OPIS: Um Método para a Identificação e a Busca de Páginas-Objeto

Miriam Pizzatto Colpo[1], Edimar Manica[1,2], Renata Galante[1]

[1] Universidade Federal do Rio Grande do Sul, Brasil
[2] Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Campus Ibirubá, Brasil
{mpcolpo, edimar.manica, galante}@inf.ufrgs.br

Abstract. Este artigo propõe um novo método, denominado OPIS, para a identificação e a busca de páginas-objeto, que são páginas que representam um único objeto do mundo real na web. A motivação para este trabalho se encontra no fato de que os motores de busca convencionais não conseguem responder a buscas por páginas-objeto de forma satisfatória atualmente, já que a quantidade de páginas-objeto recuperada é bastante limitada. OPIS caracteriza-se por adotar técnicas de pré-processamento de texto e de aprendizagem de máquina na classificação de páginas. Quando integrado a um motor de busca convencional, ele permite que somente páginas classificadas como páginas-objeto sejam recuperadas pelas consultas do usuário, ao invés de todas as páginas que contêm os termos da consulta. Experimentos preliminares mostram que o OPIS melhora em média 56% da precisão dos resultados de busca por páginas-objeto, quando comparado a um motor de busca convencional.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: busca-objeto, classificação de páginas web, páginas-objeto

1. INTRODUÇÃO

`Objetos da web` são unidades de dados sobre as quais informações da web são coletadas, indexadas e ordenadas. Esses objetos são conceitos usualmente reconhecidos (como autores, artigos, conferências ou revistas), relevantes a um domínio de aplicação e que podem ser representados por um conjunto de atributos, os quais dependem do domínio do objeto [Nie et al. 2007]. `Páginas-objeto` são páginas que representam exatamente um objeto inerente na web. Isso significa que páginas que listam diversos objetos não são consideradas páginas-objeto por não representarem um objeto em particular. A busca por páginas-objeto é feita através de consultas restringidas por atributos de domínio e pode ser chamada de `busca-objeto` [Pham et al. 2010]. Uma consulta desse tipo é "*professor de banco de dados da UFRGS*", que restringe a área e a instituição de atuação de um objeto professor e tem como objetivo recuperar páginas que descrevam esse objeto.

Os motores de busca convencionais da web (do inglês, *General Search Engines* – GSEs) são programas que visam recuperar informações da web e apresentá-las, de forma organizada e eficiente, aos usuários. Basicamente, um GSE recebe um conjunto de palavras-chave e, analisando apenas o texto não estruturado, gera uma lista de páginas que contenham essas palavras [Baeza-Yates and Ribeiro-Neto 1999]. Embora os GSEs consigam atender à maioria das consultas realizadas atualmente, eles se mostram inadequados para recuperar páginas-objeto [Pham et al. 2010]. Os resultados esperados para a busca-objeto apresentada anteriormente, por exemplo, são páginas institucionais, pessoais ou de currículos, etc. que descrevam um objeto professor e considerem as restrições (de instituição e área

Este trabalho é parcialmente financiado pelo Instituto Nacional de Pesquisa da Web, pelo CNPq e pela CAPES.

de atuação) estabelecidas. Porém, na realidade, muitas páginas relacionadas a notícias, projetos e concursos para professores serão retornadas, dificultando para o usuário a localização de informações de seu interesse.

Este artigo propõe um novo método para a identificação e a busca de páginas-objetos, denominado OPIS (acrônimo para ***O**bject **P**age **I**dentifying and **S**earching*), que adota técnicas de pré-processamento de texto e de aprendizagem de máquina aplicadas na classificação baseada em conteúdo de páginas web. O OPIS envolve a integração de um classificador a um motor de busca, de modo a permitir que somente páginas identificadas (classificadas) como páginas-objeto sejam indexadas e posteriormente recuperadas pelas consultas dos usuários. A principal contribuição deste método é a melhoria na precisão de buscas-objeto, fornecendo aos usuários finais resultados que melhor atendam a suas necessidades de informação. Experimentos preliminares no domínio real de pesquisadores mostram que o OPIS superou o motor de busca convencional *Lucene*[1] com aumentos de precisão de 65% (0.600 vs. 0.364) nos primeiros cinco resultados e de 56% (0.453 vs. 0.289) nos primeiros 20.

