



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Considerando o feedback do usuário para melhoria dos resultados de consultas

Trabalho de Graduação

Aluno: Thiago Henrique Fernandes da Paz (thfp@cin.ufpe.br)

Orientadora: Ana Carolina Salgado (acs@cin.ufpe.br)

Dezembro 2010

Universidade Federal de Pernambuco

Centro de Informática

Graduação em Engenharia da Computação

2010.2

Thiago Henrique Fernandes da Paz

Considerando o feedback do usuário para melhoria dos resultados de consultas

Trabalho apresentado à graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco para obtenção do grau de Bacharel em Engenharia da Computação

Orientadora: Ana Carolina Salgado (acs@cin.ufpe.br)

Dezembro 2010

Agradecimentos

Agradeço primeiramente a Deus por ter me dado forças para conseguir atingir este objetivo. Agradeço também a toda minha família, principalmente meus pais, por ter me incentivado sempre em toda minha vida.

Agradeço também a minha orientadora, a professora Ana Carolina Salgado, pela oportunidade e pela grande ajuda no desenvolvimento deste trabalho. Agradeço aos meus amigos Andrêza Leite e Thiago Pachêco, pela grande ajuda e paciência que tiveram durante o desenvolvimento do projeto.

Um agradecimento especial a Isis, por ter suportado todas as dificuldades junto comigo durante todos esses anos na graduação. Agradeço também a todos os meus amigos do curso, que sofreram e riram em todos os momentos que passamos na graduação.

Muito obrigado!

Resumo

Com o aumento da quantidade de dados existente em fontes distribuídas e heterogêneas, a busca por informações relevantes tem se tornado cada vez mais difícil. Por conta desta necessidade, surgiram os sistemas PDMS (*Peer Data Management System*), que têm como objetivo integrar esses dados que se encontram espalhados por essas fontes de dados que estão distribuídas em uma plataforma *Peer-to-Peer* (P2P).

O sistema SPEED (*Semantic Peer-to-peer Data Management System*) é um PDMS que utiliza informações semânticas para melhor integrar os dados. Não há um esquema global que represente todos os pontos da rede. Ao invés disso, cada ponto possui seu próprio esquema de dados, e são gerados mapeamentos entre os esquemas dos pontos. Esses mapeamentos são usados na reformulação de consultas, para que possam ser obtidos resultados que sejam semanticamente semelhantes. Porém, resultados considerados irrelevantes pelos usuários podem ser retornados como resposta da consulta. Desta forma, o sistema deve possuir alguma forma de aprendizado, que se baseie em informações passadas pelos usuários do sistema.

Este trabalho visa o desenvolvimento do módulo de aquisição de *feedback* dos usuários do sistema SPEED. Desta forma, pode-se utilizar tais informações no refinamento dos resultados retornados pela consulta. Assim, os usuários poderão obter melhores resultados, uma vez que serão consideradas informações realmente relevantes do domínio do usuário.

Sumário

1. Introdução	7
1.1 Motivação	7
1.2 Objetivos	8
1.3 Estrutura do documento.....	8
2. Feedback do usuário para consultas	10
2.1 Feedback de usuários.....	10
2.2 Sistemas com mecanismos de feedback.....	11
2.2.1 SearchWiki.....	11
2.2.2 K-Space Image Retrieval System	12
2.3 Considerações	13
3. Reformulação de consultas no SPEED.....	14
3.1 SPEED.....	14
3.2 Reformulação de consultas.....	16
3.3 Considerações	18
4. Módulo de feedback dos usuários no SPEED.....	19
4.1 Especificação	19
4.1.1 Descrição técnica	21
4.1.2 Casos de Uso.....	21
4.2 Implementação.....	23
4.3 Considerações	27
5. Conclusão	28
5.1 Considerações finais	28
5.2 Contribuições.....	28
5.3 Trabalhos Futuros.....	29
Referências bibliográficas	30

Lista de figuras

Figura 1. Arquitetura do SPEED

Figura 2. Ontologia de domínio

Figura 3. Visão geral da arquitetura do módulo de feedback

Figura 4. Casos de uso do módulo de feedback

Figura 5. Tela de exibição dos resultados

Figura 6. Interface de envio de feedback

1. Introdução

Neste capítulo será feita uma breve introdução do contexto em que foi desenvolvido este trabalho. Nele, serão mostradas as necessidades atuais do sistema SPEED, além da motivação à implementação deste trabalho.

1.1 Motivação

Devido à grande quantidade de informação existente em diversas fontes de dados nos sistemas de informação atuais, vários estudos têm sido feitos na tentativa de resolver problemas de gerenciamento de dados, como, por exemplo, o processamento de consultas. Sistemas *peer-to-peer* são um bom exemplo de sistemas em que há uma grande dificuldade de se encontrar informação relevante dentre todas as informações existentes.

Uma das tentativas de se contornar este problema foi o desenvolvimento dos *Peer Data Management System* (PDMS) [Sung 2005], em que o acesso às fontes de dados, geralmente heterogêneas, é feita de forma transparente, sem a necessidade de um controle centralizado. Quando uma consulta é submetida à um determinado *peer*, é feita uma reformulação na consulta para que outros *peers* da rede possam processá-la. A reformulação das consultas é feita baseada nos mapeamentos existentes entre cada *peer* na rede.

Devido à grande variedade na representação dos dados e na semântica dos *peers*, foi sugerido um novo modelo de PDMS, baseado em ontologias [Xiao 2006]. Desta forma, a representação dos dados no sistema é feita de uma forma mais uniforme. Esse modelo de sistema é chamado de OPDMS (*Ontology-based Peer Data Management System*).

O sistema SPEED (*Semantic Peer-to-peer Data Management System*) é um OPDMS baseado em semântica que provê a integração de dados dos pontos conectados à rede. Como não há um esquema global único que represente todos os pontos da rede, a geração dos mapeamentos semânticos entre os esquemas dos pontos que compõem a rede é uma tarefa essencial.

Com os mapeamentos, o usuário poderá obter resultados semanticamente semelhantes que estejam distribuídos pelos outros pontos da rede.

Porém, resultados irrelevantes em relação ao domínio do usuário podem ser retornados pela consulta. Esses resultados equivocados podem ter sido gerados devido a alguma incoerência existente nos mapeamentos semânticos entre os pontos. O *feedback* do usuário é uma importante fonte de conhecimento para validar as informações usadas na busca do resultado [Elsas 2009]. O armazenamento dessa informação pode ser útil para futuro reuso com usuários que tenham os mesmos interesses de usuários anteriores.

