UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MIRIAM PIZZATTO COLPO

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

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof[a]. Dr[a]. Renata Galante
Orientadora

Porto Alegre, agosto de 2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Colpo, Miriam Pizzatto

OPIS: Um Método para Identificação e Busca de Páginas-Objeto / Miriam Pizzatto Colpo. – Porto Alegre: PPGC da UFRGS, 2014.

79 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Orientadora: Renata Galante.

1. Página-objeto. 2. Busca-objeto. 3. Realimentação de relevância. 4. Classificação de páginas web. I. Galante, Renata. II. Título.

“A gente sempre deve sair à rua como quem foge de casa,
Como se estivessem abertos diante de nós todos os caminhos do mundo.
Não importa que os compromissos, as obrigações, estejam ali...
Chegamos de muito longe, de alma aberta e o coração cantando!”
— MARIO QUINTANA

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por abençoar-me com a família que tenho, através da qual recebo, incondicionalmente, toda a força e orientação necessárias para seguir minha vida, buscando a realização do que anseio e acredito.

Aos meus pais, Cláudio e Eliane, pela educação e pelas oportunidades proporcionadas, pela confiança e fé que sempre depositaram em mim, pelo amor e constante incentivo e, principalmente, pelo exemplo de vida e caráter. À minha irmã, Danieli, pelo companheirismo, por todos os ensinamentos e pela amizade inquestionável. Ao meu irmão caçula, Miccael, por se fazer sempre presente, por cuidar do pai e da mãe durante esses longos anos de minha formação profissional e por fazer a nossa família tão feliz.

À minha orientadora, Renata Galante, pela oportunidade de cursar o mestrado, por mostrar-se sempre acessível e disposta a compartilhar seus conhecimentos e experiências, e pela compreensão, confiança e atenção que me foram dispensadas ao longo desse período.

Aos amigos de todas as fases da minha vida, em especial à Glivia e à Laurem, pela compreensão, paciência e torcida, pelos conselhos e incentivos, por todas as risadas e momentos compartilhados. Aos amigos que ficaram da graduação, em especial ao Felipe (vulgo FF), pelos conselhos e por fazer piada de todos os problemas, trazendo um alento para os dias difíceis; ao Bruno, também colega no mestrado, por ter aguentado, quase diariamente, meu mau humor e impaciência, aflorados nesse último ano; ao Cristiano e ao Grahl, antigos colegas de laboratório na graduação e com quem, infelizmente, perdi o convívio diário na UFRGS, pelas conversas, ajudas e risadas.

Aos colegas do Laboratório de Banco de Dados da UFRGS, Anderson, Edimar, Ernando, Eduardo, Guilherme, Gustavo, Márcio, Maurício, Renato e Solange, pelo acolhimento, pelas risadas e pelas participações em prévias e experimentos. Um obrigada reforçado ao Edimar, por ter colaborado diretamente para a evolução deste trabalho ao participar de nossas reuniões semanais, fornecendo inúmeras ideias e sugestões, e ao revisar meus textos.

À UFRGS e ao Instituto de Informática, pela infraestrutura disponibilizada e pelo apoio financeiro na participação de conferências. Ao Instituto Nacional de Ciência e Tecnologia para a Web (InWeb), ao CNPq e à CAPES, pelo apoio financeiro.

Aos professores do PPGC, pelos ensinamentos fornecidos, em especial aos professores Carlos Alberto Heuser e Viviane Moreira, pelas sugestões e ideias advindas do seminário de andamento, e à professora Karin Becker, por apontar caminhos alternativos para possíveis melhorias deste trabalho.

Por fim, agradeço aos membros da banca examinadora, professores Carlos Alberto Heuser, Carina F. Dorneles e Viviane Moreira, por aceitarem o convite e dedicarem parte de seu tempo a esse trabalho.

OPIS: A Method for Object Page Identifying and Searching

ABSTRACT

Object pages are pages that represent exactly one inherent real-world object on the web, regarding a specific domain, and the search for these pages is named as object search. General Search Engines (GSE) can satisfactorily answer most of the searches performed in the web nowadays, however, this hardly occurs with object search, since, in general, the amount of retrieved object pages is limited. This work proposes a method for both identifying and searching object pages, named OPIS (acronyms to **O**bject **P**age **I**dentifying and **S**earching). The kernel of OPIS is to adopt relevance feedback and machine learning techniques in the task of content-based classification of object pages. OPIS does not discard the use of GSEs and, instead, in his search step, proposes the integration of a classifier to a GSE, adding a filtering step to the traditional search process. This simple approach allows that only pages identified as object pages are retrieved by user queries, improving the results for object search. Experiments with real datasets show that OPIS outperforms the baseline with average boost of 47% considering the average precision.

Keywords: object page, object search, relevance feedback, web page classification.

RESUMO

Páginas-objeto são páginas que representam exatamente um objeto inerente do mundo real na web, considerando um domínio específico, e a busca por essas páginas é chamada de busca-objeto. Os motores de busca convencionais (do Inglês, *General Search Engine* - GSE) conseguem responder, de forma satisfatória, à maioria das consultas realizadas na web atualmente, porém, isso dificilmente ocorre no caso de buscas-objeto, uma vez que, em geral, a quantidade de páginas-objeto recuperadas é bastante limitada. Essa dissertação propõe um novo método para a identificação e a busca de páginas-objeto, denominado OPIS (acrônimo para ***O**bject **P**age **I**dentifying and **S**earching*). O cerne do OPIS está na adoção de técnicas de realimentação de relevância e aprendizagem de máquina na tarefa de classificação, baseada em conteúdo, de páginas-objeto. O OPIS não descarta o uso de GSEs e, ao invés disso, em sua etapa de busca, propõe a integração de um classificador a um GSE, adicionando uma etapa de filtragem ao processo de busca tradicional. Essa abordagem permite que somente páginas identificadas como páginas-objeto sejam recuperadas pelas consultas dos usuários, melhorando, assim, os resultados de buscas-objeto. Experimentos, considerando conjuntos de dados reais, mostram que o OPIS supera o *baseline* com ganho médio de 47% de precisão média.

Palavras-chave: Página-objeto, busca-objeto, realimentação de relevância, classificação de páginas web.

LISTA DE ABREVIATURAS E SIGLAS

GSE	General Search Engine
OPIS	Object Page Identifying and Searching
MAP	Mean Average Precision
RI	Recuperação de Informação
VSM	Vecture Space Model
KDD	Knowledge Discovery in Databases
HTML	HyperText Markup Language
TF-IDF	Term Frequency/Inverse Document Frequency
SWT	Structure-oriented Weighting Technique
FI	Filtragem de Informação
TREC	Text REtrieval Conference
SVM	Support Vector Machine
ODP	Open Directory Project
URL	Uniform Resource Locator
NB	Naive Bayes
SW	Semantic Web
ENS	Entity Name System
RDF	Resource Description Framework
OSE	Object Search Engine
LBJ	Learning Based Java
WEKA	Waikato Environment for Knowledge Analysis
kNN	k-Nearest Neighbor
MLP	Multilayer Perceptron
JAR	Java Archive
API	Application Programming Interface
GCS	Google Custom Search

p@n	Precisão em n
JSP	JavaServer Pages
AvP	Average Precision

LISTA DE FIGURAS

LISTA DE TABELAS

===

SUMÁRIO

1 INTRODUÇÃO

Os motores de busca convencionais da web (em Inglês, *General Search Engines* – GSEs) são programas que visam recuperar de forma eficiente informações contidas na web e apresentá-las organizadamente aos seus usuários (BAEZA-YATES; RIBEIRO-NETO, 1999). Basicamente, um motor de busca recebe um conjunto de palavras-chave e, analisando o texto não estruturado, gera uma lista de páginas nas quais essas palavras foram encontradas. Embora os GSEs consigam suprir as necessidades da maioria das consultas realizadas atualmente, eles se mostram muitas vezes inadequados para recuperar páginas que descrevam objetos do mundo real, como, por exemplo, buscar por um professor dado seus atributos de nome, universidade ou departamento (PHAM et al., 2010).

`Objetos web` são unidades de dados sobre as quais informações são coletadas, indexadas e ordenadas. Esses objetos são, em geral, conceitos reconhecíveis (como autores, artigos, conferências ou revistas) e relevantes para um domínio de aplicação, que podem ser representados por um conjunto de atributos, sendo este definido de acordo com os requisitos específicos do domínio de cada tipo de objeto (NIE et al., 2007).

`Páginas-objeto` são páginas que representam exatamente um objeto inerente do mundo real na web, considerando um domínio específico. Isso significa que páginas que listam diversos objetos de um mesmo domínio não são consideradas páginas-objeto, uma vez que elas não descrevem um objeto em particular. Páginas de departamentos de universidades que listam diversos professores, por exemplo, não são consideradas páginas-objeto para o domínio de professor; porém, poderiam ser consideradas páginas-objeto para o domínio de departamentos.

A busca por páginas-objeto é feita através de consultas de usuários que restringem os atributos de um domínio de objeto e pode ser chamada de `busca-objeto` (PHAM et al., 2010). Um exemplo desse tipo de consulta é "*professor de banco de dados da UFRGS*", que restringe a área e a instituição de atuação de um objeto professor.

O problema apresentado pelos motores convencionais ao responder buscas-objeto é mostrado na Figura 1.1, que apresenta a primeira página de resultados recuperados pelo Google[1] para a consulta por palavras-chave "*professor UFRGS*". Resultados esperados para essa consulta são páginas que descrevam professores que lecionem na UFRGS, podendo ser, por exemplo, páginas institucionais, páginas pessoais, currículos, etc. Porém, pode-se perceber que, dentre as 10 primeiras páginas recuperadas pelo Google, nenhuma satisfaz a esse desejo, estando a maioria relacionada a notícias e a concursos de professores. O resultado que mais se aproximaria de uma página-objeto seria o último, relacionado ao blog de um professor da instituição. Porém, ainda assim, essa página não corresponde

[1] http://www.google.com.br

a uma página-objeto, uma vez que, embora possa apresentar uma pequena seção que descreva e ofereça informações do professor (autor), a mesma possui como principal objetivo apresentar informações e comentários sobre os mais diversos assuntos de interesse deste professor e não descrevê-lo especificamente. Dentre as limitações do processo de busca convencional que causam esse problema, encontra-se a ambiguidade das palavras-chave. Por exemplo, mesmo que o objetivo de uma busca com a palavra-chave "*Paris*" seja encontrar páginas relacionadas à cidade capital da França, muitas páginas relacionadas ao primeiro nome de "*Paris Hilton*" serão retornadas (MIKLÓS et al., 2010).

Figura 1.1: Resultados recuperados pelo Google para a consulta "*professor UFRGS*".

A maioria dos trabalhos que objetivam melhorar os resultados dos GSEs propõe a cri-

ação de novos motores de busca e funções de *ranking*, ambos específicos a determinados domínios, a fim de considerar as particularidades de cada domínio. PHAM et al. (2010), que introduziu o conceito de página-objeto, propõe que uma função de *ranking* seja treinada para cada novo domínio de páginas-objeto. Para isso, um usuário deve fornecer exemplos de páginas-objeto e escolher quais características, dentre um conjunto extraído automaticamente, são mais apropriadas para constituírem a função de *ranking* do novo domínio. A fim de tratar o problema de busca-objeto com uma participação menos efetiva do usuário, o presente trabalho propõe o OPIS.

OPIS (acrônimo para ***O**bject **P**age **I**dentifying and **S**earching*) é um método de `identificação` e `busca` de páginas-objeto. Para a `identificação`, o OPIS adota técnicas de realimentação de relevância, pré-processamento de texto e aprendizagem de máquina na tarefa de classificação baseada em conteúdo de páginas web. Um modelo de classificação é criado para cada novo domínio através da ajuda de um usuário, que fornece exemplos de páginas-objeto, mas não precisa selecionar um subconjunto de características para representar o domínio, o que reduz seu esforço e nível de especialidade. O OPIS não descarta o uso de GSEs e, ao invés disso, em sua etapa de `busca`, propõe a integração de um classificador a um GSE, adicionando uma etapa de filtragem ao processo de busca convencional. Essa simples abordagem permite que somente páginas identificadas como páginas-objeto sejam recuperadas pelas consultas dos usuários, melhorando, assim, os resultados de buscas-objeto. As principais contribuições desse método estão na melhoria da precisão média dos resultados de buscas-objeto e na redução do esforço gasto pelos usuários para este fim.

O OPIS foi avaliado nos domínios reais de pesquisador, *notebook* e câmera digital, considerando como *baseline* o trabalho de PHAM et al. (2010). Além de compartilhar do mesmo objetivo que o OPIS (melhoria dos resultados de busca-objeto), a abordagem utilizada por PHAM et al. (2010) também independe de páginas anotadas semanticamente (ainda escassas na web), o que contribuiu para sua adoção. Os experimentos consistiram na submissão de 51 buscas-objeto para cada domínio e na avaliação das 100 primeiras páginas recuperadas por cada consulta. Os resultados mostram que o OPIS supera o *baseline* com ganhos de MAPs (do Inglês, *Mean Average Precision*) de 38%, 43% e 61% para os domínio de pesquisador, *notebook* e câmera, respectivamente.

O restante deste trabalho esta organizado da seguinte forma. No Capítulo 2 é fornecida uma fundamentação teórica, com os principais conceitos necessários para a compreensão desta dissertação. No Capítulo 3 são apresentados os principais trabalhos relacionados. No Capítulo 4 é descrito o OPIS, o método proposto para a identificação e a busca de páginas-objeto, incluindo sua implementação. No Capítulo 5 é apresentada uma avaliação experimental, considerando casos de estudo relacionados aos domínios de pesquisador, *notebook* e câmera digital. E, por fim, no Capítulo 6 este trabalho é concluído e possíveis direções futuras são apontadas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos necessários para a compreensão desta dissertação. Na Seção 2.1 é apresentado o que são e como funcionam, em geral, os motores de busca convencionais da web. Na Seção 2.2 o conceito de objetos web, incluindo as definições de páginas-objeto e de buscas-objeto, é abordado. E, finalmente, na Seção 2.3 é fornecida uma fundamentação sobre a classificação de páginas web, incluindo definições, formas de representação e pré-processamento de páginas, além dos relacionamentos entre a classificação e as áreas de Busca na Web e Filtragem de Informação. Esses conceitos se relacionam, uma vez que o presente trabalho visa a melhoria dos resultados de busca-objeto, recuperados pelos motores de busca convencionais, por meio da adoção de classificação de páginas-web.

2.1 Motores de Busca Convencionais da Web

A área de Recuperação de Informação (RI) surgiu antes do advento da Internet, em resposta a diversos desafios decorrentes do acesso à informação, com o objetivo de fornecer abordagens para a busca de variadas formas de conteúdo. Inicialmente, as pesquisas se concentravam apenas nos problemas relacionados a publicações e registros bibliotecários, porém, logo conteúdos oriundos de outras áreas da informação, como jornalismo, direito e medicina, tornaram-se objetos de estudo. Grande parte da pesquisa científica de RI continua considerando este contexto, porém, com a popularização da Internet e o crescimento gradual da quantidade de informação disponibilizada *on-line*, muitos outros desafios foram surgindo e despertando interesse. Na década de 1990, estudos mostravam que a maioria das pessoas preferia obter informações a partir de outras pessoas em vez de usar sistemas de RI. Já na década seguinte, os avanços alcançados na área elevaram os níveis de qualidade dos motores de busca da Internet, tornando-os meio padrão de acesso à informação (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Os motores de busca convencionais da web (em Inglês, *General Search Engines* – GSEs) são programas que visam recuperar, de forma eficiente, informações contidas na web e apresentá-las organizadamente aos seus usuários (BAEZA-YATES; RIBEIRO-NETO, 1999). O Google[1], o Yahoo[2] e o Bing[3] são exemplos de GSEs. Segundo CAMPOS; DIAS (2005), um sistema de busca é um conjunto organizado de computadores, índices, bases de dados e algoritmos que juntos tem a função de analisar e indexar páginas da web, armazenar os resultados dessas atividades em uma base de dados e devolvê-

[1] http://www.google.com.br
[2] http://www.yahoo.com.br
[3] http://www.bing.com.br

los posteriormente, adequando-os aos requisitos expressos pelos usuários através de uma consulta.

