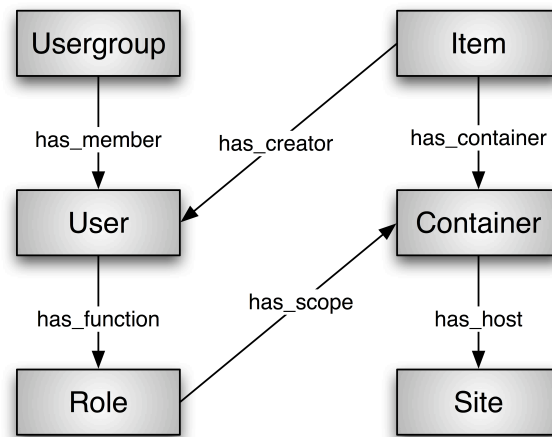


Figura 3.2: Principais classes e relacionamentos da ontologia *SIOC-Core*

SIOC:Container é uma area em que os itens de conteúdo estão contidos.

Trata-se de um conceito usado para agrupar itens de mídia social. Alguns exemplos de containers definidos no módulo SIOC-Types⁶ são: *AddressBook*, *AudioChannel*, *BookmarkFolder*, *Briefcase*, *EventCalendar*, *ImageGallery*, *ProjectDirectory*, *ResumeBank*, *ReviewArea*, *SubscriptionList*, *SurveyCollection*, *VideoChannel*, *Wiki* etc;

SIOC:Item é algo que pode estar em um *Container*. Este conceito possui subclasses que especificam tipos diferentes de itens como por exemplo **SIOC:Post**, usado para descrever artigos ou mensagens criadas em sites de comunidades, forums etc. Muitos outros exemplos de extensões à classe **SIOC:Item** são dadas no módulo *SIOC-Types*;

SIOC:Role representa a função de um usuário dentro do escopo de um forum, grupo, site etc.

SIOC:Site consiste na localização de uma comunidade online ou conjunto de comunidades, com seus usuários organizados em grupos, criando itens dentro de containers. Sistemas recomendadores online frequentemente têm seu domínio de itens recomendáveis e usuários alvos da recomendação restritos a um único *Site*. No entanto, com a disseminação crescente do uso de padrões para troca de dados sociais conforme discutidos na seção 2.2.3, o uso de relacionamentos sociais e dados de outros sites na construção e treinamento de recomendadores torna-se possível;

SIOC:User modela a conta de um usuário em uma comunidade online. Uma instância de *foaf:Person* normalmente vai possuir uma conta registrada

⁶cujá definição em RDF-XML encontra-se em <http://rdfs.org/sioc/types>. Uma representação gráfica mais completa das classes e propriedades deste módulo SIOC encontra-se na figura A.2

de usuário (*SIOC:User*) em um *SIOC:Site* (através da propriedade *foaf:holdsAccount*);

SIOC:Usergroup um conjunto de contas *SIOC:User* cujos donos possuem um interesse ou objetivo em comum.

3.1.3

Modelagem das tarefas de recomendação

Os principais conceitos e relacionamentos (relevantes para tarefas de recomendação) presentes na maioria das redes sociais online podem ser modeladas fazendo uso direto dessas classes ou através de extensões às mesmas ou de outras classes previstas nos módulos desta ontologia.

A figura 3.3 representa as principais tarefas de recomendação dentro de uma rede social modelada com o uso de classes do *SIOC-Core*. Nela a direção das setas indica quem pode ser recomendado para quem, ou seja: *itens* podem ser recomendados para *usuários*, *grupos de usuários* para *usuários* etc.

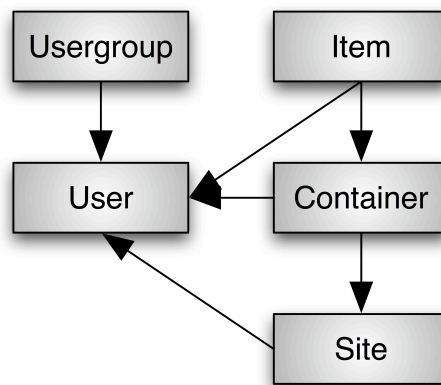


Figura 3.3: Tarefas básicas de recomendação na ontologia *SIOC-Core*

Em termos desse vocabulário, uma recomendação dentro de uma rede social online pode ser expressada como o seguinte problema:

Dada uma relação na modelagem da rede online e uma instância desse domínio, encontre uma instância candidata para o domínio para o qual essa relação deve existir

Por exemplo, “encontrar uma foto a ser submetida para um grupo”, ou “encontrar uma pessoa para ser contato de outra pessoa”.