1.2 Objetivos

Dado o contexto mostrado na seção anterior, o objetivo do trabalho apresentado é o desenvolvimento do mecanismo de aquisição do *feedback* dos usuários no módulo de consultas do SPEED. Este novo módulo deve ser capaz de armazenar quais os resultados que foram considerados irrelevantes pelo usuário e, desta forma, identificar possíveis erros nas correspondências semânticas utilizadas para gerar os resultados.

1.3 Estrutura do documento

O restante do trabalho será organizado do seguinte modo:

- **Capítulo 2 - *Feedback* do usuário para consultas:** Descreve os conceitos relacionados aos sistemas de *feedback* de usuários, além de apresentar exemplos de sistemas que possuem algum mecanismo de aquisição de *feedback*;
- **Capítulo 3 - Reformulação de consultas no SPEED:** Descreve o sistema SPEED, apresentando sua arquitetura e modo de funcionamento. Além disso, será mostrado o funcionamento do módulo de consultas e a geração dos mapeamentos semânticos entre os pontos da rede.
- **Capítulo 4 - Módulo de *feedback* dos usuários no SPEED:** Mostra os detalhes da especificação e implementação do módulo de aquisição de *feedback* dos usuários.

- **Capítulo 5 - Conclusão:** Descreve as conclusões e os conhecimentos adquiridos no desenvolvimento do trabalho. Também são mostradas algumas sugestões de trabalhos futuros que podem ser realizados com relação à ferramenta implementada.

2. Feedback do usuário para consultas

Neste capítulo serão discutidos os conceitos aplicados em relação ao *feedback* de usuários em sistemas, mais especificamente a sua utilização em submissão de consultas, além de apresentar o que está sendo desenvolvido na área.

2.1 Feedback de usuários

Atualmente, com a crescente quantidade de informação existente na Web, as consultas retornam um grande número de resultados. Muitos destes resultados são considerados desnecessários e irrelevantes pelos usuários. Desta forma, vários sistemas possuem algum mecanismo de aprendizagem com base nas ações dos usuários para que os resultados das consultas possam ser refinados [Zhao et al. 2009].

Os métodos de aquisição do *feedback* dos usuários podem ser divididos em duas categorias:

- Implícito: realizado pelo sistema, baseando-se em informações adquiridas indiretamente dos usuários, como histórico de navegação, cliques, entre outros;
- Explícito: realizado pelos usuários, baseando-se em informações diretas do usuário, como a indicação da relevância de um resultado, por exemplo;

Nos sistemas que possuem algum mecanismo de *feedback* explícito, a indicação da relevância dos resultados pode ser *binária*, quando apenas é dito se o resultado é relevante ou não, ou em *escala*, quando existem alguns níveis (mais que dois) de relevância possíveis a serem escolhidos [Kekalainen 2005]. Essa escala pode ser realizada usando números (de 1 à 5, por exemplo), letras ou com descrições (“irrelevante”, “pouco relevante”, “relevante”, “muito relevante”, por exemplo).

Por outro lado, sistemas que utilizam um mecanismo de *feedback* implícito geralmente usam informações guardadas nos *logs* do sistema. Os

logs podem conter um histórico das visualizações e consultas dos usuários, o tempo gasto visualizando a página, perfil do usuário, entre outros. Song propõe ainda o uso das informações sobre quais URLs foram clicadas e quais foram ignoradas, gerando assim relações entre as consultas realizadas e as URLs dos resultados [Song et al. 2010].

Uma grande questão ainda em aberto é a determinação da melhor estratégia de *feedback* a ser adotada nos sistemas. Existem vários estudos que tentam mostrar as vantagens de se utilizar um determinado tipo de estratégia, mas ainda não há um consenso. Porém, os sistemas com *feedback* implícito são considerados mais complexos do que os com *feedback* explícito, devido às variáveis que devem ser levadas em consideração, que geralmente não são muito claras [White et al. 2005].

A seguir serão descritos alguns sistemas que possuem um método de aquisição de *feedback* dos usuários.

2.2 Sistemas com mecanismos de feedback

Nesta seção serão mostrados alguns sistemas que possuem alguma forma de aquisição de *feedback* dos usuários. Os dois sistemas mostrados nesta seção são sistemas que processam consultas e retornam os resultados ao usuário. Além disso, os dois mecanismos de aquisição desses sistemas são feitos de forma explícita.

2.2.1 SearchWiki

Um bom exemplo de sistema que utiliza *feedback* dos usuários é o Google. Em 2008, a empresa criou o SearchWiki, uma funcionalidade que permitia que os usuários pudessem indicar quais os resultados eram satisfatórios e quais não eram [Happel 2009]. Além disso, era possível reordenar os resultados da consulta, e desta forma, criar uma espécie de *ranking* dos resultados. Para isso, existiam pequenos botões ao lado de cada resultado retornado pela consulta, onde se podia indicar a relevância ou não do resultado, além de inserir pequenos comentários em relação ao resultado em questão. Essas informações eram então utilizadas em outras consultas, para que os resultados que os usuários achassem mais relevantes viessem primeiro.

Uma outra funcionalidade do SearchWiki era a possibilidade de se visualizar as informações enviadas pelos outros usuários, como os comentários, por exemplo. Assim, os outros usuários poderiam ter uma noção do que se tratava um determinado resultado, sem ter que necessariamente visualizar a página retornada.

O SearchWiki foi descontinuado no início de 2010 e em seu lugar foi criado o Google Stars. Com essa nova funcionalidade, não é mais possível reordenar os resultados e inserir comentários sobre eles, apenas marcar com uma estrela aqueles resultados mais satisfatórios, e desta forma, estes resultados aparecem primeiro nos resultados da consulta.

2.2.2 K-Space Image Retrieval System

Além do SearchWiki e do Google Stars, vários outros sistemas também possuem um mecanismo para receber o *feedback* dos usuários, dentre eles o *K-Space Image Retrieval System* [Chandramouli 2008], um sistema de busca de imagens.

Esse sistema utiliza um módulo de *feedback* dos usuários para refinar os resultados da busca por imagens na web. O sistema possui um módulo de análise visual, que cria clusters de imagens baseadas nas características das imagens, além de um módulo de refinamento de consultas, que aplica algumas informações contextuais sobre as imagens.

Quando uma consulta é enviada ao sistema, imagens que não são relevantes podem ser retornadas como resultado ao usuário, talvez por terem sido agrupadas de forma equivocada mesmo com a utilização do contexto na busca. O usuário pode informar ao sistema quais as imagens corretas ou não, marcando-as como “relevante” ou “irrelevante”.