Um maior detalhamento sobre o processo de busca e os tipos de consultas relacionados aos motores de busca são apresentados a seguir, nas Subseções 2.1.1 e 2.1.2, respectivamente.

2.1.1 Processo de Busca

Basicamente, um GSE recebe um conjunto de palavras-chave e, analisando o texto não estruturado, gera uma lista de páginas nas quais essas palavras foram encontradas. Para isso, três funções principais são executadas (MANNING; RAGHAVAN; SCHÜTZE, 2008):

i. `Coleta` (*crawling*) – responsável por descobrir novos documentos e páginas na Internet, tornando-os consultáveis. Essa tarefa é realizada por robôs, denominados *spiders* ou *crawlers* (rastejadores), que automática e recursivamente navegam entre as páginas, através de seus *hyperlinks*, visitando, catalogando e armazenando as páginas encontradas;

ii. `Indexação` (*Indexing*) – responsável por armazenar as palavras contidas em páginas e documentos, obtidos pelo processo de coleta, em uma estrutura de índice, a fim de permitir que consultas por palavras-chave sejam executadas posteriormente nesses documentos;

iii. `Busca` (*Searching*) – responsável por percorrer o índice, resultante do processo de indexação, a fim de encontrar páginas ou documentos que possuam correspondências com a consulta.

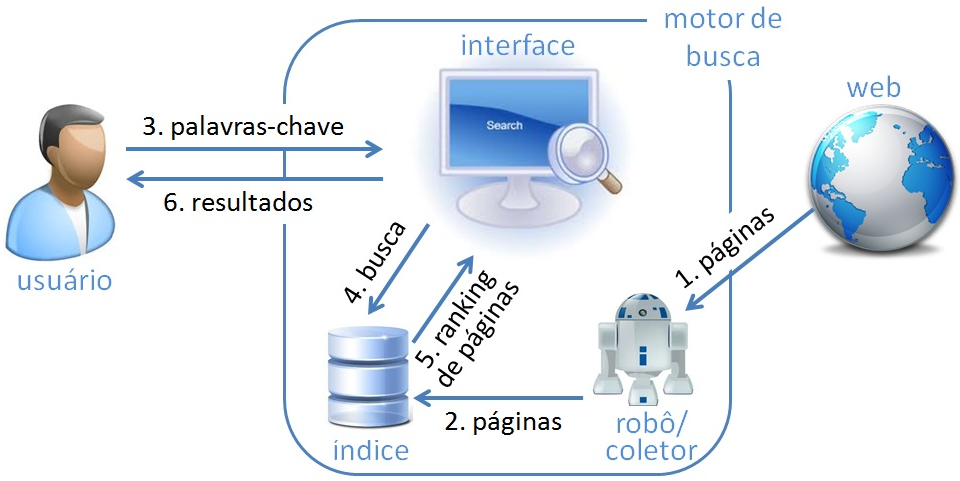

Figura 2.1: Processo de busca de um GSE.

A Figura 2.1 ilustra, de forma simplificada, o processo de busca realizado pelos GSEs. O coletor (robô) descobre páginas na web (`coleta`) e as encaminha para o índice, que armazenará as palavras contidas nessas páginas (`indexação`). Quando um usuário submete sua consulta por palavras-chave na interface de busca, as palavras-chave são procuradas no índice (`busca`), sendo retornada uma lista ordenada (*ranking*) das páginas que

contêm essas palavras. Essa lista é, então, apresentada como resultado de busca para o usuário.

A identificação das páginas relevantes para uma consulta é feita através de uma comparação entre as características apresentadas na consulta e as características presentes nas páginas indexadas. Na maioria das vezes, essas características podem ser encontradas no código (como, por exemplo: *hyperlink*, *title*, *header*, *ALT tags*) ou no próprio conteúdo da página. Essa análise de similaridade entre as características da consulta e as das páginas é realizada, na maioria das vezes, por técnicas baseadas no Modelo de Espaço Vetorial (em inglês, *Vector Space Model* - VSM). Nesse modelo, também conhecido como *Bag of Words*, as páginas e as consultas são representadas como vetores, onde cada dimensão corresponde a um termo (que geralmente é uma palavra) e os valores de cada dimensão são atribuídos através de uma ponderação da frequência em que esses termos ocorrem nas páginas (SANTOS, 2009).

2.1.2 Tipos de Consultas

Segundo MANNING; RAGHAVAN; SCHÜTZE (2008), a maioria das consultas realizadas em GSEs pode ser agrupada, de acordo com as necessidades e intenções dos usuários, em três tipos:

- Consultas Informacionais – visam a obtenção de informações sobre um tópico, como "`leucemia` ", por exemplo. Em geral, uma única página web não contém todas as informações desejadas (não sendo suficiente para satisfazer esse tipo de consulta) e o usuário tenta assimilar informações oriundas de várias páginas;

- Consultas Navegacionais – buscam pelo *site* ou página principal de uma única entidade, como "`companhia aérea TAM` ", por exemplo. Nesses casos, os usuários não estão interessados em várias páginas que contenham o termo `TAM`, mas sim em encontrar, preferencialmente como primeiro resultado de busca, o *site* da companhia;

- Consultas Transacionais – visam a obtenção de uma lista de serviços que forneçam interfaces para que o usuário realize uma transação na web, como comprar um produto, fazer uma reserva ou o *download* de um arquivo.

Faz-se importante observar que uma consulta pode ser considerada como pertencente a mais de um categoria, como é o caso da consulta dada como exemplo de tipo navegacional, a qual também poderia ser classificada como transacional, considerando que o usuário tivesse a intenção de comprar uma passagem aérea. Além disso, os tipos de consultas não determinam apenas quais e como os resultados devem ser retornados, mas também a adequação da consulta aos resultados patrocinados, uma vez que as consultas podem revelar uma intenção de compra.

2.2 Objeto Web

Objetos web são unidades de dados sobre as quais as informações da web são coletadas, indexadas e ordenadas. Esses objetos são, em geral, conceitos reconhecíveis (como autores, artigos, conferências ou periódicos) e relevantes para um domínio de aplicação, que podem ser representados através de um conjunto de atributos, sendo este definido de acordo com os requisitos específicos do domínio de cada tipo de objeto (NIE et al., 2007).

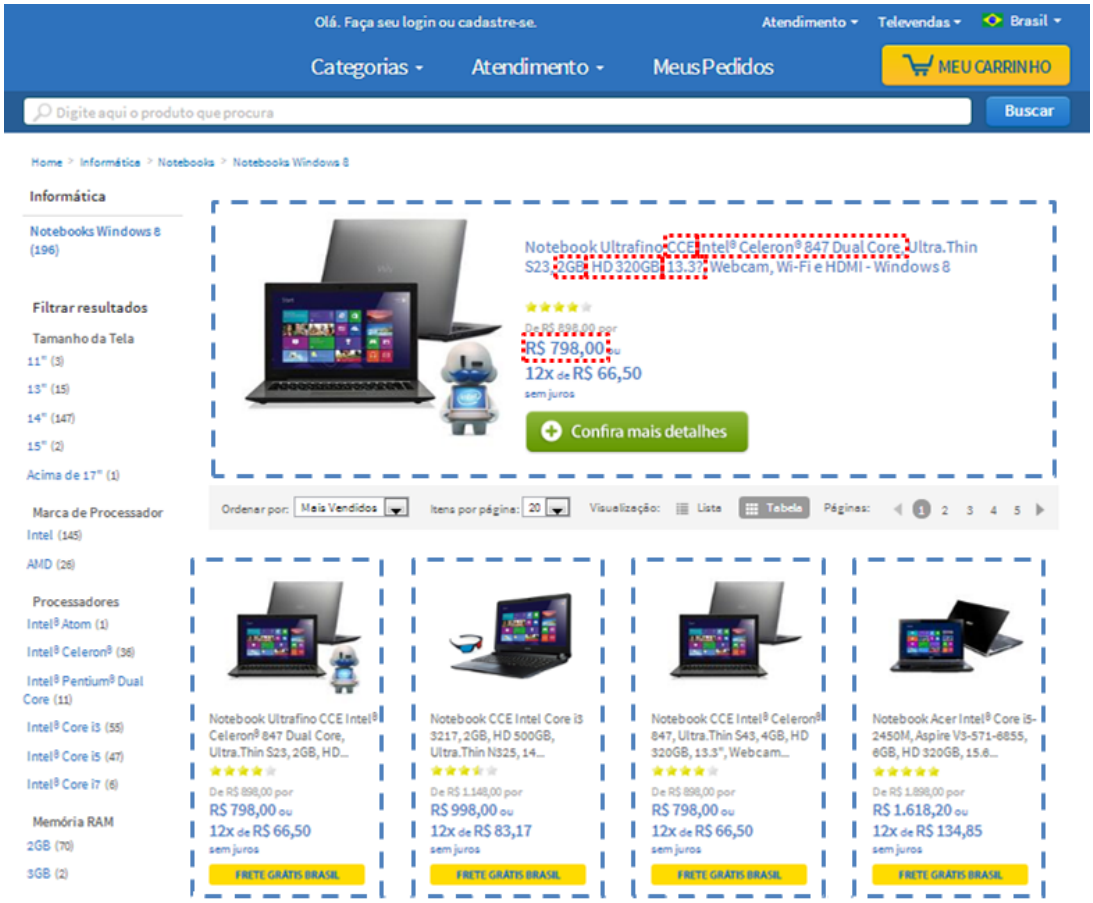

Figura 2.2: Exemplos de objetos web para o domínio de *notebook*.

A Figura 2.2 apresenta, como exemplo, uma página que contém cinco objetos web, os quais aparecem em destaque com borda tracejada (azul), para o domínio de *notebook*. Também são mostrados, em destaque com borda pontilhada (vermelha), seis dos atributos que representam um desses objetos (marca, processador, memória RAM, HD, tamanho de tela e preço).

2.2.1 Página-Objeto

Páginas-objeto são páginas que representam um objeto inerente do mundo real na web, considerando um determinado domínio. Isso significa que páginas que listam diversos objetos de um mesmo domínio não são consideradas páginas-objeto pois, apesar de mencionarem, elas não descrevem um objeto em particular. Em geral, pode-se dizer que existem poucas páginas-objeto relacionadas a um objeto especifico e muitas que o mencionam (PHAM et al., 2010).

Considerando essa definição, a página de um departamento de universidade, que lista diversos professores, não é uma página-objeto para o domínio de professor/pesquisador, assim como a página apresentada na Figura 2.2, que lista vários *notebooks*, não é uma página-objeto para o domínio de *notebook*.

Porém, é importante notar que o conceito de página-objeto está diretamente relacionado a um domínio. Desse forma, o julgamento de um página, quanto a ser ou não uma página-objeto, muda de domínio para domínio. Uma página de departamento de universidade, mesmo não sendo considerada uma página-objeto para o domínio de pro-

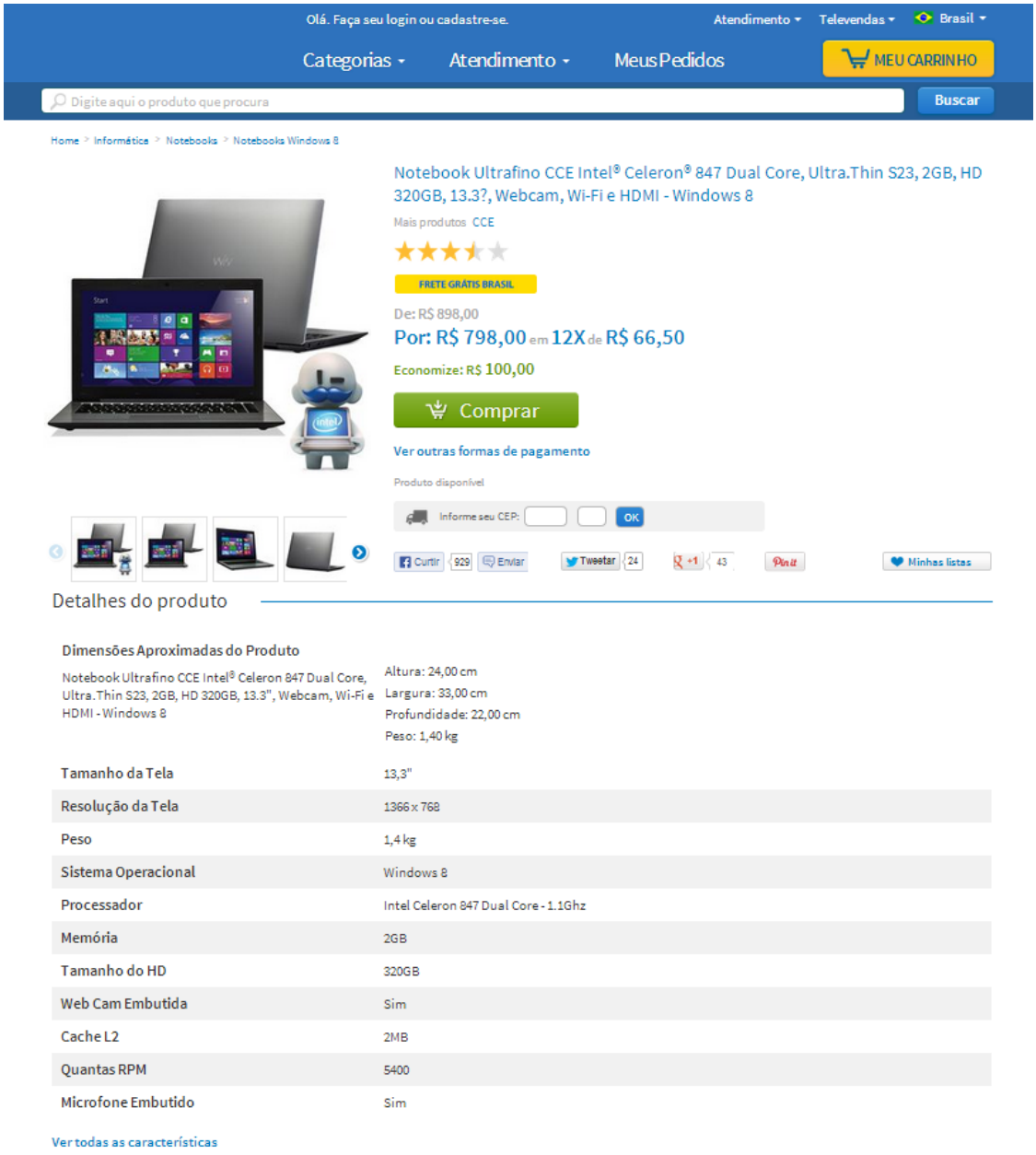

Figura 2.3: Exemplo de página-objeto para o domínio de *notebook*.

fessor/pesquisador, é uma página-objeto para o domínio de departamento, por exemplo.