As classes que podem ser recomendadas para usuários também se aplicam a grupos de usuários. Nestes casos, as métricas para semelhança de usuários em um recomendador baseado em usuário levariam em consideração alguma heurística para cálculo da preferência média dos membros de cada grupo.

Não é necessário estender uma classe da ontologia base quando o conceito da ontologia descreve sem ambiguidade o que existe na rede social. Conceitos básicos como `SIOC:User` e `SIOC:Usergroup` raramente precisam ser estendidos para fazer a modelagem da rede social sendo analisada. Já conceitos como `SIOC:Item` e `SIOC:Container` são bons candidatos a extensões já que numa rede social online encontramos diversos tipos de itens que se agrupam num mesmo ou em distintos tipos de containers.

Uma vez estabelecidas as possíveis recomendações nesse modelo básico, podemos construir recomendadores genéricos para essas tarefas. Como entrada, esses recomendadores genéricos usariam os relacionamentos entre esses conceitos conhecidos. Podemos então reutilizar esses recomendadores para quaisquer redes sociais que possam ter seus conceitos e tarefas de recomendação mapeadas para essa modelagem base. Nesse caso, bastaria realizar a tradução dos dados de relacionamento entre a rede de origem e os conceitos básicos compreendidos pelo motor de recomendação genérico.

Notação gráfica

A figura 3.4 sugere uma notação gráfica para a representação das possíveis tarefas de recomendação numa rede social online nesse contexto.

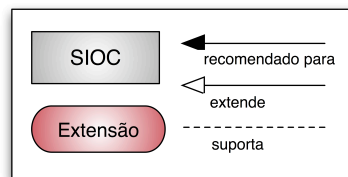


Figura 3.4: Notação gráfica para a representação de redes sociais e tarefas de recomendação

Nesta notação destacamos os seguintes elementos:

- conceitos base da ontologia SIOC (seus módulos e ontologias relacionadas), representados em caixas com ângulos retos;
- conceitos da rede social sendo modelada que estendem os conceitos base, em caixas com ângulos arredondados. Setas de linha contínua e terminação vazada apontam para o conceito base estendido;
- possíveis recomendações de uma entidade para a outra são representadas com setas de linha contínua e terminação cheia, partindo da entidade que é recomendada;
- linhas tracejadas conectam uma recomendação às entidades que podem agir como seu suporte. Fica subentendido que na recomendação da

entidade A para a B , os relacionamentos diretos entre essas entidades sempre podem agir como suporte para a recomendação, não precisando assim serem representados no diagrama.

3.1.4

Exemplos de modelagem

Apresentamos nessa seção dois exemplos de modelagem usando a *framework* proposta e no capítulo 4 fazemos o uso da *framework* para a análise de recomendadores na rede social *Flickr*.

Last.fm

A rede social online *Last.fm*⁷, tem como objetivo o compartilhamento de gostos musicais, dando a seus usuários acesso a milhares de faixas de variados gêneros.

Antes de usar o sistema, usuários devem criar uma conta e toda interação com o serviço pode ser feita através da interface web, via um tocador Flash no browser ou através de um aplicativo desktop que pode ser baixado. O perfil musical dos usuários é criado à medida em que ele ouve músicas no seu tocador de música padrão ou no tocador disponibilizado pelo *Last.fm*. Neste último caso, o usuário pode fornecer três ratings distintos para uma música recém-ouvida: **Amei**, **Pular** ou **Banir**. Estes ratings são também consideradas para realizar recomendações.

Baseado nesses perfis de gostos musicais, o serviço disponibiliza um número de funcionalidades personalizadas, como encontrar mais artistas que o usuário pode gostar, outros usuários com gostos similares (lá denominado “vizinhança” do usuário), rankings de **músicas**, **artistas** e **albums** mais ouvidos pelo usuário, **eventos** relevantes próximos de sua **localização** geográfica etc.

Os usuários podem organizar os itens musicais de seu interesse na sua **biblioteca** (que reúne todos os itens de interesse do usuário) ou criar **playlists** com algumas dezenas de músicas para compartilhar com amigos ou ouvir através do site.