O restante desse artigo está organizado da seguinte forma. Na Seção 2, são apresentados trabalhos relacionados. Na Seção 3, o OPIS é detalhadamente especificado. Na Seção 4, é apresentada a implementação e a experimentação de um caso de estudo no domínio de pesquisadores e, na Seção 5, o artigo é concluído e direções futuras são apontadas.

2. TRABALHOS RELACIONADOS

Muitos esforços têm sido feitos para melhorar os resultados recuperados pelos GSEs. A maioria deles propõe a criação de motores de busca verticais [Ji et al. 2009][Lee et al. 2011][Luo 2009], que são motores de busca específicos a determinados domínios, levando em consideração as particularidades de cada tópico. Além disso, outros trabalhos [Bennett et al. 2010][Geng et al. 2009][Pham et al. 2010] usam funções de *ranking* específicas para recuperar páginas relacionadas a domínios específicos. Em geral, esses trabalhos usam técnicas de processamento de texto e aprendizagem de máquina para extrair o conteúdo das páginas e, com base nisso: aprender um determinado domínio e filtrar apenas páginas consideradas pertencentes a esse domínio no processo de coleta; ou determinar as características do domínio a serem usadas em uma função de *ranking* específica. O OPIS se difere desses trabalhos à medida que não deseja aprender apenas o tópico das páginas, mas também seu tipo funcional (se a página é ou não uma página-objeto).

Blanco [Blanco et al. 2008] propõe um método para coletar automaticamente páginas da web que publicam dados relacionados à instancias de entidades conceituais com um esquema implícito. Esse método assume que o usuário fornece exemplos de páginas de entidades a partir de sites distintos e percorre cada um desses sites procurando por páginas que apresentam *templates* e caminhos similares aos respectivos exemplos. Esse trabalho difere do OPIS por focar na coleta de páginas de entidades com *templates* similares, enquanto o OPIS busca identificar páginas-objeto, sem considerar *templates* específicos, para melhorar a busca-objeto. Também com relação à coleta de páginas, Assis [Assis 2008] propõe um coletor focado para tópicos de interesse que possam ser representados por características de gênero e de conteúdo. Quando o usuário deseja buscar por páginas de planos de ensino de disciplinas de banco de dados, por exemplo, um conjunto de características (termos) que descreva o gênero (planos de ensino) e outro que descreva o conteúdo (banco de dados) devem ser informados por um usuário especializado, de modo que o coletor possa analisar cada página através da sua similaridade com os termos de ambos os aspectos. No OPIS, o conteúdo e o gênero das páginas não são considerados separadamente, o que reduz o nível de especialidade do usuário, uma vez que ele não precisa discernir entre esses dois aspectos e nem selecionar termos manualmente para caracterizá-los.

Para Pham [Pham et al. 2010], cuja proposta esta mais próxima do OPIS, o problema de busca-objeto se assemelha ao de aprendizagem de *ranking*, em que o principal objetivo é aprender uma função

[1] Apache Lucene, http://lucene.apache.org/core

de *ranking* através de uma função de aprendizagem, com base em um conjunto de características relevantes. A solução proposta consiste em desenvolver diversos motores de busca verticais para suportar a busca por páginas-objeto em diferentes domínios. Para isso, uma função de *ranking* deve ser aprendida para cada domínio específico. O desenvolvedor[2] deve submeter consultas por palavras-chave e anotar um conjunto de treinamento a partir das páginas recuperadas. Esse conjunto de treinamento tem suas características extraídas automaticamente e usadas em uma função de aprendizagem. O OPIS também adota o conteúdo das páginas para melhorar a busca-objeto através da classificação funcional (páginas-objeto ou não) das páginas. Porém, ele não considera a busca-objeto como um problema de aprendizagem de *ranking*. Ao invés disso, o processo de *ranking* fica a cargo do motor de busca ao qual acoplamos o método. Isso torna desnecessária a análise das informações estruturadas embutidas nas páginas, durante o processo de busca, para casá-las com as características que integram a função de *ranking* aprendida (por exemplo, "a palavra professor aparece no título").

3. OPIS: OBJECT PAGE IDENTIFYING AND SEARCHING

Esta seção descreve o método para identificação e busca de páginas-objeto, denominado OPIS (acrônimo para *Object Page Identifying and Searching*). O OPIS caracteriza-se por adotar técnicas de pré-processamento de texto e de aprendizagem de máquina na classificação de páginas baseada em conteúdo, a fim de permitir que somente páginas classificadas como páginas-objeto possam ser indexadas e recuperadas por consultas de usuários. O OPIS abrange a `construção de um classificador` e a `integração deste a um motor de busca convencional`, permitindo que os resultados de buscas-objeto se tornem mais precisos e, dessa forma, mais adequados às necessidades dos usuários.