Na Figura 2.3 é apresentada uma página-objeto para o domínio de *notebook*. Esse exemplo se refere a uma página de um determinado modelo de *notebook* que se encontra no site de uma loja virtual, porém, poderia ter sido apresentada uma página do fabricante ou de outas lojas virtuais que também descrevessem um único objeto desse domínio.

Na Figura 2.4, uma página-objeto para o domínio de professor/pesquisador é mostrada. Além dessa página institucional, o mesmo objeto professor poderia ter outras páginas-objeto, como sua página pessoal ou de currículo.

2.2.2 Busca-Objeto

PHAM et al. (2010) definem o problema de busca-objeto como a atividade de encontrar páginas que se referem apenas ao objeto consultado (páginas-objeto), considerando

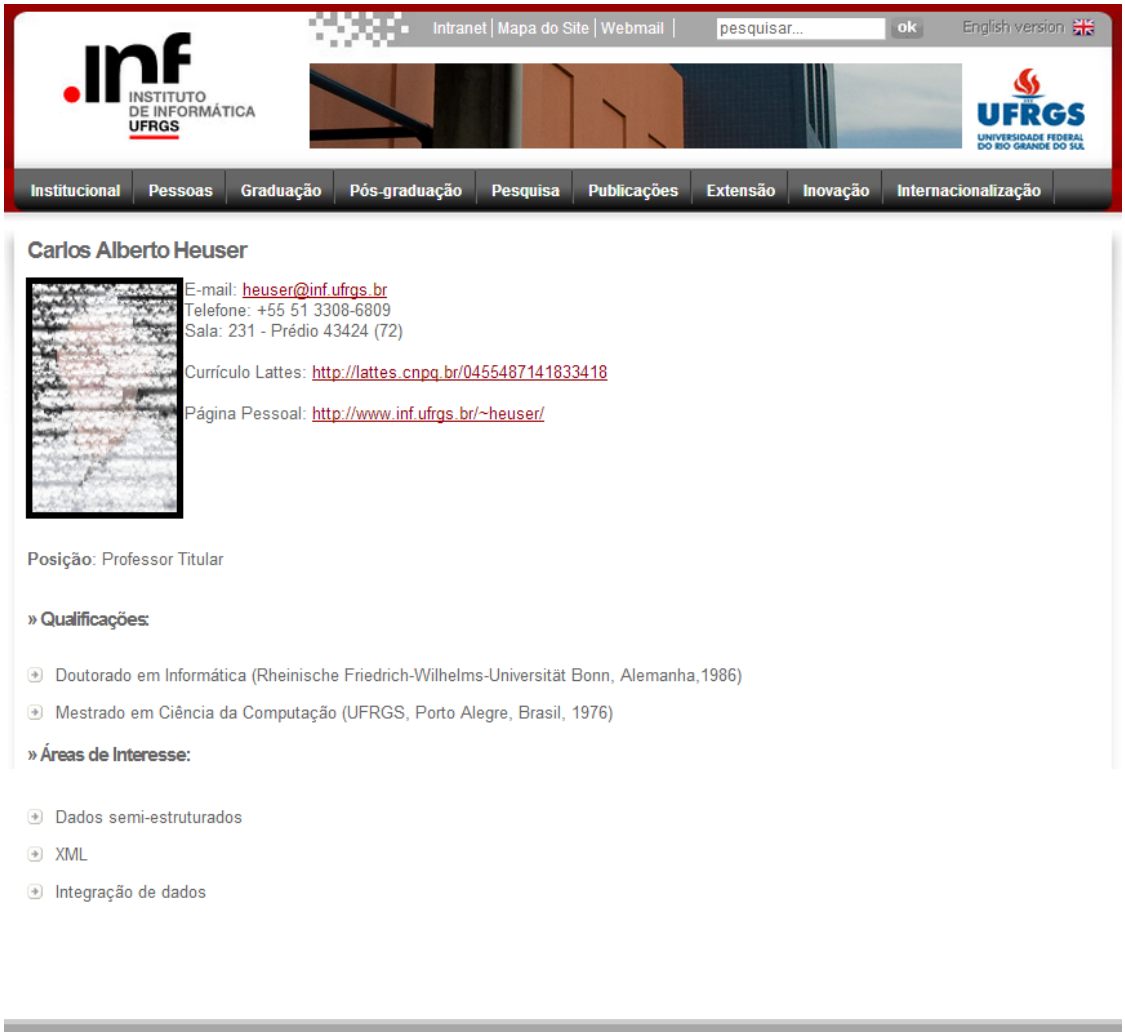

Figura 2.4: Exemplo de página-objeto para o domínio de professor/pesquisador.

que um objeto pertence a um domínio e que este, por sua vez, estabelece um conjunto de atributos para esse objeto.

Uma busca-objeto é representada através de um conjunto de restrições que são criadas sobre os atributos do domínio de um objeto. Como exemplos de buscas-objeto para os domínios de professor/pesquisador e *notebook*, respectivamente, pode-se citar: "`professor banco de dados UFRGS`" e "`notebook sony`". Na primeira consulta, são estabelecidas restrições sobre os valores dos atributos de área e instituição de atuação de um objeto professor e, na segunda, o valor do atributo marca é restringido para um objeto *notebook*.

O objetivo de buscas-objeto de um determinado domínio é obter como resposta um conjunto de páginas-objeto que satisfaça as restrições, estabelecidas pelo usuário, sobre os atributos desse domínio. Dessa forma, para a primeira consulta do exemplo anterior, espera-se recuperar páginas pessoais, institucionais, de currículo, dentre outras, que descrevam um objeto professor que possua banco de dados como área de interesse e que atue na UFRGS. Para a segunda, páginas de lojas virtuais e do fabricante, por exemplo, que descrevam um *notebook* da marca *sony* são bem-vindas.

Considerando a categorização de consultas apresentada na Subseção 2.1.2 e os exemplos de páginas-objeto apresentados na seção atual (Figuras 2.3 e 2.4), pode-se perceber

que a busca-objeto não pode ser condicionada a uma única categoria, uma vez que ela pode pertencer às três, de acordo com as necessidades do domínio de interesse. Aproveitando os exemplos de buscas-objeto mencionados anteriormente, a consulta "`professor banco de dados UFRGS`" do domínio de pesquisador, poderia pertencer ao tipo informacional, enquanto a consulta "`notebook sony`", do domínio de *notebook*, poderia pertencer ao tipo transacional, além do informacional.

PHAM et al. (2010) comparam o objetivo de buscas-objeto ao de um problema de aprendizagem de *ranking*, no qual se deseja aprender uma função capaz de ordenar qualquer par documento-consulta de um determinado domínio. Porém, esses autores também destacam algumas diferenças entre esses problemas. Os resultados de buscas-objeto, por exemplo, são "focados" na medida em que eles devem conter um objeto, o que se opõe à ampla noção de relevância de um problema de aprendizagem de *ranking* e da área de RI em geral, que atribuem um valor para uma determinada página de acordo com as características que esta página apresenta e de acordo com a similaridade que essas características possuem com o domínio ou a consulta em questão, sem considerar a funcionalidade da página (se ela é ou não uma página-objeto). Dessa forma, páginas relacionadas a notícias ou a concursos públicos para docentes, que não fazem parte do objetivo de buscas-objeto para o domínio de professor/pesquisador, podem ser tratadas como relevantes em um problema de aprendizagem de *ranking* ou um problema genérico de RI, caso essas páginas apresentem determinadas características. Essa situação foi ilustrada na Figura 1.1, apresentada no Capítulo 1, na qual pode-se perceber a dificuldade que os mecanismos de busca convencionais possuem para responder buscas-objeto, justamente por não possuírem esse objetivo.

2.3 Classificação de Páginas Web

PAGE et al. (1999) já faziam referência à necessidade de melhorar o processo de RI e desenvolver um sistema mais preciso, visto que, embora seja possível obter, em questão de segundos, referências a milhares de sites ordenados de forma que possivelmente os mais relevantes apareçam no topo, o usuário não necessariamente encontra o que procura.

Embora os GSEs tenham melhorado muito a qualidade de seus resultados, o que conquistou a confiança dos usuários e os transformou no principal meio de acesso à informação atualmente (MANNING; RAGHAVAN; SCHÜTZE, 2008), essa popularização também tornou maior o nível de exigência dos usuários e a especialidade das necessidades de informação. Com isso, a recuperação de resultados irrelevantes continua sendo um problema, principalmente para determinados tipos de consultas, como as de busca-objeto (PHAM et al., 2010). Parte das pesquisas que vem sendo realizadas com o intuito de melhorar os resultados de busca se concentram na classificação de páginas web.

A classificação (ou categorização) de páginas web faz parte do processo de mineração de conteúdo da web, que visa a extração de conhecimento a partir do conteúdo dos documentos e de seus metadados (descrição, autores, palavras-chave, etc.) (SANTOS, 2009), e consiste em atribuir uma página web a um ou mais rótulos de classe pré-definidos (QI; DAVISON, 2009). A classificação é muitas vezes representada como um problema de aprendizagem supervisionada (WITTEN; FRANK; HALL, 2011), em que um conjunto de dados rotulados é usado para treinar um classificador que, por sua vez, poderá ser usado para rotular futuras instâncias.

De acordo com QI; DAVISON (2009), dentre os subproblemas que fazem parte do problema geral de classificação de páginas web estão os de:

• Classificação de tópico – preocupa-se em identificar o tópico (assunto) da página, como, por exemplo, quando se deseja julgar se o conteúdo de uma página é sobre arte, negócio ou esporte;

• Classificação funcional (ou classificação de gênero (CHOI; YAO, 2005)) – preocupa-se em identificar o papel que uma página web desempenha, por exemplo, determinar se uma página é uma página pessoal ou uma página de curso;

• Classificação de sentimento – preocupa-se em identificar a opinião que é apresentada em uma página web, como descobrir o posicionamento de um autor perante um determinado assunto.

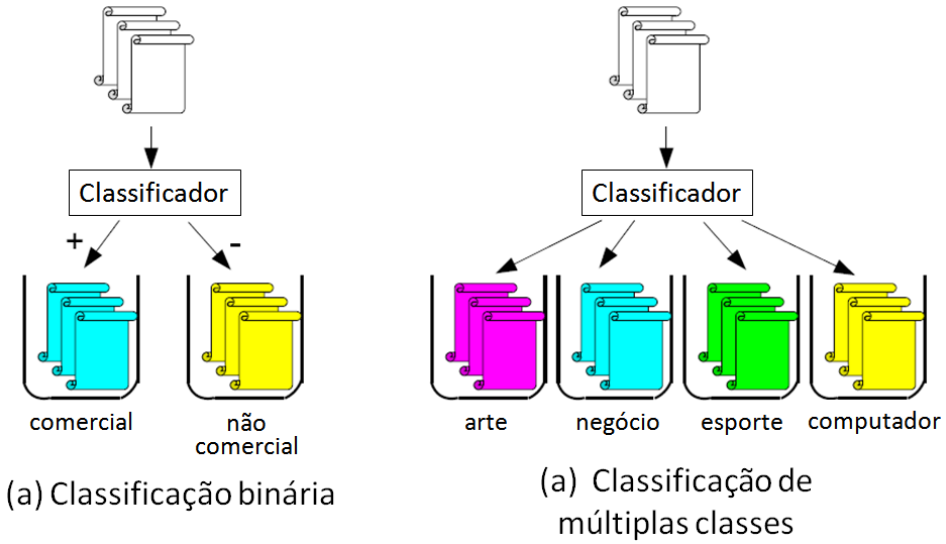

Figura 2.5: Classificação binária e de múltiplas classes [Adaptada de QI; DAVISON (2009)].

O problema de classificação também pode ser dividido, quanto ao número de classes usadas, em classificação binária (que considera apenas duas classes) e classificação de múltiplas classes (que considera mais de duas classes), como é ilustrado na Figura 2.5. Da mesmo forma, de acordo com o número de classes (rótulos) que podem ser atribuídas a uma instância, a classificação pode ser de único (uma instância só pode pertencer a uma única classe) ou múltiplos (uma instância pode pertencer a mais de uma classe) rótulos.

Uma dificuldade da classificação de páginas web é o fato de que os algoritmos existentes, usados no processo padrão de KDD (do inglês, *Knowledge Discovery in Databases*), pressupõem que os dados estejam estruturados de forma relacional, com os atributos dispostos em colunas e as instâncias de dados em linhas. Como os dados da web raramente seguem um padrão, exceto em tarefas de análise de *logs*, o esforço destinado a tarefas de pré-processamento dos dados acaba se tornando maior (SANTOS, 2009).

Nas subseções seguintes, são apresentadas as principais formas de representação de páginas web e o relacionamento entre a classificação e as áreas de busca na web e filtragem de informação.

2.3.1 Representação das Páginas Web

Há basicamente duas estratégias para a mineração de conteúdo: uma realiza a mineração diretamente do conteúdo dos documentos, que já foram recuperados e encontram-se

prontos para serem minerados, e a outra incrementa o poder de busca de outras ferramentas e serviços, ajudando no processo de indexação e categorização dos documentos, onde se insere os problemas de classificação de páginas (MARINHO; GIRARDI, 2003).

O conteúdo da web abrange uma ampla coleção de tipos de dados, variando desde texto e hipertexto até dados multimídia, como áudio e vídeo. Embora exista uma área de pesquisa que visa o estudo da mineração de dados multimídia, o foco da mineração de conteúdo ainda se concentra em dados textuais, visto que estes constituem grande parte do volume de dados da web (SANTOS, 2009). Dessa forma, muitos trabalhos consideram o problema de classificação de conteúdo de páginas web como um problema de classificação de texto, reutilizando as técnicas já consolidadas dessa área. Para isso, as páginas web devem ser pré-processadas de modo que suas características menos importantes sejam descartadas. Dentre as técnicas mais comuns de pré-processamento estão (CHOI; YAO, 2005):

- Remoção de *tags* HTML – consiste na eliminação das estruturas de marcação da linguagem HTML (do Inglês, *HyperText Markup Language*), mantendo apenas o texto a ser minerado. Nessa atividade, antes da efetiva remoção, as *tags* HTML podem ou não, dependendo da forma de representação adotada, ser usadas para ponderar a importância das palavras;

- *Tokenization* (Atomização) – divide o texto em unidades mínimas, geralmente através do reconhecimento de espaços em branco, tabulações e sinais de pontuação. Essas unidades mínimas são denominadas termos, que podem ser compostos tanto por uma única palavra (unigrama), quanto por várias palavras (bigramas, trigramas, ..., n-gramas);

- Remoção de *stop words* – remove palavras pouco significativas (que carregam pouca informação e possuem baixo poder discriminatório), como artigos, conjunções e preposições. Para isso, usa-se uma lista de *stop words* construída previamente;

- Remoção de palavras de baixa ou alta frequência – remove palavras com muita ou pouca ocorrência no texto, pressupondo que estas não são capazes de discriminar um documento;

- *Stemming* (radicalização) – objetiva reduzir as palavras a seus radicais, de forma que palavras de um mesmo campo com variações linguísticas, como plural ou gênero, passam a ser representadas da mesma forma.

As técnicas listadas anteriormente objetivam reduzir o léxico e normalizar os termos. Porém, para que esses dados textuais pré-processados se tornem legíveis aos algoritmos de classificação tradicionais, faz-se necessário adotar uma forma de representação para estruturá-los. A forma mais simples de representação consiste na conversão dos dados textuais em vetores de termos, por meio do modelo VSM, apresentado na Seção 2.1. A seguir, são apresentados métodos de ponderação que podem ser utilizados nessa representação.