Outras conexões sociais podem ser criadas no *Last.fm* através de relações de amizade: o **usuário** pode adicionar amigos através da interface web, aceitar pedidos de amizade e torná-la mútua.

Da mesma forma que com “vizinhos”, o usuário pode ouvir a estação de rádio de seus amigos. A estação de rádio de um usuário é composta das faixas preferidas dele.

⁷<http://last.fm/>

usuários com gostos e interesses em comum também podem formar **grupos**. Estatísticas musicais também são geradas para grupos da mesma forma como para usuários individuais. Qualquer usuário pode criar grupos e convidar outros usuários para fazer parte.

Outra funcionalidade do *Last.fm* bastante relevante para a criação de recomendadores é a possibilidade dos usuários criarem **tags** para descrever qualquer artista, album ou faixa em seu perfil.

Na literatura destacamos os dois seguintes trabalhos onde encontramos introduções a essa rede social online e investigações de recomendadores para ela:

O trabalho [Symeonidis et al., 2008] modela com tensores de terceira-ordem os relacionamentos entre usuários, tags (palavras-chave) e artistas. Nele são usadas técnicas de análise probabilística de semântica latente (LSA) e decomposição de valor singular de ordem maior (HOSVD) para realizar a recomendação de artistas para usuários.

Em [Firan et al., 2007], Firan, Nejdl e Paiu estudam a aplicabilidade do emprego de tags — usadas nesse site para descrever artistas e músicas — para construir o perfil de usuários ao invés da abordagem mais comum de se usar apenas a ocorrência de músicas no perfil do usuário. Neste trabalho é proposto também um algoritmo para a recomendação de músicas para usuários.

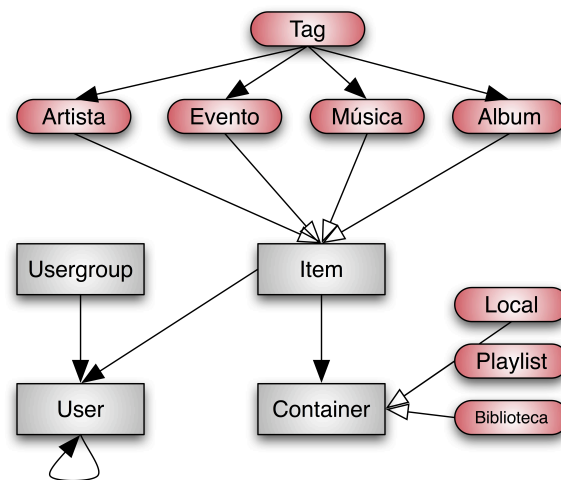


Figura 3.5: Tarefas de recomendação na rede social *Last.fm*

Representamos na figura 3.5 as principais entidades dessa rede e algumas tarefas de recomendação. Para essa representação é usada a notação introduzida na figura 3.4, da seção 3.1.3.

Nesta modelagem usamos as classes `SI0C:Usergroup` para representar comunidades de usuários do *Last.FM* e a classe `SI0C:User` para as contas de usuários. O restante dos itens de conteúdo são modelados como extensões

à classe *SIOC:Item*. **Artistas, Músicas e Albums** têm como *containers* as extensões **Playlist** e **Biblioteca**.

Destacamos as seguintes tarefas de recomendação:

Tags para Artista, Evento, Música e Album: já que tags são usadas para descrever essas entidades, essa recomendação fornece uma ajuda — sob forma de sugestões — para o usuário no momento em que ele está anotando itens. O uso de tags para anotar itens serve como auxílio futuro aos usuários na tarefa de recuperar itens;

Artistas, Músicas, Albums e Eventos para um Usuário: baseado no perfil de gosto construído para um usuário ou sua localização geográfica, o sistema pode realizar essas recomendações sugerindo itens musicais que o usuário ainda não tem conhecimento;

Eventos para um Local: teria utilidade para organizadores de eventos, podendo ajudar a responder perguntas como: baseado nos gostos musicais dos usuários de cada localização, qual o melhor local para eventos de um determinado artista?

Grupos de usuários para usuários: baseado em gostos comuns dos usuários de um grupo pode-se construir um perfil “médio” para o grupo e usá-lo na recomendação para usuários.

Orkut

A rede social online Orkut⁸ tem como foco relacionamentos reais entre os usuários, que podem declarar sua relação de amizade com outro usuário em diferentes graus (amigo, parente, conhecido, trabalho...). Um usuário pode entrar diversas informações pessoais sob forma de texto livre para descrever diversos aspectos de seus gostos, o que vai compor seu perfil nesse sistema: culinária, músicas e filmes preferidos, orientação política, localização, idade, etc.