O módulo de *feedback* utiliza algoritmos de aprendizagem supervisionada em uma rede SOM (*Self Organizing Maps*). Primeiramente, o sistema cria um perfil dos usuários e armazena o histórico das buscas no sistema, além das marcações de “relevante” ou “irrelevante” feitas pelos usuários. Até o sistema armazenar informações suficientes de *feedback*, o sistema opera no modo de *clusterização*, agrupando imagens sem a utilização imediata do *feedback*. Quando a quantidade de informações salvas for maior

que um determinado valor limite, o sistema passa a operar no modo de classificação.

No modo de classificação, as informações armazenadas dos usuários são utilizadas no algoritmo de classificação supervisionada das imagens, sem a utilização do algoritmo de *clusterização*. Nesta fase, existe um limite de tempo (em meses) em que o sistema irá operar no modo de classificação. Após o sistema passar desse limite, o sistema volta a operar no modo de *clusterização* e continua armazenando as informações do *feedback* dos usuários.

2.3 Considerações

Este capítulo abordou as principais características dos métodos de aquisição do *feedback* de usuários. Foram discutidas as duas principais abordagens que podem ser utilizadas, implícita ou explicitamente. Também foram mostrados alguns sistemas que utilizam algum mecanismo de *feedback*, e como as informações adquiridas são usadas pelos sistemas.

3. Reformulação de consultas no SPEED

Este capítulo tem como objetivo descrever o módulo de reformulação de consultas do sistema SPEED (*Semantic Peer Data Management System*). Primeiramente, serão descritos as características do SPEED e de sua arquitetura, e então, será detalhado o funcionamento do seu módulo de consultas.

3.1 SPEED

O sistema SPEED é um PDMS (*Peer Data Management System*) baseado em semântica, em que os *peers* têm seus esquemas de dados representados através de ontologias. Sistemas PDMS são aplicações *peer-to-peer* que se caracterizam por proverem um ambiente de integração de dados de fontes heterogêneas e autônomas. Assim, não há necessidade de se ter um esquema global único que descreva toda a rede, mas sim uma coleção de mapeamentos semânticos entre os pontos que fazem parte da rede. A Figura 1 mostra a arquitetura de SPEED.

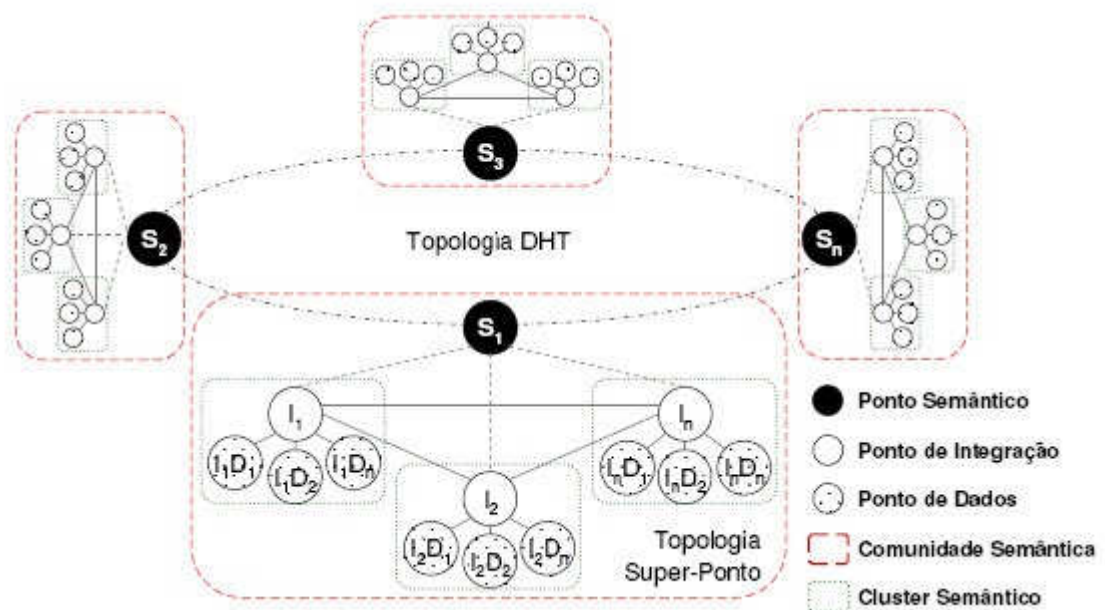


Figura 1. Arquitetura do SPEED

Duas topologias de rede são usadas no SPEED: a DHT (*Distributed Hash Table*), que organiza os *peers* semânticos, e a *super-peer*, que gerencia os *peers* de integração e de dados, e liga os *clusters* semânticos. A rede DHT é responsável pela organização dos *peers* na rede, dado suas informações semânticas, através de uma função *hash*. Nas comunidades semânticas, os *peers* são organizados com a topologia *super-peer* [Pires 2007].

No SPEED existem três diferentes tipos de *peers*: os pontos de dados, os pontos de integração e os pontos semânticos. Os pontos de dados são os *peers* que compartilham os dados, tanto estruturados como semi-estruturados, com os outros pontos da rede. Eles são agrupados nos chamados *clusters* semânticos, que são conjuntos de pontos de dados que possuem dados semanticamente semelhantes. São nos pontos de dados que as consultas submetidas são executadas. Na Figura 1, I_1D_1 , I_1D_2 e I_1D_n são exemplos de pontos de dados.

Os pontos de dados de um *cluster* semântico estão ligados a um ponto de integração. Ele é responsável pelo processamento das consultas e pelo repasse delas aos pontos de dados. Para tanto, esse *peer* deve ter um poder computacional maior do que os pontos de dados. Além disso, é necessário que este ponto tenha conhecimento dos dados de todos os pontos ligados a ele. Desta forma, cada ponto de integração possui uma ontologia que representa o seu *cluster*, gerada através de uma integração das ontologias dos pontos de dados. Os pontos I_1 , I_2 e I_n na Figura 1 são exemplos de pontos de integração no SPEED.

Os pontos de integração também estão ligados aos pontos semânticos. Esse tipo de pontos contém uma ontologia que representa os *clusters* semânticos a ele ligados, criando assim uma comunidade semântica, conjunto de *clusters* com dados semanticamente semelhantes. Os pontos S_1 , S_2 e S_3 na Figura 1 são exemplos de pontos semânticos.

Assim que um novo ponto de dados se conecta ao sistema SPEED, é necessário encontrar a comunidade semântica a que pertence esse novo *peer*. Isso é realizado através da análise das palavras-chaves dos domínios existentes.