2.3.1.1 TF-IDF

A principal forma de ponderação utilizada no processamento de texto é o TF-IDF (do inglês, *Term Frequency/Inverse Document Frequency*), que considera que documentos

que possuam termos (palavras) usados na consulta possuem uma grande probabilidade de serem interessantes ao usuário (CAMPOS; DIAS, 2005).

TF-IDF é uma medida para a atribuição de pesos usada na avaliação da importância de um termo para o documento em que ele ocorre, em relação a todos os documentos da coleção. Essa medida favorece termos encontrados várias vezes em um documento, mas que ocorram em poucos documentos de uma coleção, considerando que um termo que apareça em muitos documentos não pode ser usado para distinguir os objetos da coleção.

Seja *term frequence* de *t* ($tf_{t,d}$) igual ao número de vezes que o termo *t* ocorre em um documento *d*; *document frequence* de *t* (df_t) igual ao número de documentos que contêm o termo *t*; *N* igual ao número de documentos da coleção; e *inverse document frequence* de *t* (idf_t) igual a $\log \frac{N}{df_t}$; a fórmula conhecida como TF-IDF para a atribuição de pesos é dada pela combinação dessas duas medidas (MANNING; RAGHAVAN; SCHÜTZE, 2008), resultando em:

$$w_{t,d} = tf_{t,d} \times idf_t \tag{2.1}$$

$$w_{t,d} = tf_{t,d} \times \log \frac{N}{df_t} \tag{2.2}$$

2.3.1.2 SWT

Embora o conteúdo textual seja a característica mais fácil e direta de ser considerada, a presença de muito ruído em páginas web pode impedir que a ponderação com TF-IDF alcance um alto desempenho (QI; DAVISON, 2009). Na tentativa de melhor explorar os recursos dos documentos HTML, a estrutura dos documentos, representada pelas *tags* HTML, passou a ser considerada como um indicativo da localização dos termos mais significativos. Dessa forma, os termos contidos dentro das *tags* `<title>` e `</title>` passam a ser considerados mais importantes que os cercados pelas *tags* `<body>` e `</body>`, por exemplo (CHOI; YAO, 2005).

Para ponderar os termos (características) das páginas web, considerando as *tags* nos quais eles estão localizados, utiliza-se uma técnica de ponderação orientada à estrutura (em Inglês, *Structure-oriented Weighting Technique* – SWT), que atribui pesos maiores a termos cercados por *tags* consideradas mais importantes (expressivas) para a representação do conteúdo da página.

Seja e_k um elemento HTML, w_{e_k} o peso atribuído a esse elemento e_k, e $TF(t_j, e_k, d_i)$ o número de vezes que o termo t_j ocorre no elemento e_k da página HTML d_i, define-se a técnica SWT por meio da fórmula (RIBONI, 2002):

$$SWT_w(t_j, d_i) = \sum_{e_k} (w(e_k) \times TF(t_j, e_k, d_i)) \tag{2.3}$$

Na Tabela 2.1 são apresentados, como exemplo, os elementos HTML (*tags*) e os respectivos pesos utilizados por HRESKO (2012) na SWT. O autor atribui pesos diferenciados apenas para o título da página e para três níveis de títulos/subtítulos do corpo do texto, sendo que os termos contidos em outros elementos recebem o valor "1".

Embora o uso de *tags* HTML permita que sejam exploradas as informações estruturais (que geralmente são ignoradas, utilizando-se apenas o conteúdo textual), é importante considerar que a maioria das *tags* são orientados à representação e não à semântica. Isso permite que os autores de páginas web gerem diferentes estruturas de *tags* conceitualmente equivalentes, podendo essa inconsistência prejudicar a classificação (QI; DAVISON, 2009).

Tabela 2.1: Exemplo de elementos HTML e pesos para SWT.

Elemento (e_k)	Peso (w_{e_k})
title	10
h1	5
h2	3
h3	2
outro	1

2.3.2 Classificação e Busca na Web

Segundo QI; DAVISON (2009), a classificação desempenha um papel essencial em muitos sistemas de gerenciamento e tarefas de recuperação de informações, pois sua aplicação na categorização de conteúdos de páginas web, dentre outros benefícios, pode ajudar a melhorar a qualidade dos resultados recuperados pelas consultas. Os mesmos autores apresentam importantes aplicações da classificação de páginas web em problemas relacionados à busca, dentre as quais, encontram-se:

• construir, manter ou expandir diretórios da web (hierarquias web) – diretórios da web permitem a procura de informações dentro de um conjunto pré-definido de categorias de forma mais eficiente. Diretórios têm sido construídos e mantidos principalmente por editores, o que exige grande esforço humano e se torna menos viável com o crescimento e a mutabilidade da web. Nesse contexto, classificadores podem ser construídos para auxiliar na atualização e ampliação desses diretórios;

• melhorar a qualidade dos resultados de busca – a classificação das páginas de acordo com o assunto tratado por elas possibilita que buscas sejam feitas de forma a retornar apenas páginas pertencentes a uma determinada categoria de interesse. Isso evita que sejam recuperadas páginas de um assunto não desejado, apenas por estas conterem palavras-chave da consulta;

• ajudar sistemas de perguntas e respostas – sistemas de perguntas e respostas são sistemas onde usuários interagem uns com os outros, fazendo ou respondendo perguntas. Nesses sistemas, as perguntas e as respostas são armazenadas de forma categorizada para facilitar a recuperação em usos futuros. Sempre que um usuário faz um questionamento, o sistema primeiramente procura a resposta dentre as respostas já armazenadas e, caso não encontre, envia a pergunta para ser respondida por especialistas no assunto (ISKE; BOERSMA, 2005);

• construir motores de busca vertical – a classificação pode ser usada para avaliar a relevância de uma página web a um dado conjunto de tópicos de interesse, a fim de limitar a tarefa de *crawling*, permitindo a construção de motores de busca verticais, que se diferem dos motores de busca convencionais por serem focados em um domínio específico de conteúdo.

2.3.3 Classificação e Filtragem da Informação

Os motores de busca da web retornam, em geral, uma quantidade grande de informação e atribuem ao usuário a responsabilidade de analisar e selecionar o que é realmente de seu interesse. Essa tarefa de seleção consome muito tempo, principalmente pelo fato das páginas web, em geral, mesclarem informações relacionadas a diversos assuntos e

apresentarem anúncios publicitários, que fazem uso de imagens e vídeos para chamar a atenção do usuário (LOPEZ; SILVA, 2010). Na busca por reduzir o tempo que o usuário consome acessando e descartando essas informações, deu-se origem à área de pesquisa chamada de Filtragem de Informação (FI), que objetiva auxiliar os usuários a encontrarem informações que melhor atendam aos seus interesses, podendo também organizar e estruturar essas informações (PALME, 1998).

Para METEREN; SOMEREN (2000), sistemas de FI podem ser vistos como uma tarefa de classificação, onde o modelo de usuário é induzido através dos dados de treinamento, permitindo que o sistema de filtragem classifique itens desconhecidos como relevantes (classe positiva) ou irrelevantes (classe negativa) para o usuário. Nesse caso, o conjunto de treinamento consiste de itens que o usuário já achou interessante, sendo que cada item corresponde a uma instância de treinamento e todas as instâncias possuem um mesmo atributo, que especifica a classe do item. Dessa forma, a tarefa do algoritmo de aprendizagem é criar um modelo, com base no conjunto de treinamento, que possa classificar qualquer item da coleção, inclusive os novos e desconhecidos, como relevantes ou irrelevantes ao usuário.

WIVES (2002) também compartilha dessa percepção e afirma que FI possui os mesmos fundamentos da classificação, mas recebeu uma atenção e nomenclatura especial devido ao interesse da comunidade científica, na época, em desenvolver sistemas específicos de filtragem e recomendação de informações.

Por outro lado, HANANI; SHAPIRA; SHOVAL (2001), mesmo considerando o processo de classificação similar ao de FI, citam como característica discriminante o fato de os sistemas de classificação serem naturalmente estáticos, já que suas categorias não mudam com frequência, e os de FI serem dinâmicos, podendo existir mudanças frequentes nos usuários e nos perfis de usuários. Porém, é difícil encontrar trabalhos atuais que usem o termo “Filtragem de Informação”, em geral, outras nomenclaturas (como Classificação e Recomendação de Informação, dependendo do propósito do trabalho) passaram a ser utilizadas.

2.4 Considerações Finais

Neste capítulo foram apresentados conceitos relacionados a motores de busca convencionais da web e suas funções. Também foram abordadas definições a respeito de objetos da web, incluindo as de páginas-objeto e de buscas-objeto, que, em geral, não obtêm resultados satisfatórios quando executadas em motores de busca convencionais. Além disso, a classificação de páginas web foi apresentada, juntamente com técnicas de pré-processamento e representação de páginas web, que são necessárias para a aplicação de algoritmos de classificação. Os relacionamentos entre a classificação e as áreas de Busca na Web e de Filtragem de Informação também foram considerados, uma vez que muitos dos trabalhos relacionados, apresentados no próximo capítulo, fazem parte desse contexto.

Os conceitos abordados neste capítulo serão úteis para o entendimento do restante do texto desta dissertação, visto que o presente trabalho apresenta um método que utiliza técnicas de pré-processamento e classificação, baseadas no conteúdo de páginas web, para permitir a filtragem de páginas-objeto e, dessa forma, melhorar a precisão dos resultados recuperados pelos motores de busca convencionais para buscas-objeto.

3 TRABALHOS RELACIONADOS

Neste capítulo, apresenta-se, por meio de trabalhos relacionados, o contexto no qual se insere esta dissertação. Os trabalhos apresentados foram agrupados, de acordo com seus objetivos e abordagens, em três categorias, de modo a facilitar uma análise comparativa. Na Seção 3.1 são apresentados trabalhos relacionados à classificação e ao *ranking* de páginas web. Na Seção 3.2 são contempladas pesquisas envolvendo as áreas de busca vertical e coleta focada. E, por fim, na Seção 3.3 são abordados trabalhos que consideram os conceitos de entidades e objetos da web. Ao final de cada seção, apresenta-se um comparativo entre os trabalhos estudados, considerando as suas principais características.

3.1 Classificação e *Ranking* de Resultados

Muitos dos trabalhos que visam a melhoria dos resultados de busca se concentram na classificação e no *ranking* de páginas. A classificação, na maioria das vezes, objetiva retornar apenas as páginas classificadas como sendo do interesse do usuário, seja esse interesse relacionado ao assunto ou ao tipo funcional da página. O *ranking* visa a ordenação das páginas recuperadas por meio de diferentes critérios que possam beneficiar o posicionamento das páginas mais relevantes. Em geral, ambas as técnicas extraem um conjunto de características a partir do conteúdo das páginas web e consideram essas características no treinamento de um algoritmo de classificação ou na composição de uma função de *ranking*.

Motivados pelos problemas de respostas não distintas e baixas taxas de revocação[1] obtidos pela maioria dos sistemas de resposta à questões de lista (onde espera-se um conjunto de entidades distintas como resposta, como, para a consulta "*nome de marcas de chocolate Belga*"), YANG; CHUA (2004) propuseram uma abordagem que utiliza classificação funcional de páginas web para encontrar respostas para esse tipo de pergunta. Nessa abordagem, para cada questão de lista é formulado, de acordo com padrões de heurísticas, um conjunto de consultas, que são submetidas a motores de busca da web (Google, Altavista e Yahoo). As páginas resultantes são recuperadas e representadas por um conjunto de características previamente definido; sendo, então, classificadas por meio de classificadores C4.5 (de árvores de decisão), treinados com uma base extraída do TREC (do Inglês, *Text REtrieval Conference*), entre as seguintes categorias: `(I) páginas de coleção`, que apresentam uma lista de possíveis respostas; `(II) páginas de tópico`, que melhor representam uma resposta; `(III) páginas relevantes`, que são relevantes para as páginas de tópico; e `(IV) páginas irrelevantes`. Após, é feita uma navegação por meio dos *links* contidos nas páginas de coleção a fim

[1] proporção dos documentos relevantes que foram retornados como resultado para uma consulta.

de coletar novas páginas de tópico, aumentando a cobertura dos resultados. As páginas de tópico são, então, agrupadas por similaridade, a fim de eliminar respostas redundantes, sendo extraídas as respostas a partir desses grupos. Com essa metodologia de classificar as páginas de acordo com suas funcionalidades e, posteriormente, agrupar as páginas de tópico resultantes, o sistema consegue eliminar a maioria das respostas redundantes, aumentando as chances de encontrar respostas distintas na web, que é uma das dificuldades da maioria dos sistemas de resposta a perguntas do estado da arte.

Em RAJAN et al. (2010) é apresentado um sistema de categorização de tópicos que foi implantado em uma plataforma de correspondência de anúncios na web. O sistema foi projetado para detectar páginas sobre uma série de temas sensíveis – como páginas relacionadas a conteúdo adulto, jogos de azar, notícias difíceis (sobre mortes e sofrimento, por exemplo) e drogas – visto que, em geral, os anunciantes não querem que seus anúncios estejam associados a páginas dessa natureza. O sistema de categorização proposto usa um classificador binário para cada tema de interesse. Os classificadores são baseados em aprendizagem ativa, na qual o conjunto de treinamento aumenta iterativamente por meio de um número limitado de exemplos de toda a população. Inicialmente, para gerar dados de treinamento, o desenvolvedor submete consultas (como "`cassinos online`"), que provavelmente recuperarão resultados sobre a classe desejada, a um motor de busca, rotulando e coletando exemplos positivos e negativos dentre as páginas recuperadas. As páginas rotuladas são representadas por meio do modelo VSM, considerando unigramas; *stop words* são removidas; e um algoritmo de classificação de Máquina de Vetores de Suporte (em inglês, *Support Vector Machine* – SVM) com núcleo linear é, então, aplicado para gerar o modelo de classificação. Após, o modelo gerado é avaliado sobre dados de teste, podendo ser feitos ajustes sobre o modelo, através da mudança de parâmetros. Embora este sistema de categorização não seja aplicado diretamente à busca na web, as técnicas utilizadas na coleta de páginas de treinamento e na classificação baseada em conteúdo são do interesse desta dissertação.