Usuários podem também criar e fazer parte de comunidades, que reúnem usuários de interesse comum. Os donos de comunidades podem escolher comunidades relacionadas para serem apresentadas na página de sua comunidade.

Recentemente essa rede acrescentou a possibilidade de integrar ao perfil dos usuários aplicativos desenvolvidos por terceiros que fazem algum uso da rede social do usuário que opta por acrescentar à sua página algum aplicativo.

Podemos modelar o relacionamento entre as principais entidades da rede Orkut usando as duas classes básicas da ontologia *SIOC-Core*: **Usergroup**,

⁸<http://orkut.com>

3.2

Realização de experimentos

Durante o estudo de algoritmos recomendadores é necessária a realização de diversas atividades repetitivas, como validar múltiplos subconjuntos da massa de dados de treinamento, repetir o mesmo experimento diversas vezes para aumentar a confiabilidade de resultados obtidos a partir de experimentos envolvendo fatores aleatórios, representar graficamente os resultados obtidos de variadas formas etc.

Torna-se necessário então a existência de uma infra-estrutura em termos de ferramentas, códigos reaproveitáveis, procedimentos e padrões que podem ser representados através de uma *framework*, que será proposta nessa seção.

A *framework* baseia-se na experiência adquirida durante os experimentos realizados para a construção de recomendadores para redes sociais, descritos no capítulo 4.

A figura 3.7 mostra o fluxo de dados entre as principais entidades que fazem parte da *framework* proposta. Na descrição destas entidades a seguir destacamos também os principais pontos de extensão (*hotspots*).

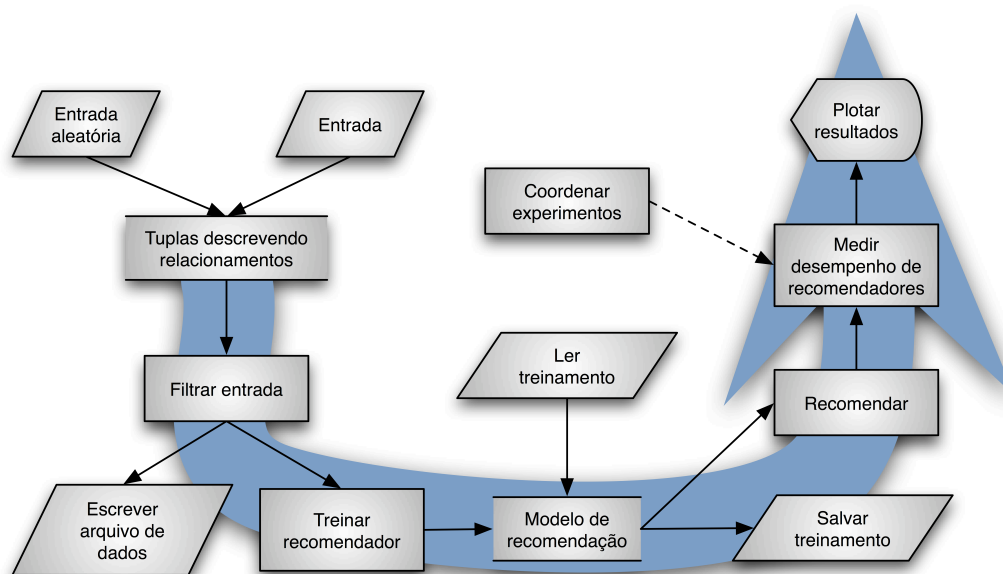


Figura 3.7: Fluxo de informações entre os componentes da *framework* proposta

Entrada Responsável por carregar dados sobre relações existentes entre as entidades de uma rede social. Tal carregamento pode ter como origem arquivos em texto puro (CSV), YAML, XML ou bases de dados relacionais. Conforme discutido na seção 2.2.3, uma extensão da interface de entrada de dados pode realizar a leitura de dados utilizando padrões

abertos para descrição de mídia social. Tal consulta a uma base de dados RDF pode ser realizada usando a linguagem SPARQL para seleção de itens. Vemos a seguir exemplos de consultas usando dados semânticos integrados à ontologia SIOC e FOAF.