Após isso, é necessário se determinar o *cluster* semântico que o novo *peer* deve pertencer. Essa etapa é realizada através de um *matcher* semântico, que relaciona a ontologia do ponto com as ontologias dos clusters. O *matcher* irá gerar uma medida do grau de similaridade entre as ontologias em questão. Desta forma, pode-se identificar a qual *cluster* irá pertencer o ponto. As informações desse novo ponto devem ser inseridas na ontologia que representa o *cluster* por meio de uma operação de *merging*.

Um valor limite pode ser usado para se descartar *clusters* que possuam pouca ou nenhuma similaridade com o *peer*. Caso nenhum *cluster* na rede apresente um valor de similaridade maior que o limite, um novo *cluster* será gerado, contendo o novo ponto adicionado a rede, e a sua ontologia será apenas a ontologia do ponto.

3.2 Reformulação de consultas

Um dos principais serviços oferecidos pelo SPEED é o processamento de consultas. Assim que uma consulta é submetida num determinado ponto de dados, a consulta vai até o ponto de integração do cluster a que pertence o *peer*. O ponto de integração é responsável por identificar quais pontos de dados podem responder à consulta, além de enviar a consulta aos outros pontos de integração que estão conectados a ele. Os pontos que podem responder à consulta são chamados de pontos relevantes.

Para que a consulta seja executada em todos os pontos relevantes, é necessário que a consulta original seja adaptada aos esquemas dos pontos de dados. O ponto de integração utiliza correspondências semânticas entre os pontos, tanto entre os pontos de integração, como também entre os pontos de dados e o ponto de integração. As correspondências semânticas são relações identificadas entre os conceitos e propriedades dos esquemas dos pontos. Essa reformulação da consulta não deve modificar a semântica original quando propagada aos outros pontos da comunidade. Após a realização da consulta, os resultados obtidos de todos os pontos relevantes são integrados no ponto de integração e retornados ao usuário.

Para a geração das correspondências semânticas entre os conceitos, são utilizadas as chamadas ontologias de domínio. Essas ontologias contêm

conceitos e propriedades de um domínio de conhecimento em particular [Souza 2009]. A Figura 2 mostra um exemplo de uma ontologia de domínio. Desta forma, duas ontologias quaisquer passam por uma operação de *matching* semântico, e têm seus conceitos e propriedades comparadas com os conceitos e propriedades equivalentes presentes na ontologia de domínio. Assim, a correspondência semântica entre as duas ontologias pode ser inferida baseada nos relacionamentos semânticos presentes na ontologia de domínio. Percebe-se então, que as ontologias de domínio têm um papel fundamental na geração das correspondências semânticas no sistema SPEED.

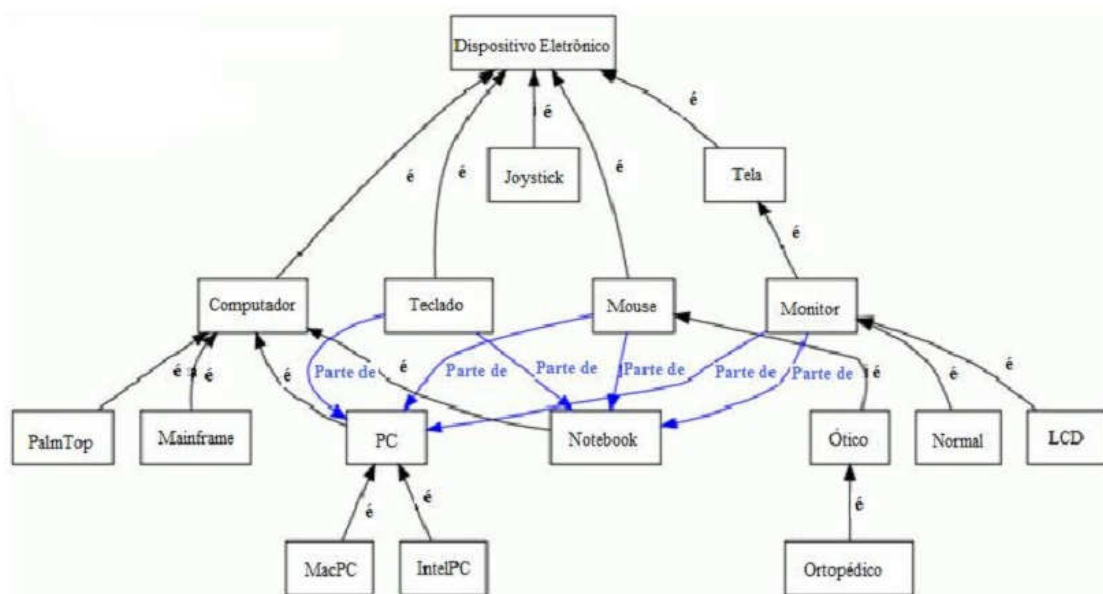


Figura 2. Ontologia de domínio

Na geração das correspondências semânticas no SPEED, são utilizados sete tipos diferentes de correspondência entre os conceitos:

- Equivalência (*isEquivalentTo*)
- Disjunção (*DisjointWith*)
- Aproximação (*IsCloseTo*)
- Subconceito (*isSubconceptOf*)
- Superconceito (*isSuperconceptOf*)
- “Parte de” (*isPartOf*)
- “Conjunto de” (*isWholeOf*)

Essas correspondências semânticas são, então, utilizadas no enriquecimento das consultas, no momento da reformulação de um ponto para outro de acordo com seus esquemas [Neves 2010]. Com esse enriquecimento, mais resultados significativos para os usuários poderão ser retornados.

Porém, alguns resultados irrelevantes poderão ser retornados após o processo de enriquecimento da consulta, possivelmente por haver inconsistências nos mapeamentos semânticos encontrados. Desta forma, o sistema deve ter alguma maneira de identificar e armazenar quais correspondências foram consideradas equivocadas pelos usuários, para que estas possam ser modificadas.

O módulo de aquisição de *feedback* dos usuários do sistema SPEED tem como objetivo adquirir dos usuários quais resultados foram considerados irrelevantes, para que assim, as correspondências utilizadas para a geração do resultado possam ser atualizadas. Este módulo (sua especificação e implementação) será detalhado no próximo capítulo.

3.3 Considerações

Neste capítulo foi apresentado o funcionamento e a arquitetura do SPEED. Também foi abordado o funcionamento do módulo de reformulação de consultas do sistema e como é a utilização da semântica para se realizar os mapeamentos entre os conceitos.

4. Módulo de feedback dos usuários no SPEED

Neste capítulo, será descrito o módulo de aquisição de *feedback* do sistema SPEED. Primeiramente, serão mostradas as características e os principais objetivos do módulo, através da especificação. Após isso, serão mostrados detalhes sobre a sua implementação.