Muitos dos trabalhos relacionados a *ranking* constroem funções de *ranking* específicas a determinados domínios (ZHENG et al., 2007) (PHAM et al., 2010) ou propõem modelos de *ranking* adaptativos (ZAREH BIDOKI et al., 2010) (NAKATANI; JATOWT; TANAKA, 2010) (WANG et al., 2013), que se adéquam a diferentes domínios e usuários, reduzindo o esforço para a construção de diversas funções separadamente. Porém, como o interesse dessa dissertação se concentra mais na classificação de páginas web, optou-se por apresentar trabalhos que utilizam a classificação para melhorar o processo de *ranking*. Neste contexto, YAMAMOTO; NAKAMURA; TANAKA (2009) desenvolveram um sistema que suporta tarefas de busca exaustiva de páginas web ou publicações, através de operações de edição, utilizadas tanto para classificar quanto para reordenar os resultados de busca. Uma tarefa de busca exaustiva ocorre quando o usuário precisa analisar muitos resultados de busca para alcançar seu objetivo, como quando pesquisa-se por publicações relacionadas a um tema de pesquisa e todos os resultados recuperados pela busca devem ser conferidos. Na abordagem proposta, o usuário submete uma consulta ao sistema, que a encaminha para um motor de busca (Google e GoogleScholar são usados como motores de busca web e de publicações, respectivamente). Após, o sistema recebe os resultados do motor de busca e os apresenta ao usuário, que, ao analisar os resultados, pode usar três operações: `(I) excluir ou dar ênfase a termos`, com o objetivo de melhorar a precisão dos resultados de pesquisa posicionados no topo; `(II) arrastar e soltar`, para categorizar, em diretórios, os resultados da busca; e `(III) adicionar uma nova consulta ao sistema`, a fim de obter ou-

tros resultados de busca. Quando a primeira operação é usada, os resultados da consulta são reordenados, de forma que os que contenham o termo excluído/enfatizado sejam prejudicados/beneficiados no *ranking*. A classificação dos resultados de busca a partir da segunda operação pode apresentar diferentes situações: o usuário pode arrastar e soltar um termo em um diretório, a fim de adicionar os resultados de busca que contêm esse termo ao diretório; o usuário pode arrastar e soltar um resultado de busca em um diretório, a fim de incluí-lo, assim como resultados semelhantes, no diretório; e o usuário pode arrastar e soltar um diretório, contendo um conjunto de resultados de consulta já classificados, em outro, sendo os resultados de ambos armazenados no diretório destino. O sistema proposto permite a interação do usuário, que, por meio de operações de edição, pode classificar e reordenar os resultados de busca, facilitando a navegação entre os resultados, o que é muito útil em busca exaustiva.

BENNETT; SVORE; DUMAIS (2010) propõem uma abordagem que usa a classificação de tópicos para melhorar o *ranking* dos resultados recuperados em uma busca. Essa proposta considera que um resultado retornado por uma consulta e selecionado pelo usuário é o tipo de resultado desejado, sendo que determinar a classe a que esse resultado pertence é uma forma de permitir que uma nova ordenação seja realizada, beneficiando outros resultados da mesma classe. Para determinar a classe de um resultado, foram, primeiramente, coletadas páginas do ODP (do inglês, *Open Directory Project*) (NETSCAPE, 2013), distribuídas em 219 categorias. Para cada uma dessas categorias, foi treinado um classificador de regressão logística com base no conteúdo extraído dessas páginas, considerando técnicas padrões de classificação de texto. O sistema é constituído de dois componentes principais, sendo o primeiro um método que usa dados de cliques para derivar as distribuições de classe da consulta a partir das distribuições de classe de cada documento resultante. No segundo, ao invés de tentar estabelecer um peso às correspondências entre distribuições de classes de consulta e de documento, é introduzido um conjunto de características que capturam, dentre outros fatores, importantes propriedades dessas duas distribuições. Esse conjunto de características serve, então, como entrada para um algoritmo de *ranking*, que aprende os pesos durante o treinamento e os usa para fazer previsões durante o processo de recuperação. Dessa forma, o sistema define a classe de consulta por meio dos resultados relevantes à consulta (estimados a partir do comportamento de clique) e realiza uma generalização ao longo da dimensão da classe, de modo a identificar e melhor posicionar outros resultados relevantes.

Embora não seja possível comparar os resultados obtidos por esses trabalhos, visto que eles possuem objetivos e dados distintos, na Tabela 3.1 é apresentada uma síntese desses trabalhos, considerando os seguintes itens:

- `Objetivo` – refere-se ao propósito geral do trabalho;
- `Abordagem` – indica se o trabalho adota classificação de páginas puramente ou em conjunto com técnicas de *ranking*;
- `Base de treino` – origem dos dados de treinamento;
- `Representação` – forma utilizada para representar as páginas a fim de prepará-las para a aplicação do algoritmo de classificação;
- `Algoritmo` – algoritmo de classificação utilizado.

Pode-se notar que a obtenção de dados de treinamento, em geral, conta com a contribuição direta do usuário ou com a existência de bases de conhecimento previamente

Tabela 3.1: Comparativo entre os trabalhos de classificação e *ranking* de páginas web.

	Objetivo	**Abordagem**	**Base de Treino**	**Representação**	**Algoritmo**
YANG; CHUA (2004)	encontrar respostas completas e distintas para questões de lista	classificação	questões de lista extraídas do TREC	conjunto de características definido manualmente	C4.5, de árvores de decisão
RAJAN et al. (2010)	detectar páginas de uma série de temas sensíveis	classificação	páginas rotuladas pelo usuário	*bag of words*, considerando unigramas e remoção de *stop words*	SVM com núcleo linear
YAMAMOTO; NAKAMURA; TANAKA (2009)	facilitar a navegação entre os resultados de busca exaustiva	classificação e *ranking*	páginas editadas pelo usuário	-	-
BENNETT; SVORE; DUMAIS (2010)	melhorar o *ranking* dos resultados de busca por meio de classificação de tópicos	classificação e *ranking*	páginas e categorias extraídas do ODP	técnicas padrões de processamento de texto (remete ao uso de *bag of words*)	um algoritmo de regressão logística

rotuladas. A adoção de bases de conhecimento previamente construídas é vantajosa à medida que reduz o esforço do usuário, porém, ela também limita a representatividade dos dados, uma vez que esses repositórios não apresentam uma grande cobertura em relação as várias formas de representação das entidades (KAPTEIN et al., 2010). Por isso, a contribuição do usuário ao indicar a relevância de determinados resultados é uma prática (chamada de realimentação de relevância (MANNING; RAGHAVAN; SCHÜTZE, 2008)) que tem sido bastante explorada atualmente para fomentar a melhoria dos resultados de busca, permitindo que o usuário influencie diretamente no processo de busca, julgando os itens que considera necessário.

Quanto à representação das páginas, a extração de um conjunto de características é inevitável, porém, esse conjunto pode ser previamente definido pelo desenvolvedor da aplicação ou automaticamente extraído. A extração automática pode comprometer o desempenho do sistema, uma vez que nem sempre vai considerar as melhores características para o domínio, porém, também é importante considerar que a definição prévia de um conjunto de características depende do conhecimento e da invariabilidade do domínio da aplicação. O domínio da aplicação também interfere na escolha do algoritmo, que, como mostram os trabalhos apresentados, pode variar bastante.

3.2 Busca Vertical e Coleta Focada

Como mencionado na Seção 2.3, a popularização dos motores de busca vem fazendo com que, cada vez mais, os usuários apresentem necessidades de informações refinadas. Segundo KANG et al. (2012), uma das tendências nesse contexto é a busca vertical. Motores de busca verticais são motores específicos a determinados domínios, como o Google Scholar[2] e o Amazon[3] que concentram suas buscas nos domínios acadêmico e de

[2]http://www.scholar.google.com
[3]http://www.amazon.com

produtos, respectivamente. Para construir e atualizar o índice de um motor vertical, considerando apenas páginas do domínio de interesse, usam-se coletores focados, que, como o próprio nome sugere, são especializados na coleta de páginas relacionadas a domínios específicos. Em geral, esses coletores utilizam classificadores baseados em conteúdo para filtrar apenas páginas relevantes ao domínio considerado.

Em JI et al. (2009) é proposto um motor de busca vertical, chamado ExSearch, para o domínio de negócios de troca on-line. O ExSearch é constituído por cinco módulos. Primeiro, uma `coleta focada` é realizada por meio do *framework* proposto por CHAKRABARTI; BERG; DOM (1999), iniciando pela seleção manual (por parte do usuário) de um conjunto de URLs (do Inglês, *Uniform Resource Locator*) sementes, que correspondem a sites comerciais ou de grupos de notícias relacionados à troca de produtos. As páginas desses sites são representadas por *bag of words* e analisadas por um classificador *Naive Bayes* (treinado previamente por meio de exemplos positivos e negativos fornecidos pelo usuário), sendo descartadas as páginas consideradas irrelevantes para o domínio de troca. Quando uma página é classificada como relevante, ela é armazenada e seus *hyperlinks* são adicionados à fila de URLs candidatas, continuando o processo iterativamente até que algum critério de parada seja atingido. No segundo módulo, um algoritmo de `extração de informação` baseado em regras é aplicado sobre as páginas coletadas a fim de extrair e construir uma base de dados estruturada com as informações de troca. Para preencher dados perdidos sobre as informações de troca, os objetos de troca são `classificados` (terceiro módulo) em categorias de produtos previamente definidas e os valores perdidos são estimados por regressão. Após, os dados passam por um processo de `indexação` (quarto módulo), de modo a serem posteriormente `recuperados` (quinto módulo) por consultas de usuários.

BLANCO et al. (2008) propõem um método para coletar automaticamente páginas da web que publicam dados relacionados a instâncias 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, por meio de um algoritmo chamado `INDESIT`, cada um desses sites procurando por páginas que apresentem uma lista de *links* que direcionam para páginas com estruturas similares aos respectivos exemplos. A estrutura de uma página é descrita por meio das propriedades de apresentação e disposição dos *links* que ela possui, sendo a semelhança estrutural entre páginas obtida por meio dessas características. O `INDESIT` se baseia, principalmente, na observação de que, quando a grande maioria dos *links* de duas páginas compartilham as mesmas propriedades de apresentação e disposição, é provável que as duas páginas possuam a mesma estrutura e, dessa forma, tratem de instâncias da mesma entidade conceitual. Após essa busca dentro dos sites dos exemplos de entrada, as páginas de entidades coletadas são usadas para estender a procura para a web, por meio de um algoritmo chamado `OUTDESIT`, com o objetivo de encontrar outras fontes que forneçam instâncias da mesma entidade. Para isso, são propagadas buscas na web para cada uma das páginas coletadas pelo `INDESIT`, sendo as palavras-chave dessas consultas constituídas pelas âncoras dos *links* que apontavam para essas páginas, além de outros termos que atuem como descritores do domínio e da entidade conceitual. As páginas resultantes dessa busca são analisadas pelo `OUTDESIT`, de modo que só as páginas que apresentam os termos descritores da entidade localizados na estrutura da página são consideradas instâncias da entidade, podendo estas, então, servirem de entrada para o `INDESIT`, reiniciando o processo de coleta. Esse método objetiva alimentar e permitir a criação de motores de busca verticais sobre entidades conceituais, uma vez que restringe a coleta a esse tipo de página.

Também com relação à coleta de páginas, ASSIS et al. (2009) propõem 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 considerados, separadamente, para julgar se uma determinada página é (ou não) relevante para o tópico de interesse. Isso não é possível em coletores guiados por classificadores, uma vez que eles não conseguem separar os aspectos de gênero e conteúdo. Basicamente, a abordagem proposta assume que o usuário deve informar um conjunto de termos que descreva o gênero e outro conjunto que descreva o conteúdo do tópico desejado, além de um conjunto de páginas sementes. O processo de coleta é iniciado a partir das páginas sementes, sendo que a relevância de cada página é obtida a partir de uma combinação dos pesos obtidos pelos cálculos de similaridade que essa página apresenta em relação ao conjunto de termos de gênero e ao conjunto de termos de conteúdo. Esses cálculos são feitos por meio da similaridade de cosseno, considerando TF-IDF. As páginas que atenderem a um determinado limiar de similaridade são coletadas e suas URLs são adicionadas à fila de candidatas, dando prosseguimento ao processo.

Na Tabela 3.2 é apresentado um comparativo dos trabalhos apresentados nesta seção, considerando os seguintes itens:

- `Objetivo` – refere-se ao propósito geral do trabalho;
- `Abordagem` – indica se o trabalho propõe a busca vertical (incluindo a coleta focada) ou apenas a coleta focada;
- `Participação do Usuário na Coleta` – atividades que dependem da participação do usuário durante o processo de coleta;
- `Representação` – forma utilizada para representar as páginas de modo a permitir, posteriormente, a análise de relevância;
- `Indicador de Relevância` – forma utilizada para definir a relevância das páginas em relação ao domínio de interesse.

Por meio da Tabela 3.2, pode-se perceber que a participação do usuário é essencial para o processo de coleta, sendo que, no mínimo, ele precisa informar um conjunto de páginas sementes. No trabalho de ASSIS et al. (2009) o esforço e a especialização do usuário são maiores, uma vez que, além das páginas sementes, ele precisa informar um conjunto de termos que descreva o gênero e outro que descreva o conteúdo das páginas desejadas, o que requer mais conhecimento do usuário do que apenas indicar exemplos de páginas desejadas.

Quanto à forma de representação adotada para as páginas, considerá-las por meio de seus termos (*bag of words*) mostra-se comum, seja para a posterior aplicação de um classificador ou para a análise de similaridade, que são os principais indicativos de relevância utilizados. Também é importante observar que no trabalho de BLANCO et al. (2008), embora a relevância seja indicada por meio de similaridade, esta é medida em relação à disposição e apresentação dos links (estrutura), considerando um comportamento específico das páginas do domínio de entidades, e não por meio dos termos (conteúdo) das páginas, o que limita o método proposto a esse tipo de aplicação.

Tabela 3.2: Comparativo entre os trabalhos de busca vertical e coleta focada.

	Objetivo	**Abordagem**	**Participação do Usuário na Coleta**	**Representação**	**Indicador de Relevância**
JI et al. (2009)	possibilitar busca no domínio de trocas	busca vertical	páginas sementes e base de treino do classificador	*bag of words*	classificador *Naive Bayes*
BLANCO et al. (2008)	coletar páginas que representam instâncias de entidades	coletor focado	páginas sementes	propriedades de apresentação e disposição dos *links*	similaridade
ASSIS et al. (2009)	coletar páginas de tópicos que possam ser presentados pelos aspectos de gênero e conteúdo	coletor focado	páginas sementes, termos de gênero e termos de conteúdo	*bag of words*, com TF-IDF	combinação das similaridades de cosseno

3.3 Entidades e Objetos Web

Nesta seção, são apresentados trabalhos que visam a busca por entidades ou objetos da web. Muitos desses esforços consideram a Web Semântica (em Inglês, *Semantic Web* – SW) (BERNERS-LEE; HENDLER; LASSILA, 2001), que, embora proponha soluções para a maioria dos problemas de busca na web, ainda traz poucas contribuições nesse sentindo devido à escassez de páginas publicadas com anotações semânticas (BLANCO et al., 2008).

MIKLÓS et al. (2010) propõem o *Entity Name System* (ENS), que provê um serviço de identificação de entidades global para permitir a chamada web de entidades. As palavras que apontam para entidades nomeadas (como pessoas, localizações geográficas, etc.) são anotadas com uma referência a uma descrição dessa entidade. O serviço de identificação representa uma entidade por meio de um conjunto de pares atributo-valor que descrevem essa entidade, bem como através de informações relacionadas a sua evolução e um identificador único. Os usuários podem requisitar um identificador de entidade por meio de palavras-chave ou de pares atributo-valor, como "`Paris`" ou "`first name = Paris`" respectivamente. Para isso, o armazenamento de entidades é realizado em dois níveis: o armazenamento de pares atributo-valor e um índice invertido, no qual a lista de *postings* inclui, além do identificador do documento, informações dos atributos onde os termos ocorrem. O índice invertido é usado para encontrar os documentos relevantes para a consulta, enquanto o repositório atributo-valor contém os perfis das entidades propriamente. O processo de busca no ENS é dividido em duas fases: primeiro, um conjunto de entidades candidatas é selecionado (por meio do índice invertido) e, após, elas são ordenadas de acordo com uma ponderação que agrega várias características dos níveis de atributo (como similaridade de rótulo e similaridade de valor do atributo) e de entidade (como similaridade de entidade e popularidade de entidade). Dessa forma, a principal funcionalidade do ENS é processar solicitações de busca por entidades e retornar um único identificador para essa entidade. Assim, é possível anotar a palavra-chave ou a página com esse identificador único, evitando ambiguidades.