Fig. 1. Visão geral do OPIS: Integração das atividades de identificação e busca

Na Figura 1 é apresentada uma visão geral do OPIS. Note que as páginas coletadas passam através de um processo de classificação e apenas as páginas classificadas como páginas-objeto seguem para a tarefa de indexação, permitindo que apenas estas sejam recuperadas em futuras consultas nesse domínio. Considerando o processo convencional realizado pelos GSEs, a única diferença é a introdução do classificador (identificador), após a tarefa de coleta, para a filtragem das páginas-objeto.

O classificador do OPIS é construído com base em um conjunto de atividades. Inicialmente, o `conjunto de treinamento`, que é a base do processo de aprendizagem, deve ser criado, sendo ne-

[2]Pessoa responsável por guiar o treinamento da função de *ranking* para um domínio específico, permitindo que usuários possam, então, submeter consultas relacionadas a esse domínio.

cessária a definição de um domínio, a coleta (que pode ser simplificada com o auxílio de um coletor focado [Chakrabarti et al. 1999][Torkestani 2012]) de um conjunto de páginas relacionadas a esse domínio e a rotulação (quanto a ser ou não uma página-objeto) dessas páginas de treinamento. Após, devido a maioria do conteúdo das páginas web ser textual e não estruturado, algumas atividades de `pré-processamento`, como a remoção de *tags* HTML, a tradução de termos estrangeiros, a remoção de *stopwords* e a representação do conteúdo através do Modelo de Espaço Vetorial (do Inglês, *Vector Space Model* – VSM) [Manning et al. 2008], são aplicadas às páginas coletadas. Por fim, a `criação do modelo de classificação`, para a identificação de páginas-objeto, pode ser iniciada. Esse passo envolve a escolha e parametrização de um algoritmo de classificação, o seu treinamento através de um conjunto de páginas e a análise de desempenho do modelo gerado.

4. CASO DE ESTUDO

Nesta seção, é apresentada a implementação e a avaliação do OPIS em um caso de estudo a fim de demonstrar a aplicação e a viabilidade do método em um domínio real. O objetivo desse caso de estudo é permitir a realização de buscas-objeto, tanto em um motor de busca convencional quanto no OPIS integrado a esse motor, e mostrar, através da avaliação dos resultados, que o OPIS melhora a precisão de buscas-objeto. O domínio escolhido foi o de pesquisadores, por possuir um número representativo de páginas-objeto disponíveis e por abranger diferentes tipos de páginas-objeto (como institucionais, pessoais e currículos).

4.1 Implementação

Inicialmente, um `conjunto de páginas` foi coletado, de modo que um subconjunto (25% dessa coleção) fosse considerado no treinamento do classificador e restante usado como coleção a ser indexada pelo motor de busca. Para isso, foram selecionadas páginas-*hub*, que se caracterizam por apresentarem uma lista de pesquisadores e *links* para suas páginas-objeto, relacionadas ao corpo docente de cursos de graduação em Ciência da Computação, Matemática e Estatística de 10 universidades brasileiras. Essas páginas-*hub* foram automaticamente analisadas, por meio de uma aplicação desenvolvida com o auxílio da biblioteca *HTML Parser*[3], e as páginas-objeto indicadas pelos *links* presentes nelas foram coletadas. A biblioteca *Google Custom Search*[4] foi utilizada para coletar exemplos negativos (páginas não objeto), através da submissão de consultas gerais, relacionadas às universidades consideradas, e do armazenamento das páginas resultantes. Após, todas as páginas armazenadas foram manualmente classificadas entre as classes `páginas-hub`, `página-objeto` e `páginas não objeto`, tendo cada uma obtido a quantidade de 114, 939 e 2012 páginas, respectivamente. A classe *hub* foi considerada nesse domínio na tentativa de melhorar a aprendizagem do classificador, uma vez que suas páginas podem conter diversas semelhanças com as páginas-objeto.