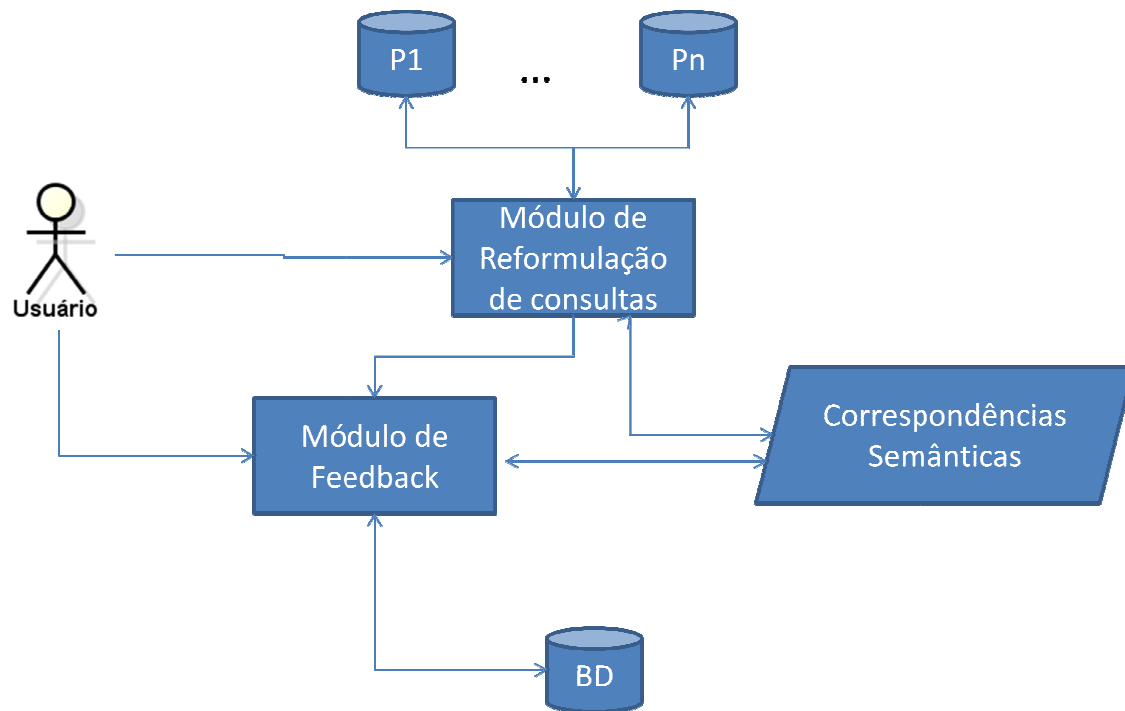
4.1 Especificação

O sistema SPEED tem como uma de suas principais características o processamento de consultas. Como já dito, as consultas submetidas a um determinado *peer* da rede é enviada ao ponto de integração conectado a ele. O ponto de integração é responsável pela reformulação das consultas, para que estas possam ser enviadas e respondidas pelos outros *peers*. A reformulação das consultas é feita utilizando as correspondências semânticas existentes entre os *peers*.

Caso alguma dessas correspondências seja mapeada equivocadamente, alguns resultados indesejados ou irrelevantes poderão ser retornados. O sistema deve, então, ter alguma forma de refinar os resultados através da aprendizagem *online* e supervisionada, ou seja, o sistema deve se ajustar de acordo com as informações enviadas pelos usuários com o uso do sistema.

Assim, o módulo de aquisição do *feedback* dos usuários visa atender às necessidades identificadas no SPEED. Este novo módulo do sistema SPEED tem por objetivo o armazenamento das informações (ontologias usadas no mapeamento, tipo do relacionamento entre eles, dentre outros) sobre resultados considerados irrelevantes pelo usuário, para assim prover um refinamento dos resultados apresentados para aquela consulta. O módulo irá se integrar ao módulo de reformulação de consultas já existente do sistema SPEED [Neves 2010].

Uma visão geral sobre a arquitetura do módulo de *feedback* integrado ao módulo de reformulação de consulta é mostrada na Figura 3.



feedback

O objetivo é que os usuários possam informar ao sistema quais resultados foram considerados irrelevantes no seu entendimento. Para isso, os resultados das consultas são obtidos a partir do módulo de reformulação de consultas e enviados ao módulo de *feedback*. Os resultados são exibidos para o usuário, e se alguns deles não o satisfizerem, ele pode simplesmente marcá-los como indesejados e o sistema irá armazenar essas informações para futuras melhorias nos mapeamentos semânticos.

Para isso, algumas funcionalidades foram identificadas:

- Aquisição de informações sobre as consultas enviadas pelo módulo de consultas;
- Exibição dos resultados retornados pela consulta;
- Identificação dos resultados irrelevantes retornados;
- Armazenamento das informações adquiridas.

4.1.1 Descrição técnica

As funcionalidades do módulo de *feedback* são descritas de forma mais técnica:

- Entrada 1: Informações sobre os resultados retornados pelo módulo de reformulação de consultas
- Entrada 2: Identificação dos resultados considerados indesejados pelos usuários
- Saída 1: Exibição dos resultados da consulta, com opção para que os usuários marquem os indesejados
- Saída 2: Dados a serem armazenados no banco de dados.
- Requisitos funcionais:
 - Receber as informações sobre os resultados gerados pela consulta
 - Apresentar os resultados da consulta para o usuário
 - Prover interface de identificação dos resultados irrelevantes
 - Enviar dados para serem armazenados no banco de dados
- Requisitos não-funcionais
 - Executar no ambiente do sistema SPEED
 - Ser independente de plataforma
 - Ter interface amigável
- Usuário do módulo: Qualquer usuário que faça uma consulta no sistema SPEED e que deseje informar sua opinião sobre os resultados.

4.1.2 Casos de Uso

O diagrama apresentado na Figura 4 mostra os casos de uso para o módulo de *feedback* do sistema SPEED, e logo em seguida os mesmos serão descritos detalhadamente.