Em CHENG; QU (2008), é apresentado um motor de busca, chamado Falcons, que provê a busca por palavras-chave de objetos da SW, descritos em RDF (do Inglês, *Re-*

source Description Framework) (KLYNE; CARROLL, 2004). Para cada objeto, o sistema constrói um documento virtual (representado no modelo VSM, considerando *idf* na ponderação dos termos) que consiste em descrições textuais extraídas de sua descrição RDF. Esse documento inclui não só nome local, rótulos e outros literais associados, como também descrições textuais de todos os outros recursos vizinhos. A partir dos termos desses documentos virtuais, é construído um índice invertido para os objetos, que servirá à busca por palavras-chave. Também é extraído o tipo (classe), explicitamente especificado no RDF, de cada objeto, sendo construído um outro índice invertido, permitindo um mapeamento classes–objetos. Combinando os dois índices, o sistema permite que o usuário, além de submeter suas consultas por palavras-chave, estabeleça restrições sobre a classe dos objetos desejados. Dessa forma, dada a submissão de uma consulta, o sistema, baseado nos índices, recupera um conjunto de objetos cujas descrições textuais contêm todos os termos da consulta e que pertençam à classe desejada. Esses objetos resultantes são, então, ordenados de acordo com critérios de relevância para a consulta (com base na similaridade de cosseno) e de popularidade (número de documentos RDF que usam os objetos). Como, muitas vezes, apenas uma classe é informada na descrição RDF de um objeto e o usuário pode considerar uma superclasse ao restringir sua consulta, o sistema realiza um raciocínio de inclusão de classe, mapeando para esse objeto também as classes inferidas. Dessa forma, uma hierarquia de classes, de navegação amigável, é oferecida aos usuários, permitindo que as consultas sejam incrementalmente refinadas. O principal diferencial desse trabalho, em relação a outros motores de busca na SW, encontra-se na inferência de classes, que, em geral, não é realizada devido sua complexidade computacional.

Considerando as limitações da real aplicação da SW, PHAM et al. (2010) propõem o *Object Search Engine* (OSE), que trata o problema de busca por páginas-objeto de forma similar aos problemas de aprendizagem de funções de *ranking* tradicionais de RI, em que o principal objetivo é aprender uma função de *ranking* por meio de aprendizagem de máquina e de um conjunto de características relevantes. A solução proposta consiste em desenvolver diversos motores de busca verticais para permitir a busca por páginas-objeto em diferentes domínios. Para isso, uma função de *ranking* deve ser treinada para cada domínio específico. O usuário (desenvolvedor) deve submeter consultas por palavras-chave e anotar (indicar se algumas páginas são ou não páginas-objeto para o domínio considerado) um conjunto de treinamento inicial a partir das páginas recuperadas. Esse conjunto de treinamento tem suas características extraídas automaticamente, por meio de um método chamado *Feature Generation Function*, pertencente à biblioteca LBJ (do Inglês, *Learning Based Java*). O usuário pode inserir novas características de forma manual, além de escolher um subconjunto a partir das que foram extraídas automaticamente, de modo a indicar quais características devem ser consideradas pela função de *ranking*. Na função de *ranking*, é usado um vetor de pesos aprendidos por meio do algoritmo de aprendizagem de máquina *Averaged Perceptron*, que é treinado a partir de vetores obtidos com a aplicação das funções características aos pares documento-consulta. Esse processo pode continuar iterativamente até que a função seja considerada satisfatória pelo usuário, ou seja, até que ela seja capaz de calcular adequadamente a probabilidade de uma determinada página conter um objeto de interesse. Assim, outros usuários, além do que guiou o treinamento da função de *ranking*), podem realizar buscas-objeto no domínio já aprendido. Esse trabalho, embora exija um esforço humano considerável no treinamento, tem a vantagem de não depender de páginas anotadas semanticamente (ainda bastante escassas na web), o que o torna mais próximo ao trabalho dessa dissertação e, dessa forma,

o capacita a ser usado como *baseline* nos experimentos apresentados no Capítulo 5.

Na Tabela 3.3 é apresentado um comparativo dos trabalhos apresentados nesta seção, considerando os seguintes itens:

- `Objetivo` – refere-se ao propósito geral do trabalho;
- `Abordagem` – indica a solução adotada para atingir o objetivo;
- `Participação do Usuário na Coleta` – indica quais atividades dependem da participação do usuário na construção do sistema;
- `Representação` – forma utilizada para representar as páginas/objetos/entidades para a posterior análise de relevância;
- `Indicador de Relevância` – forma utilizada para definir a relevância das páginas/objetos/entidades.

Tabela 3.3: Comparativo entre os trabalhos relacionados a entidades e objetos web.

	Objetivo	**Abordagem**	**Participação do Usuário**	**Representação**	**Indicador de Relevância**
MIKLÓS et al. (2010)	prover a identificação e a busca de entidades	uso de uma estrutura de índice diferenciada, através do qual entidades candidatas são recuperadas e, então, ordenadas de acordo com os indicadores de relevância	nenhuma	cada entidade é representada por um conjunto de pares atributo-valor, um conjunto de meta-dados relacionado a sua evolução e um identificador único	ponderação que agrega características dos níveis de atributo e entidade, estando algumas dessas características relacionadas à similaridade
CHENG; QU (2008)	prover um motor de busca por palavras-chave para objetos da web semântica	criação de índices invertidos para as descrições textuais e de classe (explícitas ou inferidas) dos objetos e ordenação de acordo com os indicadores de relevância	nenhuma	VSM com *idf*, construído a partir das descrições textuais extraídas do RDF de cada objeto	similaridade de cosseno (relevância para a consulta) e popularidade do objeto
PHAM et al. (2010)	prover a construção de motores de busca verticais para páginas-objeto	aprendizagem de uma função de *ranking* para cada domínio	páginas de treinamento e seleção de características	características extraídas automaticamente	função de *ranking*

Considerando a Tabela 3.3, pode-se perceber que o problema de busca de objetos ou entidades na web semântica, em geral, é tratado por meio da construção de sistemas de busca que realizem a indexação e a ordenação desses objetos/entidades. Nesses casos (aqui exemplificados pelos dois primeiros trabalhos), o fato de os documentos já possuírem anotações semânticas torna desnecessária a participação do usuário na realização dessas tarefas. Porém, faz-se necessário observar que essas anotações foram realizadas

previamente por meio de um esforço humano. Em geral, na representação, são consideradas as descrições textuais dos documentos semânticos e, como indicadores de relevância, são considerados critérios como similaridade e popularidade.

A abordagem proposta por PHAM et al. (2010) visa tratar a busca por objetos na web atual (sintática), considerando que um objeto é representado por uma página-objeto (cujo conceito foi apresentado na Subseção 2.2.1), a qual consiste no objetivo da busca. Considerando que um objeto possui um domínio e um conjunto de atributos, que não são explicitamente indicados nas páginas web, a participação do usuário é necessária para permitir a criação das funções de *ranking*, que determinarão a relevância das páginas. Embora esse trabalho necessite de um esforço humano considerável, pode-se dizer que ele se adéqua melhor a realidade atual da web, visto que páginas anotadas semanticamente ainda são bastante escassas.

3.4 Considerações Finais

Neste capítulo, foram apresentados os principais trabalhos relacionados a esta dissertação. Embora um agrupamento, de acordo com objetivos e abordagens, tenha sido considerado para facilitar a análise comparativa, faz-se importante destacar que alguns trabalhos poderiam ser considerados como parte de mais de um grupo. O trabalho que considera a criação de funções de *ranking* para tratar o problema de busca por páginas-objeto (PHAM et al., 2010), citado no grupo "Entidades e Objetos Web" (Seção 3.3), também poderia ter sido apresentado no grupo "Classificação e *Ranking* de Resultados" (Seção 3.1), por exemplo.

Nesta dissertação, é abordado o problema de busca por objetos, assim como nos trabalhos da Seção 3.3, porém, considerando as limitações da real aplicação da web semântica, é buscada uma solução aplicável à web atual (sintática). Para isso, são adotadas técnicas de classificação de páginas web, como nos trabalhos da Seção 3.1, de modo a construir modelos de classificação que permitam a filtragem de páginas-objeto de domínios específicos, assemelhando-se à criação de diversos motores de busca verticais (Seção 3.2). O trabalho de PHAM et al. (2010) adota uma abordagem parecida, considerando, porém, técnicas de aprendizagem de *ranking*, cujo desempenho depende das características utilizadas. Embora os autores considerem uma função para extrair automaticamente essas características, uma intervenção adicional do usuário (além da participação na rotulação de um conjunto de páginas de treinamento) pode ser necessária, de modo a adicionar ou descartar algumas características que ele, com conhecimento prévio de domínio, considere estar prejudicando o *ranking*. No presente trabalho, busca-se eliminar esse esforço adicional por meio da adoção da classificação baseada em conteúdo de páginas-web.

4 OPIS: OBJECT PAGE IDENTIFYING AND SEARCHING

Neste capítulo, é proposto um novo método para a identificação e a busca de páginas-objeto, denominado OPIS (acrônimo para ***O**bject **P**age **I**dentifying and **S**earching*). O OPIS permite que somente páginas identificadas como páginas-objeto, considerando um domínio específico, sejam apresentadas para o usuário em resposta as suas consultas por palavras-chave. Dessa forma, os resultados de buscas-objeto se tornam mais precisos, atendendo de forma mais satisfatória às necessidades de informação dos usuários. Na Seção 4.1 é apresentada uma visão geral do OPIS, mostrando a integração entre as atividades de identificação e busca, as quais são abordadas de forma mais detalhada nas Seções 4.2 e 4.3, respectivamente.

4.1 Visão Geral

O OPIS é um método que visa a identificação e a busca de páginas-objeto, de modo a tornar os resultados de buscas-objeto mais precisos e adequados às necessidades de informação dos usuários. O cerne do OPIS se concentra na adoção de técnicas de pré-processamento de texto e de aprendizagem de máquina no treinamento de um modelo de classificação, baseado no conteúdo das páginas web, responsável por identificar páginas-objeto de um domínio de interesse. Após treinado, o classificador é integrado a um motor de busca convencional, de modo que uma etapa adicional de filtragem (classificação) seja considerada no processo de busca.

Na Figura 4.1 é apresetada uma visão geral do OPIS, mostrando a integração das atividades de identificação e busca. Note que o usuário submete uma consulta por palavras-chave na interface de busca. Essa consulta é executada no índice e obtém como resposta o *ranking* das páginas nas quais os termos da consulta foram encontrados, seguindo, até então, o processo de busca convencional. A diferença introduzida pelo OPIS (ilustrada por meio de segmentos tracejados) se encontra no fato de que o *ranking* de páginas não é apresentado diretamente ao usuário, passando previamente por uma atividade adicional de filtragem. Nessa etapa, as páginas são submetidas a um processo de classificação e somente as que forem classificadas como sendo páginas-objeto são apresentadas ao usuário.

O classificador é o núcleo do OPIS, uma vez que é o responsável pela tarefa de identificação das páginas-objeto, da qual depende a filtragem e, assim, os resultados de busca. A construção de um classificador, além da seleção e parametrização de um algoritmo de classificação, depende de um conjunto inicial de páginas de treinamento. Esse conjunto é a base do processo de aprendizagem do algoritmo (WITTEN; FRANK; HALL, 2011) e, nesse caso, consiste de exemplos positivos e negativos de páginas-objeto, considerando um domínio específico. A fim de obter essas páginas, o OPIS adota um processo de realimentação de relevância (MANNING; RAGHAVAN; SCHÜTZE, 2008), no qual o usuário

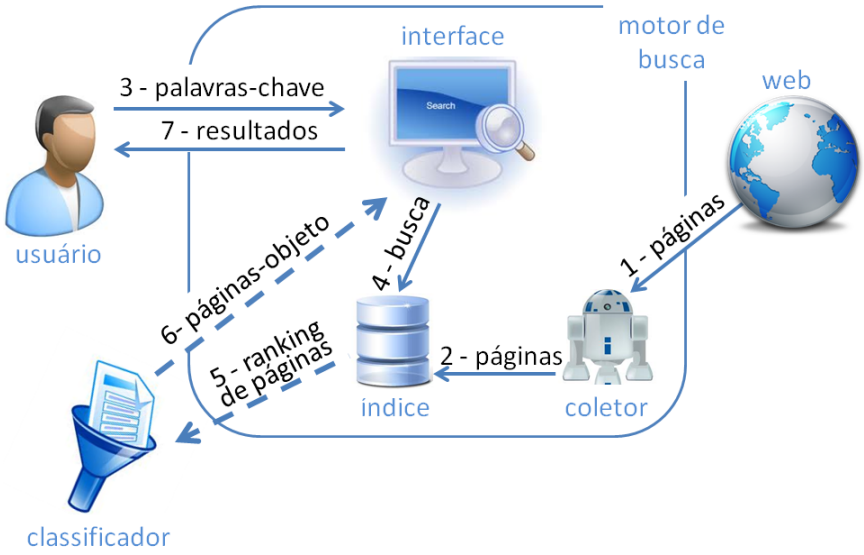

Figura 4.1: Visão geral do OPIS.

avalia a relevância (julga se uma página é ou não uma página-objeto) de um grupo de páginas relacionadas ao domínio de interesse e esse conjunto de páginas analisadas é, então, considerado como base de treinamento. O modelo de classificação resultante é intrinsecamente relacionado ao usuário, uma vez que é este quem determina o conjunto a ser usado no treinamento do classificador. Assim, a corretude da coleção de treinamento e do classificador esta sujeita a correta avaliação do usuário quanto ao que é ou não uma página-objeto, sendo, dessa forma, importante que o usuário tenha conhecimento prévio acerca do domínio a ser treinado.

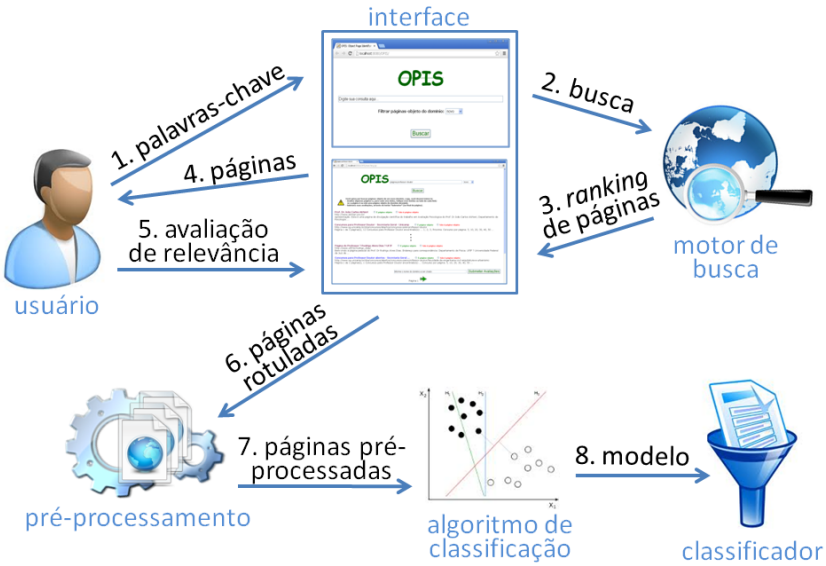

Figura 4.2: Construção do modelo de classificação apoiada por realimentação de relevância.