Durante o `pré-processamento`, a biblioteca *HTML Parser* foi utilizada na extração do conteúdo textual (sem *tags* HTML) das páginas web. Como a coleção é composta por páginas de universidades brasileiras e, no domínio de pesquisadores, é frequente a ocorrência de termos em Inglês, usou-se também a biblioteca *Web Translator Java*[5] para traduzir esses termos para o Português. Para representar o conteúdo textual das páginas através do modelo VSM, usou-se o filtro *StringToWordVector*, considerando TF-IDF [Manning et al. 2008] como forma de ponderação dos termos e o uso de uma lista de *stopwords* em Português, fornecida pelo *Snowball*[6], em sua parametrização. Esse filtro pertence à biblioteca de mineração *Weka*[7], usada neste trabalho por ser de código aberto e apresentar diversos algoritmos tanto para o pré-processamento dos dados quanto para o processo de aprendizagem.

[3]HTML Parser, http://htmlparser.sourceforge.net
[4]Google Custom Search, https://developers.google.com/custom-search
[5]Web Translator Java API, http://sourceforge.net/projects/webtranslator
[6]Snowball Portuguese Stop Word List, http://snowball.tartarus.org/algorithms/portuguese/stop.txt
[7]Weka Data Mining Software API, http://www.cs.waikato.ac.nz/ml/weka

Para a `criação do modelo de classificação` foram testados três algoritmos: *J48* (Árvore de Decisão), *NaiveBayes* (Probabilístico) e *LibSVM* (Máquinas de Vetores de Suporte). Embora não tenham apresentado grandes diferenças quanto à precisão da classificação, o primeiro algoritmo se mostrou mais lento e o terceiro obteve melhor precisão que os demais. Dessa forma, o *LibSVM* [Chang and Lin 2013] foi utilizado, com núcleo linear (por ser de rápida execução), através do *Weka* e treinado com 25% da coleção criada anteriormente. O motor de busca convencional *Lucene* foi usado na `integração do classificador`, de modo a permitir que somente páginas classificadas como páginas-objeto sejam indexadas e recuperadas. O *Lucene*, assim como as ferramentas já mencionadas, fornece uma biblioteca livre em Java, o que garante a portabilidade do sistema. Para a integração, uma aplicação em Java, que usa as bibliotecas *Lucene* e *Weka* para classificar, indexar e buscar páginas, foi desenvolvida. Quando o método de indexação é chamado, o documento a ser indexado passa por uma atividade adicional, onde é testado pelo classificador, e somente passa pelo processo de indexação se for classificado como uma página-objeto.

4.2 Avaliação Experimental

Experimentos foram realizados, considerando o caso de estudo desenvolvido, com o objetivo de avaliar a influência exercida pelo OPIS nos resultados de busca-objeto, em relação a um motor de busca convencional. Os experimentos foram executados em um PC padrão, usando as ferramentas mencionadas anteriormente, e consistiram na submissão de buscas-objeto do domínio de pesquisadores e na avaliação da relevância dos resultados recuperados por essas consultas. Participaram desses experimentos 10 usuários, que submeteram e avaliaram cinco consultas cada. A fim de tornar o processo menos exaustivo, os usuários foram divididos em dois grupos, de modo que cada grupo realizasse suas consultas em apenas um dos métodos, ou seja, no motor de busca com a adição do classificador (OPIS) ou sem (*baseline*). Após, essas consultas foram reproduzidas e avaliadas no método oposto. Para evitar o uso de diferentes critérios de relevância durante a reprodução das consultas, os usuários especificaram o objetivo e os tipos de resultados que consideraram relevantes em cada uma de suas consultas através de uma área de texto adicional presente na interface de usuário, que foi especialmente desenvolvida para a realização desses experimentos.

Como não era viável solicitar que os usuários avaliassem todos os resultados recuperados pelas consultas, apenas os 20 primeiros foram apresentados para serem analisados. Para medir a precisão das páginas apresentadas e ter um indicativo de suas posições no *ranking*, a métrica de precisão em n (p@n)[Manning et al. 2008], que considera somente os primeiros n resultados recuperados pelo sistema, foi usada com n de 5, 10, 15, e 20.

As médias para todas as consultas em ambos os casos (com e sem o OPIS) são apresentadas na Tabela I. Considerando as 20 primeiras páginas retornadas, o OPIS apresenta uma melhora de 56% na precisão. Isso acontece devido ao fato de muitas páginas irrelevantes (páginas não objeto) retornadas pelo motor de busca convencional para buscas-objeto serem descartadas pelo OPIS, o que torna os resultados mais precisos. Com base no Teste-T, é possível afirmar que o OPIS melhora significativamente (valor-p do Teste-T < 0.01) a precisão para buscas-objeto, em todos os pontos de corte considerados, quando comparado ao motor de busca convencional.