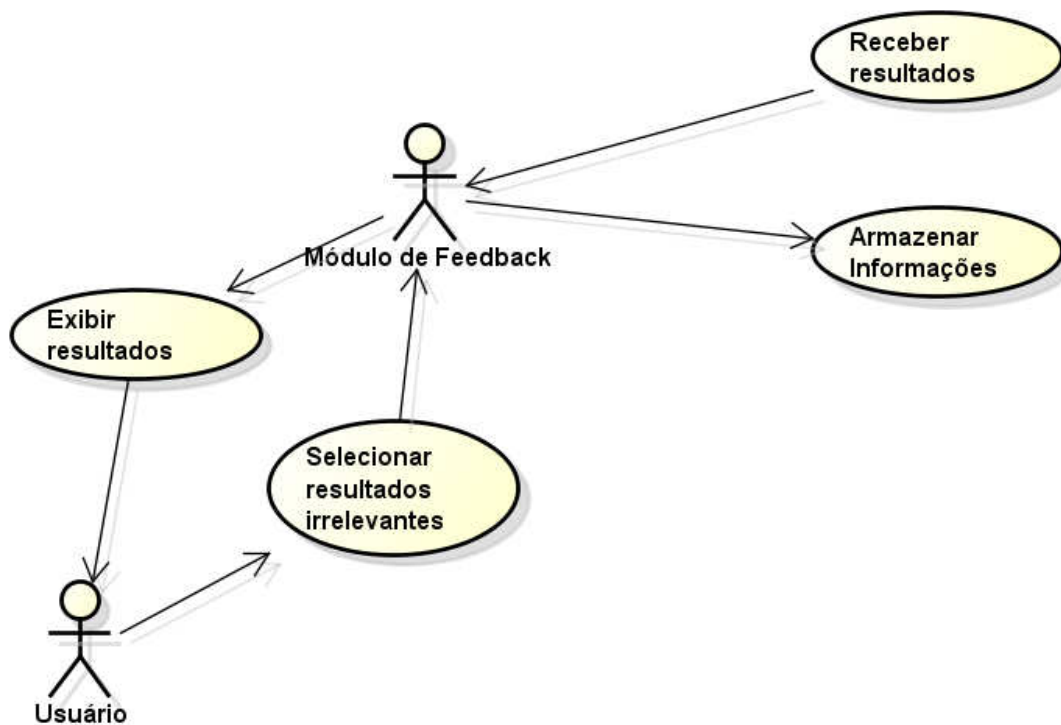


Figura 4. Casos de uso do módulo de feedback

Descrição dos casos de uso:

Caso de uso 1:	Receber resultados
Entrada:	Informações sobre os resultados da consulta realizada pelo usuário
Saída:	Conjunto de todas as informações necessárias sobre os resultados
Descrição:	O módulo de reformulação das consultas irá enviar as informações (descrição do resultado, ontologias usadas no mapeamento, tipo de relacionamento, etc.) que são necessárias para o armazenamento no banco de dados

Caso de uso 2:	Exibir resultados
Entrada:	Conjunto de informações sobre os resultados da consulta
Saída:	Resultado exibido na interface do usuário

Descrição:	O usuário irá visualizar os resultados retornados pela sua consulta
Observação:	Caso o usuário deseje identificar alguns resultados indesejados, ele irá ser levado para outra tela, onde poderá selecionar e enviar suas informações

Caso de uso 3:	Selecionar resultados irrelevantes
Entrada:	Identificação do resultado selecionado como irrelevante
Saída:	Conjunto de informações sobre o resultado selecionado
Descrição:	O usuário irá selecionar, dentre os resultados retornados, quais são considerados irrelevantes para seu domínio e enviá-los ao sistema

Caso de uso 4:	Armazenar informações
Entrada:	Conjunto de informações sobre o resultado selecionado
Saída:	Armazenamento das informações no banco de dados
Descrição:	O sistema irá salvar as informações sobre o resultado considerado equivocado pelo usuário no banco de dados para posterior refinamento dos mapeamentos semânticos

4.2 Implementação

Para a implementação do módulo de *feedback* do sistema SPEED foi escolhida a linguagem Java [Java 2010], com o auxílio da ferramenta Eclipse. A linguagem Java foi escolhida pois o módulo de reformulação de consultas foi desenvolvido também em Java, facilitando assim a integração do novo módulo com as funcionalidades já implementadas. Além disso, Java é uma linguagem independente de plataforma, facilitando assim o desenvolvimento e posterior implantação em várias plataformas diferentes com pouca ou nenhuma alteração.

A interface com o usuário do módulo de *feedback* foi desenvolvido para Web utilizando HTML5, CSS e Javascript, e executa em qualquer *browser* normalmente. A interface com o usuário é um dos principais pontos a ser pensado em sistemas com algum mecanismo de *feedback*. Sistemas que possuem a interface do seu mecanismo de *feedback* não muito amigáveis e com pouca usabilidade, tendem a não incentivar os usuários a enviarem suas opiniões sobre os resultados [Stumpf et al. 2008]. A Figura 5 mostra a interface de exibição dos resultados do módulo desenvolvido.