Na Figura 4.2, o processo de construção do classificador é apresentado. Inicialmente, o usuário deve estabelecer os principais tipos de páginas-objeto que existem no domínio a

ser treinado (para o domínio de professores/pesquisadores, por exemplo, seriam páginas institucionais, pessoais e de currículos, etc. que descrevem um objeto pesquisador). Com base nisso, uma ou mais consultas por palavras-chave devem ser criadas com o objetivo de recuperar essas páginas (como "*professor doutor homepage*", por exemplo) a fim de iniciar o processo de realimentação de relevância. Note que quando o usuário submete uma consulta ela é encaminhada para um motor de busca convencional e as páginas recuperadas são apresentadas ao usuário, para que este possa avaliar a relevância (é ou não uma página-objeto) de um conjunto de páginas. As páginas rotuladas são, então, submetidas a uma etapa de pré-processamento, na qual serão preparadas para servir de entrada para o treinamento do algoritmo de classificação, que resultará no classificador a ser integrado ao motor de busca. Esse processo pode ser repetido com mais consultas, atualizando o conjunto de treinamento e, consequentemente, o classificador, até que o usuário julgue necessário, ou seja, até que o usuário considere ter rotulado uma quantidade suficiente de páginas para exemplificar tanto páginas não objeto quanto os principais tipos de páginas-objeto estabelecidos inicialmente.

4.2 Identificação (Classificação)

Como mencionado na Seção 4.1, a etapa de identificação de páginas-objeto do OPIS é realizada por meio de uma tarefa de classificação, que considera o conteúdo das páginas web. Como a definição de páginas-objeto está diretamente atrelada a um domínio, um modelo de classificação deve ser construído para cada domínio de interesse. Para isso, o OPIS conta com a participação do usuário em um processo de realimentação de relevância, no qual o usuário deve rotular um conjunto de páginas como sendo ou não páginas-objeto, considerando o domínio de interesse, e esse conjunto serve, então, como base para o treinamento do algoritmo de classificação.

Quando o usuário submete uma consulta por palavras-chave com o objetivo de rotular páginas para o treinamento de um novo domínio de páginas-objeto, essa consulta é encaminhada e respondida por um motor de busca convencional, sendo apresentados na interface 10 resultados de busca por página de navegação. Ao rotular alguns resultados de uma página de navegação, o usuário deve submeter sua avaliação antes de trocar de página, atualizando incrementalmente a base de treinamento e o modelo de classificação. A fim de ajudar a popular automaticamente a base de treinamento, a cada submissão de avaliação, o OPIS também considera uma pseudo realimentação de relevantes [1], na qual o classificador, caso já tenha sido treinado com um conjunto de páginas inicial, é aplicado aos resultados não rotulados da presente página de navegação. Dessa forma, para cada resultado de busca não rotulado pelo usuário é obtida uma estimativa das probabilidades deste resultado pertencer às classes positiva e negativa de páginas-objeto. Caso uma página de resultado obtenha, para uma das classes, uma estimativa de probabilidade igual ou superior a 90%, essa página é rotulada com a classe predita e adicionada ao conjunto de treinamento. Essa abordagem assume como correta uma classe predita com um percentual de probabilidade bastante alto. Embora haja a possibilidade de predições incorretas serem adicionadas à base de treinamento, esses casos são considerados exceções, uma vez

[1] O termo "pseudo realimentação de relevantes" foi adotado nesta dissertação para diferenciar essa etapa de população automática do processo de realimentação de relevantes, ilustrado na Figura 4.2, no qual o usuário rotula manualmente as páginas. Porém, faz-se importante mencionar que o mesmo não deve ser confundido com o conceito de "*blind relevance feedback*", apresentado em MANNING; RAGHAVAN; SCHÜTZE (2008)

que, em geral, quando um modelo obtém porcentagens tão altas na classificação de uma instância é porque ele já possui um número representativo de exemplos semelhantes em sua base de treinamento.

Algorithm 1: Treinamento do modelo de classificação

```
   Input: evaluations, domain
   Output: model
 1 begin
 2     model ⟵ getModel(domain);
 3     train_set ⟵ getTrainSet(domain);
 4     for each e in the evaluations do
 5         instance ⟵ generateInstance(e);
 6         if instance.class = null then
 7             if model = null then
 8                 continue;
 9             else
10                 processed_instance ⟵ preprocess(instance);
11                 predicted_class ⟵ classify(model, processed_instance);
12                 probability ⟵ getDistribution(model, processed_instance);
13                 if (probability.objectPage >= 0.9) OR (probability.noObjectPage
                   >= 0.9) then
14                     instance.class ⟵ predicted_class;
15                 end
16             end
17         end
18         train_set.add(instance);
19     end
20     processed_set ⟵ preprocess(train_set);
21     model ⟵ trainClassifier(processed_set, domain);
22     return model;
23 end
```

Essa abordagem é especificada no Algoritmo 1, que representa o processo de treinamento/atualização do modelo de classificação, o qual recebe, como entrada, as avaliações submetidas, referentes aos resultados de uma página de navegação, além do domínio em treinamento. A avaliações são representadas internamente como pares <URL, boolean>, nos quais, se uma página de resultado (representada por uma determinada URL) tiver sido avaliada pelo usuário, esta estará relacionada ao valor `true`, caso tenha sido rotulada como sendo uma página-objeto, ou ao valor `false`, caso contrário. Caso o usuário não tenha avaliado um resultado da atual página de navegação, a URL desse resultado será relacionada ao valor `null`, indicando que essa página é passível de ser rotulada automaticamente. O algoritmo verifica a existência prévia, no banco de dados, de um modelo de classificação relacionado ao domínio de interesse e o obtém, caso exista, por meio do método `getModel` (linha 2). O mesmo ocorre com o conjunto de treinamento por meio do método `getTrainSet` (linha 3). Para cada par de avaliação, gera-se uma instância, representada no formato adotado pelo algoritmo de classificação, contendo o conteúdo textual da página e sua classe. Se a classe for `null`, ou seja, se a referida página não

tiver sido rotulada pelo usuário, inicia-se a tentativa de rotulação automática, na qual, caso tenha sido obtido (caso exista) um modelo prévio, este é utilizado para classificar e gerar a distribuição de probabilidades (linhas 11 e 12, respectivamente) da instância, anteriormente pré-processada (linha 10). Se a instância obtiver uma probabilidade igual ou superior a 90% para uma das classes (é ou não página-objeto), ela é rotulada com a classe predita (linha 14) e, então, adicionada aos conjunto de treinamento (linha 18). Após percorrer todas as avaliações, o conjunto de instâncias de treinamento resultante é pré-processado (linha 20) e usado para treinar o algoritmo de classificação (linha 21), gerando o modelo de classificação, referente à saída do algoritmo.

Durante o pré-processamento, a biblioteca *HTML Parser* (OSWALD et al., 2013) é utilizada para extrair o conteúdo textual (sem *tags* HTML) das páginas web. Considerando que páginas que possuam como idioma dominante o Português podem apresentar termos em Inglês, principalmente em determinados domínios (tais como os relacionados a comércio eletrônico), também é usada a biblioteca *Web Translator Java* (WEB TRANSLATOR JAVA, 2013) para traduzir esses termos para o Português, de modo a casar a ocorrência de uma mesma palavra representada em ambos os idiomas. Após, são removidas *stop words*, considerando a lista fornecida em SNOWBALL (2013a), e o conteúdo textual resultante de cada página é representado no modelo VSM (*bag of words*), usando TF-IDF como forma de ponderação dos termos. Essas atividades são aplicadas com o auxílio do filtro *StringToWordVector*, fornecido pela biblioteca de mineração WEKA (do Inglês, *Waikato Environment for Knowledge Analysis*) (UNIVERSITY OF WAIKATO, 2013), escolhida para auxiliar no pré-processamento e, posteriormente, no processo de classificação. O ambiente WEKA foi considerado por, além de apresentar uma ampla coleção de algoritmos, ser de código aberto e escrito em Java (o que garante portabilidade), assim como as demais bibliotecas utilizadas. O filtro *StringToWordVector* também possibilita a aplicação de *stemming*, porém, alguns testes iniciais mostraram que essa atividade não introduz ganho no processo de classificação, o que pode ser explicado pelo fato de o *stemming* melhorar a revocação, mas prejudicar a precisão (MANNING; RAGHAVAN; SCHÜTZE, 2008) dos resultados. Como a precisão é considerada mais importante para o problema de classificação em questão, optou-se pela não utilização dessa técnica. Faz-se importante notar que essas mesmas atividades de pré-processamento são aplicadas às instâncias (páginas) também no momento da filtragem e não apenas no treinamento do classificador, de modo a manter uma padronização na representação do conteúdo.

Nas subseções seguintes, são apresentados os métodos adotados para escolher um algoritmo de classificação (e sua parametrização) a ser adotado pelo OPIS e um número minimamente razoável de páginas a serem rotuladas pelo usuário, de modo a gerar modelos de classificação e, consequentemente, uma filtragem consistentes.

4.2.1 Escolha e Parametrização do Algoritmo

A fim de escolher um algoritmo de classificação para compor o OPIS, foram comparadas previamente implementações, fornecidas pela biblioteca WEKA, de alguns algoritmos utilizados na classificação de conteúdo de páginas web citados em NAVADIYA; PATEL (2012): `J48` (de árvore de decisão), `IBk` (de kNN, do Inglês *k-Nearest Neighbor*), `NaiveBayes - NB` (probabilístico, baseado no Teorema de Bayes), `LibSVM` (de Máquina de Vetores de Suporte) e `MultilayerPerceptron - MLP` (de Redes Neurais). Faz-se importante observar que a biblioteca WEKA contém apenas um *wrapper* para o LibSVM, sendo necessária a inclusão do JAR (do Inglês, *Java Archive*) referente à implementação desse algoritmo, que pode ser obtido em CHANG; LIN (2013).

Para possibilitar essa comparação, foi coletado um conjunto de páginas relacionadas ao domínio de professores/pesquisadores, escolhido por possuir um número representativo de páginas-objeto disponíveis e por ser de conhecimento prévio. Inicialmente, foi selecionado um conjunto de páginas-*hub*, que se caracterizam por apresentarem uma lista de pesquisadores e *links* para suas páginas-objeto. No domínio considerado, páginas-*hub* são, em geral, páginas de cursos, departamentos ou universidades que apresentam seu corpo docente. Na Tabela 4.1 é apresentada a procedência das páginas-*hub* coletadas, 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*, e as páginas-objeto, indicadas pelos *links* contidos nas páginas-*hub*, foram armazenadas. A API (do Inglês, *Application Programming Interface*) *Google Custom Search* (GCS) (GOOGLE, 2013) foi usada para coletar exemplos negativos (páginas que não são páginas-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 `hub`, `objeto` e `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 dos classificadores, uma vez que suas páginas podem conter diversas semelhanças com as páginas-objeto.

Tabela 4.1: Descrição da procedência das páginas-hub consideradas.

Universidades	**Cursos de Graduação**
Universidade Federal de Santa Maria	Ciência da Computação
Universidade Federal do Rio Grande do Sul	Ciência da Computação
Universidade Federal do Paraná	Ciência da Computação, Matemática e Estatística
Pontifícia Universidade Católica do Rio de Janeiro	Ciência da Computação
Universidade Federal do Rio de Janeiro	Ciência da Computação
Universidade de São Paulo	Ciência da Computação, Matemática e Estatística
Universidade Federal do Estado do Rio de Janeiro	Ciência da Computação
Fundação Universidade Federal do Rio Grande	Ciência da Computação
Universidade Federal de Pelotas	Ciência da Computação
Universidade Federal de Santa Catarina	Ciência da Computação

Todas as páginas da coleção foram pré-processadas de acordo com as atividades mencionadas na Seção 4.2, servindo como base de treinamento e teste para os algoritmos. Para avaliar o desempenho dos algoritmos de classificação, foi utilizada a validação cruzada com *k* (*folds*) igual a 10, que divide a coleção em *k* partes (*folds*) de mesmo tamanho, treina o classificador com *k-1* dessas partes e o testa com a parte restante. Este procedimento é repetido por *k* vezes, cada uma usando um subconjunto de validação diferente, sendo que, ao final, a taxa de acerto é uma média das taxas de acerto obtidas nas *k* iterações realizadas (WITTEN; FRANK; HALL, 2011).

Procurou-se fazer uma variação dos parâmetros utilizados pelos algoritmos, de modo a escolher as configurações que gerassem melhores resultados. Para os algoritmos `J48` e `IBk`, as configurações padrões fornecidas pelo WEKA obtiveram resultados mais satisfatórios. Já para o `NB`, a opção "`-K`" que gera uma estimativa do *kernel* para modelar

atributos numéricos ao invés de usar uma única distribuição normal, foi ativada. A variedade de *kernels* e de seus respectivos parâmetros disponíveis no algoritmo `LibSVM` torna difícil sua configuração. Embora a configuração padrão do WEKA tenha obtido resultados insatisfatórios para a coleção considerada, variá-la indiscriminadamente se mostrou inviável. HSU; CHANG; LIN (2010) recomendam o uso do *kernel* linear em cenários nos quais o número de instâncias é menor que o número de atributos. Esse caso se aplica ao OPIS, uma vez que o número de páginas (instâncias) rotuladas pelo usuário no processo de realimentação de relevância é limitado e o número de atributos considerado na classificação do conteúdo (texto) das páginas é grande. Dessa forma, optou-se pelo uso do *kernel* linear, o qual não necessita de outros parâmetros específicos. Para o algoritmo `MLP`, foi testada a parametrização padrão do WEKA, porém, além de não obter bons resultados, o algoritmo mostrou-se caro computacionalmente ao requerer mais memória do que estava sendo destinado ao WEKA. Embora tenha-se atendido à requisição para que fosse possível obter a avaliação do algoritmo, optou-se por cancelar a exploração de seus parâmetros.

Na Tabela 4.2 são apresentados os resultados obtidos pelos algoritmos, conforme as configurações escolhidas, considerando a média dos valores de cada classe obtidos pelas seguintes métricas (WITTEN; FRANK; HALL, 2011):

- Precisão – é o resultado da divisão do número de instâncias classificadas corretamente em uma classe pelo número total de instâncias classificadas como pertencentes a essa classe;

- Revocação – é o resultado da divisão do número de instâncias classificadas corretamente em uma classe pelo número real de instâncias dessa classe;

- F-Measure – é uma combinação das duas métricas anteriores, sendo calculada por meio da fórmula $\frac{2 \times \text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}}$.

Tabela 4.2: Desempenho dos algoritmos de classificação, considerando validação cruzada de 10 *folds*.

	Precisão	**Revocação**	**F-Measure**
J48	0.889	0.890	0.890
IBk	0.901	0.895	0.895
NB	0.893	0.870	0.878
LibSVM	**0.931**	**0.932**	**0.932**
MLP	0.442	0.472	0.439

Embora, em geral, os algoritmos não tenham apresentado grandes diferenças de desempenho, o MLP obteve resultados consideravelmente inferiores, talvez porque o número de instâncias das classes não é balanceado na coleção considerada (GUO et al., 2008) ou porque não foi encontrada a parametrização adequada. Considerando os resultados obtidos para o domínio abordado nessa coleção, optou-se pelo uso do algoritmo LibSVM (CHANG; LIN, 2011), de máquinas de suporte, que consta na literatura como um método que tem sido aplicado com sucesso em problemas de recuperação de informação, particularmente de classificação de texto (MANNING; RAGHAVAN; SCHÜTZE, 2008).