Listagem 3.1: Exemplo de query SPARQL para recuperar itens de um usuário

```

1 SELECT ?user ?item
2 WHERE {
3     <uri> foaf:holdsAccount ?user .
4     ?item sioc:has_creator ?user .
5 }
```

Na listagem 3.2 o objeto *results* na linha 22 pode ser iterado para se obter identificadores de todos os artigos num blog contribuídos pelo usuário passado.

Listagem 3.2: Exemplo de uso da API Jena para execução de queries SPARQL

```

1 // Abre o grafo RDF a partir de arquivo local
2 InputStream in = new FileInputStream(new File("flickr-data.rdf"));
3
4 // Cria modelo em memória vazio e popula com o grafo
5 Model model = ModelFactory.createMemModelMaker().createModel();
6 model.read(in, null);
7 in.close();
8
9 // Cria uma nova query SPARQL
10 String queryString =
11     "PREFIX foaf: <http://xmlns.com/foaf/0.1/>" +
12     "SELECT ?url" +
13     "WHERE {" +
14     "    ?contributor foaf:name \"Jon Foobar\"." +
15     "    ?contributor foaf:weblog ?url." +
16     "}" +
17
18 Query query = QueryFactory.create(queryString);
19
20 // Executa a query e obtém os resultados
21 QueryExecution qe = QueryExecutionFactory.create(query, model);
22 ResultSet results = qe.execSelect();
23
```

```

24 // Libera recursos
25 qe.close();

```

Como parte integrante de tal entrada de dados (tendo como fonte repositórios semânticos) teríamos por exemplo aplicações da linguagem SPARQL para realizar a tradução de dados descrito num formato de grafo para outro. Tais traduções podem ser realizadas usando o conceito *SPARQL Constructs*, conforme exemplo 3.3. Neste exemplo são traduzidos para relacionamentos da ontologia SIOC os relacionamentos entre *criador de um post* e o post em si descritos originalmente em uma ontologia qualquer.

Listagem 3.3: Exemplo de *construct* SPARQL para traduzir triplas RDF de uma ontologia para outra

```

1 CONSTRUCT { ?x a sioc:Post . ?x sioc:has_creator ?y }
2 WHERE { ?x a myont:BlogElement . ?x myont:created_by ?y }

```

Um outro exemplo para extensão à entrada de dados seria uma implementação que fornece relações aleatórias entre as entidades de uma rede social ou atributos dessas entidades.

Filtrar entrada: Aplica um filtro sobre a entrada de dados, permitindo realizar projeções (reduzir dimensões), sumarizar, completar dados ausentes com médias, acrescentar ruído, normalizar, converter inteiros e decimais em strings, converter identificadores em inteiros consecutivos (necessário em algumas bibliotecas de recomendadores para melhor desempenho) etc. Transformações mais complexas sobre os dados de entrada podem ser realizadas através da composição de filtros.

Escrever arquivo de dados: Persiste dados filtrados ou resultados de experimentos em disco para posterior processamento, plotagem etc.

Medir desempenho de recomendadores: A partir dos dados de entrada e do resultado de recomendações lidas, realiza o cálculo de medidas de desempenho como *recall*, *precision*, *F-measure*, erro absoluto médio (MAE), erro quadrático médio (MSE) etc.

Treinar recomendador: Estabelece uma interface base para o treinamento de recomendadores e acompanhamento do progresso da aprendizagem. O treinamento pode ser realizado usando técnicas como *repeated hold-out* [Blum et al., 1999].

Plotar resultados: Alimentado com medidas de desempenho de treinamento (tempo gasto), taxa de dados usada no treinamento e dos resultados medidos para experimentos feitos com recomendadores, plotam gráficos simples, histogramas ou gráficos de espalhamento para análise multi-variada.

Salvar e ler treinamento: Persiste em disco e realiza a leitura do modelo de treinamento para recomendadores. Oferece poucas possibilidades para padronização de interfaces pois há grande variação nos formatos usados para persistência do conhecimento adquirido com os treinamentos realizados pelos diferentes recomendadores.

Recomendar: Retorna itens recomendados a partir de modelo previamente aprendido, tendo como entrada através de interface padronizada o contexto dos usuários e demais itens relevantes para a recomendação.

Log: O progresso de experimentos e resultados pode ser registrado em mensagens persistidas em arquivo texto ou apenas exibidas no console. Este bloco faz parte da infra-estrutura comum a toda a *framework*.