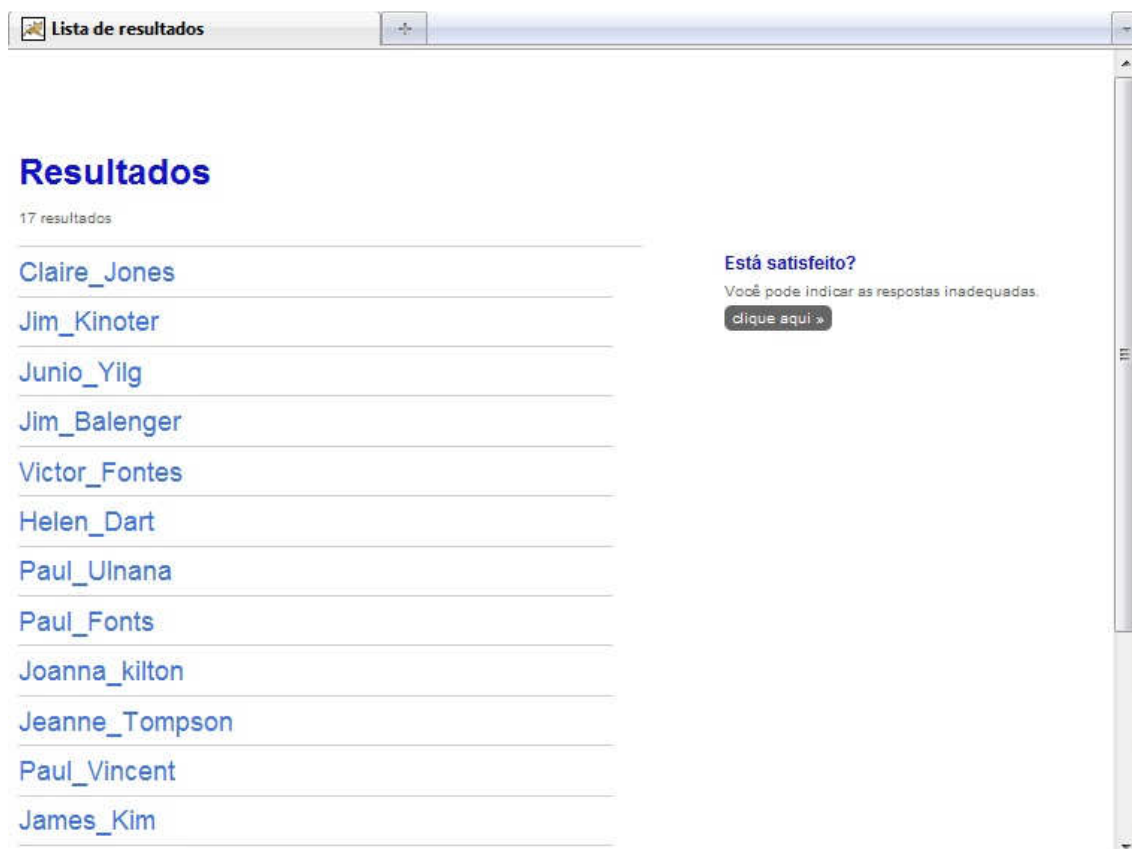


Figura 5. Tela de exibição dos resultados

A interface de visualização dos resultados deve ser o mais simples possível, para que não deixe a tela confusa e não atrapalhe o usuário. O mecanismo de aquisição de *feedback* não deve ser intrusivo, ou seja, a opção de envio das suas sugestões não deve atrapalhar a visualização dos resultados.

Caso o usuário identifique que alguns dos resultados não estão condizentes com o esperado, ele deverá ir para a tela de envio do *feedback*, clicando no link ao lado da lista de resultados. A interface de submissão do *feedback* pelo usuário é mostrada na Figura 6.

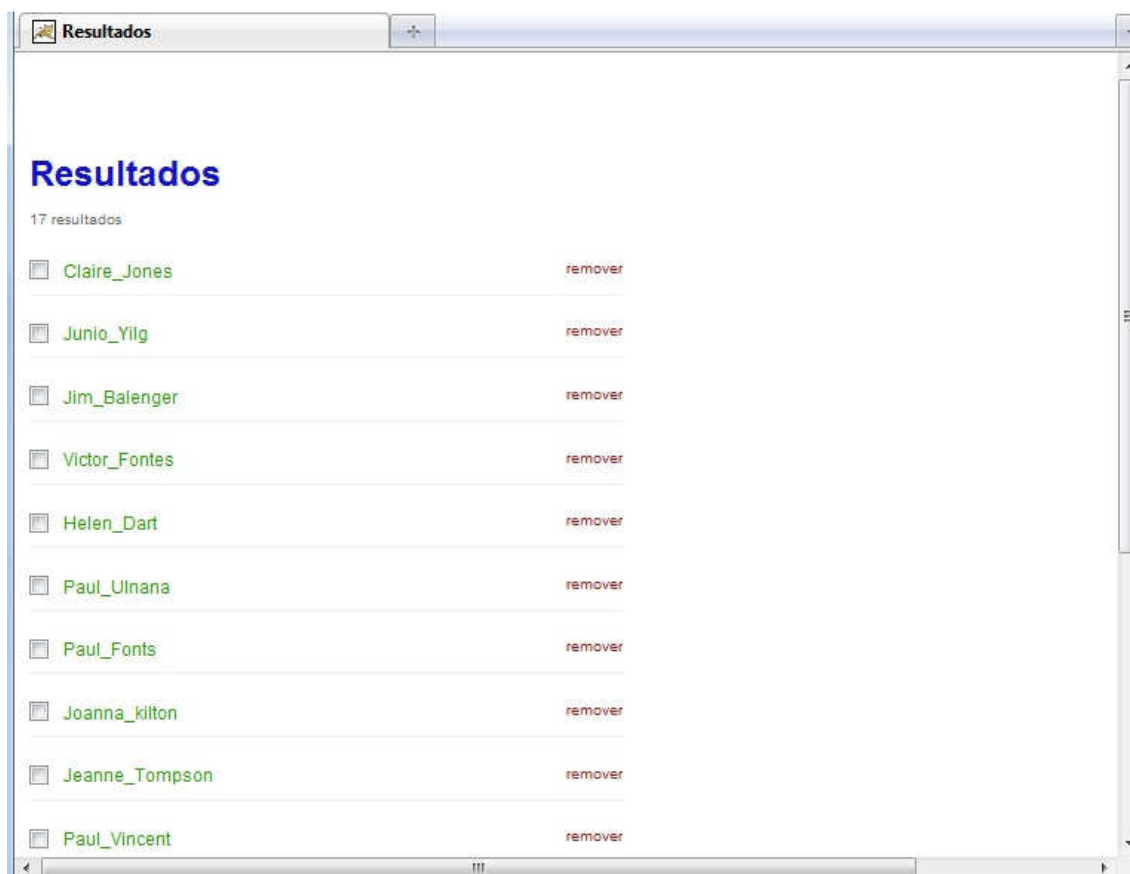


Figura 6. Interface de envio de *feedback*

Na tela de submissão do *feedback*, o usuário irá selecionar quais resultados não foram considerados corretos clicando no link “remove”. Assim, o sistema irá armazenar no banco de dados as informações referentes à correspondência semântica que gerou este resultado, para posterior uso no refinamento dos mapeamentos semânticos.

As informações sobre as correspondências que são necessárias para o armazenamento são: o nome dos conceitos envolvidos, a relação existente entre eles e as URI das duas ontologias envolvidas. Essas informações são

usadas pelo módulo de reformulação de consultas na busca pelos resultados, e são armazenadas em um *log* de reformulação, mostrando as características do processamento da consulta. Para cada resultado da consulta, também são enviadas para o módulo de *feedback* essas informações presentes no *log*.