Table I. Resultados para todas as consultas em ambos os métodos

	Média das 50 consultas dos usuários			
	p@5	p@10	p@15	p@20
OPIS	0.600	0.526	0.489	0.453
Baseline	0.364	0.342	0.312	0.289

5. CONSIDERAÇÕES FINAIS

Neste artigo, foi proposta uma solução para o problema de busca-objeto em motores de busca convencionais, através de um novo método, denominado OPIS, para a identificação e a busca de páginas-objeto. O OPIS usa técnicas aplicadas na classificação baseada em conteúdo da web, como atividades de pré-processamento de texto e algoritmos de aprendizagem de máquina, para permitir que apenas páginas classificadas como páginas-objeto sejam indexadas e recuperadas pelas consultas dos usuários.

O OPIS teve sua viabilidade e aplicação demonstrada através da implementação de um caso de estudo no domínio de pesquisadores. Experimentos preliminares foram realizados nesse caso de estudo, contando com a contribuição de usuários, que submeteram e avaliaram buscas-objeto tanto em um motor de busca convencional quanto no OPIS integrado a esse motor. Os resultados mostraram que o OPIS forneceu um ganho de aproximadamente 56%, considerando a média das precisões das 20 primeiras páginas recuperadas por todas as consultas.

Como trabalhos futuros, pretende-se simplificar a construção do classificador através do uso de realimentação de relevância e reduzir o problema de ambiguidade das palavras-chave por meio da expansão automática de consultas. Uma experimentação mais exaustiva, considerando um domínio adicional e o trabalho de busca-objeto desenvolvido por Pham [Pham et al. 2010], apresentado nos trabalhos relacionados, como *baseline*, também faz-se necessária.

REFERENCES

Assis, G. T. *Uma Abordagem Baseada em Gênero para Coleta Temática de Páginas da Web*. M.S. thesis, Universidade Federal de Minas Gerais (UFMG), Brazil, 2008.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

Bennett, P. N., Syore, K., and Dumais, S. T. Classification-Enhanced Ranking. In *Proceedings of the International World Wide Web Conferences*. Raleigh, EUA, pp. 111–120, 2010.

Blanco, L., Crescenzi, V., Merialdo, P., and Papotti, P. Supporting the Automatic Construction of Entity Aware Search Engines. In *Proceedings of the ACM Workshop on Web Information and Data Management*. Napa Valley, EUA, pp. 149–156, 2008.

Chakrabarti, S., Berg, M., and Dom, B. Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery. In *Proceedings of the International World Wide Web Conferences*. Toronto, Canada, pp. 1623–1640, 1999.

Chang, C. and Lin, C. LIBSVM: A Library for Support Vector Machines, 2013. http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Geng, B., Yang, L., Xu, C., and Hua, X. Ranking Model Adaptation for Domain-Specific Search. In *Proceedings of the Conference on Information and Knowledge Management*. Hong Kong, China, pp. 197–206, 2009.

Ji, L., Yan, J., Liu, N., Zhang, W., Fan, W., and Chen, Z. ExSearch: A Novel Vertical Search Engine for Online Barter Business. In *Proceedings of the Conference on Information and Knowledge Management*. Hong Kong, China, pp. 1357–1366, 2009.

Lee, H., Nazareno, F., Jung, S., and Cho, W. A Vertical Search Engine for School Information Based on Heritrix and Lucene. In *Proceedings of the International Conference on Convergence and Hybrid Information Technology*. Daejeon, Korea, pp. 344–351, 2011.

Luo, G. Design and Evaluation of the iMed Intelligent Medical Search Engine. In *Proceedings of the IEEE International Conference on Data Engineering*. Shanghai, China, pp. 1379–1390, 2009.

Manning, C. D., Raghavan, P., and Schütze, H. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

Nie, Z., Ma, Y., Shi, S., Wen, J., and Ma, W. Web Object Retrieval. In *Proceedings of the International World Wide Web Conferences*. Banff, Canada, pp. 81–90, 2007.

Pham, K. C., Rizzolo, N., Small, K., Chang, K. C., and Roth, D. Object Search: Supporting Structured Queries in Web Search Engines. In *Proceedings of the NAACL HTL - Workshop on Semantic Search*. Los Angeles, EUA, pp. 44–52, 2010.

Torkestani, J. A. An Adaptive Focused Web Crawling Algorithm Based on Learning Automata. *Applied Intelligence* vol. 37, pp. 586–601, 2012.