Para a transmissão de dados entre o módulo de reformulação de consultas e o módulo de *feedback*, é usado um arquivo XML. Um exemplo de arquivo XML usado é mostrado no Quadro 1. A descrição de cada um dos campos é feita logo após.

```
<resultados>
  <resultado>
    <sujeito>FullProfessor</sujeito>
    <uri1>http://swrc.ontoware.org/ontology/portal</uri1>
    <objeto>Professor</objeto>
    <uri2>http://www.lehigh.edu/~zhp2/univ-bench.owl</uri2>
    <relacao>isSubConceptOf</relacao>
    <nome>Jim_Balenger</nome>
  <resultado>
  <resultado>
    <sujeito>FullProfessor</sujeito>
    <uri1>http://swrc.ontoware.org/ontology/portal</uri1>
    <objeto>VisitingProfessor</objeto>
    <uri2>http://www.lehigh.edu/~zhp2/univ-bench.owl </uri2>
    <relacao> isCloseTo </relacao>
    <nome> Helen_Dart</nome>
  <resultado>
</resultados>
```

Quadro 1. Exemplo de XML de envio de informações

Descrição dos campos do arquivo XML:

Campo:	Descrição:
sujeito:	Nome do sujeito da correspondência semântica
uri1	A URI da ontologia que contém o sujeito
objeto:	Nome do objeto da correspondência semântica
uri2:	A URI da ontologia que contém o objeto
relação:	O tipo da correspondência existente entre o sujeito e o objeto
nome:	Nome do resultado da consulta (o nome que irá ser mostrado na tela de visualização dos resultados)

4.3 Considerações

Este capítulo descreveu o módulo de aquisição de *feedback* dos usuários do sistema SPEED, tema central deste trabalho. Suas funcionalidades foram detalhadas através da sua especificação, além de descrever os detalhes da implementação do módulo. O uso das informações de *feedback* dos usuários possibilitará um refinamento dos mapeamentos semânticos existentes entre os pontos da rede, gerados pelo *matcher* semântico [Pereira 2008].

5. Conclusão

Este capítulo apresenta algumas considerações sobre o desenvolvimento do trabalho, além de suas contribuições. Após isso, serão apresentadas algumas sugestões de melhoria do módulo de *feedback*, servindo como trabalhos futuros.

5.1 Considerações finais

No presente trabalho, foram mostrados conceitos referentes aos mecanismos de aquisição de *feedback* dos usuários em sistemas. Foi mostrada também a importância de se ter tal funcionalidade para que os sistemas possam ir se adaptando com o uso dos usuários.

Como uma das principais funcionalidades de sistema SPEED é o processamento de consultas, percebeu-se a necessidade de ter algum mecanismo de refinamento dos resultados das consultas, pois, devido à grande quantidade de informação existente atualmente, alguns resultados irrelevantes podem ser retornados como resposta às consultas.

Logo, o desenvolvimento do módulo de *feedback* do sistema SPEED foi o objetivo principal deste trabalho. Com ele, informações sobre os resultados considerados equivocados são armazenados, para que as correspondências semânticas que geraram os resultados possam ser atualizadas.

5.2 Contribuições

O módulo desenvolvido visa suprir uma das necessidades encontradas no sistema SPEED, o do refinamento dos resultados das buscas com base nas informações passadas pelos usuários. Como pôde ser observado nas seções anteriores, vários sistemas com algum mecanismo de processamento de consultas possuem alguma forma de adquirir informações dos usuários.

No caso do módulo de *feedback* implementado no SPEED, as informações necessárias a serem armazenadas são as referentes às correspondências semânticas utilizadas na geração dos resultados das

consultas enriquecidas no momento de sua reformulação. Com essas informações, é possível obter resultados mais satisfatórios para os usuários, pois os resultados considerados equivocados serão utilizados para diminuir a importância da correspondência semântica associada nas próximas consultas.

5.3 Trabalhos Futuros

O módulo de *feedback* desenvolvido neste trabalho possui apenas uma forma de se adquirir as informações dos usuários. Outras formas podem ser adicionadas no futuro:

- Aquisição de informações dos usuários de forma implícita.
- Inclusão de alguma forma de positivar o resultado, ao invés de só negativá-los.

Referências bibliográficas

[Chandramouli 2008] Chandramouli K., Kliegr T., Nemrava J., Svatek V., Isquierdo E. (2008): Query refinement and user relevance feedback for contextualized image retrieval. In: Proceedings of the 5th International Conference on Visual Information Engineering (VIE'08), Xi'an, China.

[Elsas et al. 2009] Elsas J.L., Donmez P., Callan J., Carbonell J. (2009): Pairwise Document Classification for Relevance Feedback. In: Proceedings of the 2009 Text Retrieval Conference (TREC 2009), Gaithersburg, USA.

[Happel 2009] Happel H., Mazarakis A. (2009): Considering Information Providers in Social Search. In: Proceedings of the 2nd Workshop on Collaborative Information Seeking (CIR 2010).

[Java 2010] Java, <http://www.java.com/>, Dezembro 2010

[Kekalainen 2005] Kekalainen J. (2005): Binary and graded relevance in IR evaluations - comparison of the effects on ranking of IR systems. In: Information Processing and Management: an International Journal, v.41 n.5, p.1019-1033.

[Neves 2010] Neves T. (2010): PSemRef: Personalização de Consultas em Ambientes Distribuídos e Dinâmicos. Dissertação de Mestrado. Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

[Pereira 2008] Pereira T. (2008): Mapeamento Semântico de Ontologias no SPEED. Monografia de Conclusão de Curso. Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

[Pires 2007] Pires C. (2007): Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semântica. Monografia de Qualificação e Proposta de Doutorado, CIn, UFPE. Abril, 2007.

[Song et al. 2010] Song Y., He L. Optimal Rare Query Suggestion with Implicit User Feedback. In: Proceedings of the 19th international conference on World Wide Web (WWW'2010), pp. 901-910, Raleigh, USA.

[Souza 2009] Souza D. (2009): Using Semantics to Enhance Query Reformulation in Dynamic Distributed Environments". PhD Thesis, Federal University of Pernambuco (UFPE), Recife, PE, Brazil.

[Stumpf et al. 2008] Stumpf S., Sullivan E., Fitzhenry E., Oberst I., Wong W., Burnett M. (2008): Integrating rich user feedback into intelligent user interfaces. In: Proceedings of the 13th International Conference on Intelligent User Interfaces, January 13-16, 2008, Gran Canaria, Spain.

[Sung 2005] Sung L., Ahmed N., Blanco R., Li H., Soliman M. A., Hadaller D. (2005): A Survey of Data Management in Peer-to-Peer Systems. School of Computer Science, University of Waterloo.

[White et al. 2005] White R. W., Ruthven I., Jose J. M., Rijsbergen C. J. V. (2005): Evaluating implicit feedback models using searcher simulations. In: ACM Transactions on Information Systems (TOIS), v.23 n.3, p.325-361.

[Xiao 2006] Xiao, H. (2006): Query Processing for Heterogeneous Data Integration using Ontologies. PhD Thesis. University of Illinois at Chicago, Chicago, USA.

[Zhao et al. 2009] Zhao F. C., Oard D. W., Baron J. R. (2009): Improving Search Effectiveness in the Legal E-Discovery Process Using Relevance Feedback. In: DESI III Global E-Discovery/E-Disclosure Workshop at ICAIL 2009, Barcelona, Spain.

Assinaturas

Ana Carolina Salgado

Orientadora

Thiago Henrique Fernandes da Paz

Aluno