SVMs são baseadas no princípio de Minimização de Risco Estrutural, onde a ideia é encontrar uma hipótese *h*, definida como a função de decisão com margem máxima entre os vetores de exemplos positivos e negativos (CHOI; YAO, 2005). A complexidade de *h* é baseada na margem em que ela separa os dados e não no número de atributos, o que justifica o bom desempenho obtido por SVMs ao lidar com uma grande quantidade de atributos, comum na classificação de texto (JOACHIMS, 1998).

4.2.2 Rotulação de Páginas

A fim de investigar a influência que o número de páginas rotuladas exerce sobre o desempenho do classificador e selecionar uma quantidade aconselhável para ser considerada no processo de realimentação de relevância, foi realizado um estudo empírico. Atentando para o fato de que, em situações reais, é improvável que um usuário treine um domínio indicando a relevância de uma grande quantidade de páginas, foram avaliadas as quantidades de 5, 10, 15 e 20 páginas rotuladas para cada classe de exemplo: positiva (páginas-objeto) e negativa (páginas não objeto). Para isso, foram criados quatro modelos de classificação, um para cada quantidade de páginas rotuladas, considerando o processo de treinamento apresentado na Seção 4.2.

Os classificadores foram criados considerando a mesma coleção de páginas, relacionada ao domínio de professores/pesquisadores, que foi utilizada para a escolha do algoritmo, apresentada na Subseção 4.2.1. O processo de treinamento por realimentação de relevância, que é desencadeado pelas buscas por palavras-chave, e o posterior uso dos modelos treinados na filtragem de páginas-objeto, foram realizados por meio de um protótipo de interface, desenvolvido na linguagem Java. A biblioteca *Lucene* (APACHE, 2013a), também em Java, foi utilizada como motor de busca convencional na implementação, permitindo que a coleção de páginas fosse indexada e, assim, consultada. Todas as bibliotecas relacionadas ao pré-processamento de páginas e à classificação, mencionadas na Seção 4.2, também foram usadas, contemplando todas as atividades que compõem o OPIS.

Dentre as consultas por palavras-chave submetidas durante o processo de rotulação das páginas, para treinar os modelos de classificação, estavam: "`professor doutor`" "`página professor`" e "`professor lattes`". A partir dos resultados retornados para essas consultas foram rotuladas, como exemplos positivos, páginas institucionais, pessoais ou de currículo que descrevessem um objeto professor/pesquisador e, como exemplos negativos, páginas relacionadas a concursos, corpo docente, notícias, etc. Faz-se importante ressaltar que foi tomado o cuidado de incluir ao menos um exemplo de cada um dos tipos citados em todos os conjuntos de treinamento, mesmo para o formado por 5 páginas de cada classe.

Para avaliar a qualidade da filtragem obtida pelos quatro modelos de classificação treinados, foram submetidas 10 consultas. As consultas foram construídas considerando os cursos e as universidades que fazem parte da coleção, como "`professor equações diferenciais USP`" , por exemplo, e tiveram seus resultados avaliados por meio da métrica precisão em *n* (p@*n*), com *n* de 5, 10, 15 e 20. Essa métrica considera somente os primeiros *n* resultados recuperados e possibilita um indicativo das posições dos resultados relevantes no *ranking*.

Na Tabela 4.3 são apresentados as médias de cada um dos níveis de precisão, considerando as 10 consultas realizadas, para os classificadores treinados com 5, 10, 15 e 20 páginas de exemplo por classe (positiva e negativa, em relação a ser ou não página-objeto). Embora os valores não tenham apresentado muita diferença, pode-se perceber

Tabela 4.3: Médias das precisões para as 10 consultas.

Páginas por Classe	p@5	p@10	p@15	p@20
5	0.44	0.46	0.39	0.36
10	**0.53**	**0.50**	0.41	0.36
15	**0.53**	0.49	**0.44**	**0.38**
20	0.47	0.48	0.41	0.36

que o modelo de classificação treinado com 15 páginas por classe se mostrou mais constante, obtendo os melhores resultados em três níveis de precisão. O fato de o classificador treinado com 20 páginas ter apresentado resultados piores que os de 15 é um pouco curioso, uma vez que, em geral, quanto maior a base de treinamento, mais preciso o classificador. Porém, considerando que todas as quantidades avaliadas são relativamente pequenas, a adição de apenas uma página que apresente algumas características semelhantes a uma classe e pertença à outra já pode ser suficiente para confundir um pouco o modelo. Esse pode ser o caso, uma vez que no domínio de professor/pesquisador algumas páginas de corpo docente apresentam um mini currículo de cada professor, ao invés de mostrar apenas seus principais dados e os *links* para suas páginas-objeto.

A partir dos resultados, passou-se a considerar a quantidade de 15 páginas por classe como aconselhável para o treinamento de outros domínios, embora os resultados mostrem que a quantidade de 5 páginas já contribui positivamente para a filtragem. Faz-se importante observar que não é possível garantir que uma determinada quantidade de páginas de treinamento vai gerar bons resultados sempre, uma vez que, tratando-se de mineração de dados, diferentes domínios podem apresentar comportamentos diferentes, principalmente porque alguns domínios são mais complexos (que, no caso, significa ter uma maior diversidade de tipos de páginas-objeto e não objeto), podendo necessitar de bases de treinamento maiores.

Tabela 4.4: Médias das precisões para as 10 consultas, considerando ou não a etapa de filtragem.

	p@5	p@10	p@15	p@20
Lucene	0.22	0.23	0.20	0.21
OPIS	0.53	0.49	0.44	0.38
% de ganho	**140%**	**110%**	**119%**	**86%**

Na Tabela 4.4 é apresentado um comparativo entre os resultados obtidos pelo motor de busca sem (Lucene) e com (OPIS) a adição da etapa de filtragem (considerando o classificador treinado com 15 exemplos de cada classe). Pode-se observar que o OPIS recupera resultados mais relevantes, melhorando a precisão do motor de busca convencional para buscas-objeto. O ganho percentual do OPIS em relação ao Lucene variou de 86% (p@20) a 140% (p@5), o que significa que o classificador foi capaz de identificar padrões, a partir do conteúdo das páginas, e, assim, possibilitou que fossem descartadas páginas não classificadas como páginas-objeto, permitindo que outras páginas-objeto pudessem melhorar suas posições no *ranking*.

4.3 Busca

Como mencionado na Seção 4.1, a busca no OPIS é possibilitada por meio da integração dos modelos de classificação a um motor de busca convencional, de forma que as páginas recuperadas pelo motor de busca para uma determinada consulta passem por uma etapa adicional de filtragem, onde as páginas são classificadas como sendo ou não páginas-objeto e somente as que se enquadram na primeira classe são apresentadas ao usuário.

Embora o protótipo de interface de treinamento e busca, descrito na Subseção 4.2.2, tenha sido desenvolvido em Java, de modo a agilizar sua implementação, e considerado o Lucene como motor de busca convencional, uma vez que já existia uma coleção a ser usada nos testes (bastando indexá-la e, dessa forma, permitir a busca de seus documentos), para possibilitar a generalização do uso do método, permitindo o treinamento e a posterior busca de qualquer novo domínio de páginas-objeto, essas decisões de implementação foram repensadas. A seguir, na Subseção 4.3.1, é apresentada de forma mais detalhada a interface desenvolvida e sua implementação.

4.3.1 Interface

A nova interface foi desenvolvida para facilitar o treinamento de novos domínios e a busca de páginas-objeto em domínios já treinados. Para isso, a API GCS, sem nenhuma customização, foi utilizada como motor de busca convencional, tanto no processo de treinamento por realimentação de relevância quanto na integração do classificador, de modo a permitir a filtragem dos resultados. Dessa forma, o índice geral do Google passa a ser utilizado durante a etapa de treinamento, tornando desnecessária a coleta prévia de um conjunto de páginas relacionadas ao domínio de interesse, o que generaliza e facilita o processo de adição de novos domínios.

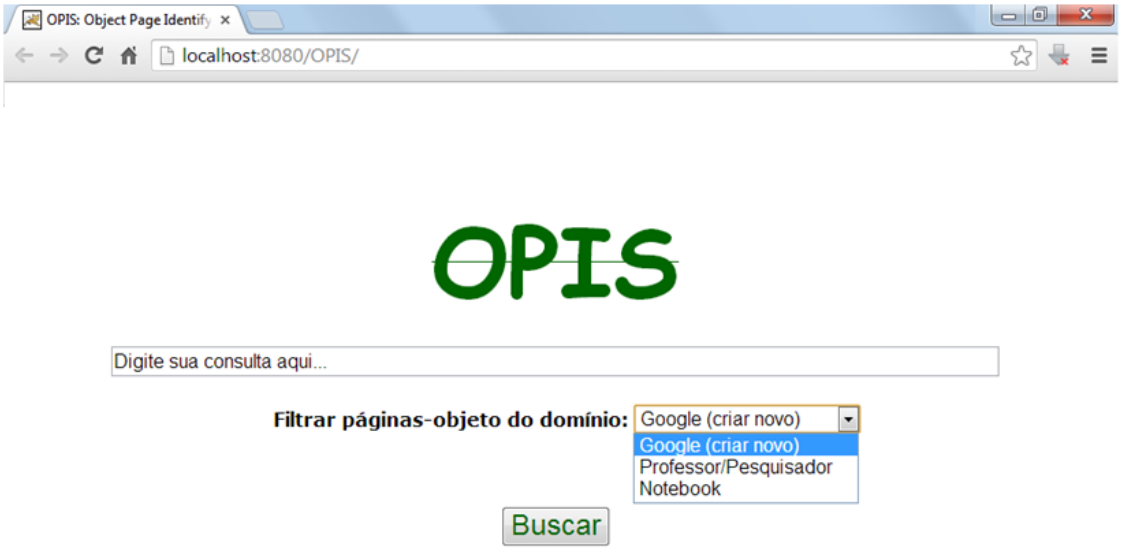

Figura 4.3: Interface do OPIS – tela inicial.

A interface foi desenvolvida na linguagem JSP (do Inglês, *JavaServer Pages*), que é uma linguagem para o desenvolvimento de páginas e aplicações web, baseada na linguagem de programação Java (o que também garante sua portabilidade), que facilita a criação de conteúdos dinâmicos na web (ORACLE, 2013). Para implantar e executar páginas JSP, necessita-se de um servidor compatível com essa tecnologia, tendo sido utilizado o Apache Tomcat (APACHE, 2013b).

Na Figura 4.3 é apresentada a tela inicial da interface do OPIS, na qual pode-se observar a existência não só de uma caixa de texto, para a digitação da consulta, mas também a de uma lista de seleção, por meio da qual o usuário informa se deseja criar/treinar um novo domínio ou buscar páginas-objeto utilizando o modelo de classificação de um domínio que já tenha sido treinado anteriormente.

Quando se trata de um novo domínio, o processo de realimentação de relevância é iniciado. A consulta é encaminhada para ser respondida pelo GCS e, então, os resultados são apresentados ao usuário, de modo que este possa rotular um conjunto de exemplos positivos e negativos de páginas-objeto. As avaliações do usuário são indicadas por meio de caixas de seleção, que indicam se uma página é ou não uma página-objeto e são apresentadas ao lado de cada resultado de busca, como pode ser observado na Figura 4.4. Ao rotular algumas páginas, o usuário pode submeter suas avaliações, informando antes o nome do domínio em treinamento, e, após, navegar pelas demais páginas de resultados (por meio da seta verde, posicionada na parte inferior da interface), rotulando mais páginas e atualizando o modelo de classificação. Novas consultas podem ser submetidas a fim de dar continuidade ao processo de treinamento, até que o usuário considere ter rotulado uma quantidade suficiente de páginas para exemplificar os tipos de páginas-objeto e não objeto que fazem parte do domínio de interesse.

Figura 4.4: Interface do OPIS – tela de treinamento.

Para executar buscas em um domínio já treinado, o usuário pode partir tanto da tela inicial quanto da de treinamento, devendo apenas selecionar o domínio desejado na lista de seleção, de forma a habilitar a etapa de filtragem. Assim, os resultados da consulta, recuperados pelo GCS, serão filtrados de acordo com o modelo de classificação selecionado, permitindo que somente as páginas classificadas como páginas-objeto (para o domínio escolhido) sejam apresentadas na interface.

4.4 Considerações Finais

Neste capítulo foi apresentado o método OPIS, que visa a identificação e a busca de páginas-objeto, de modo a tornar os resultados de buscas-objeto mais precisos e adequados às necessidades de informação dos usuários. O método proposto adota técnicas de realimentação de relevância, pré-processamento de texto e de aprendizagem de máquina para construir, com base no conteúdo das páginas web, modelos de classificação específicos a domínios, que são utilizados para determinar se uma página é ou não uma página-objeto para um determinado domínio de interesse. Esses classificadores são, então, integrados a um motor de busca convencional, adicionando uma etapa de filtragem ao processo de busca, de modo que apenas páginas classificadas como páginas-objeto sejam apresentadas como resposta a uma busca-objeto do usuário.

Na Tabela 4.5 é apresentado um resumo comparativo entre os principais trabalhos relacionados, dentre os apresentados no Capítulo 3, e o OPIS.

Tabela 4.5: Comparativo entre o OPIS e os principais trabalhos relacionados.

	Objetivo	**Abordagem**	**Usuário**
BLANCO et al. (2008)	coletar páginas que representem instâncias de entidades	coleta focada, considerando a similaridade das estruturas das páginas (propriedades de apresentação e disposição dos *links*)	comum, indica apenas um conjunto de páginas sementes
ASSIS et al. (2009)	coletar páginas de tópicos que podem ser representados por aspectos de gênero e conteúdo	coleta focada, fazendo uma combinação das similaridades de cosseno de ambos os aspectos	especialista, informa um conjunto de páginas sementes e um conjunto de termos para cada um dos aspectos
PHAM et al. (2010)	prover a construção de motores de busca verticais para permitir a busca-objeto	aprendizagem de uma função de *ranking* para cada domínio	especialista, rotula páginas de treinamento e seleciona características
OPIS (*Object Page Identifying and Searching*)	prover a filtragem de páginas-objeto durante o processo de busca	construção de um modelo de classificação para cada domínio	comum, rotula páginas considerando apenas um aspecto

Pode-se observar que, enquanto o trabalho de BLANCO et al. (2008) se concentra na coleta de páginas de entidades com estruturas similares, considerando propriedades de apresentação e disposição de *links*, o OPIS utiliza classificação baseada em conteúdo de páginas web para identificar e filtrar páginas-objeto durante o processo de busca, sem considerar as estruturas das páginas.

Já em relação à proposta de ASSIS et al. (2009), o OPIS não considera o conteúdo e o gênero das páginas de forma separada durante o treinamento de um modelo e a posterior classificação de páginas-objeto. Isso reduz o nível de especialidade do usuário, uma vez que este não precisa discernir entre esses dois aspectos e nem selecionar termos manualmente para caracterizá-los.

Assim como no OPIS, o trabalho de PHAM et al. (2010) utiliza o conteúdo das páginas para aprender um novo domínio e possibilitar a busca-objeto nesse domínio. Porém, o OPIS se concentra na criação de modelos para a classificação funcional (página-objeto ou página não objeto) de páginas e não na criação de funções de *ranking*. Além disso, no OPIS o usuário que guia o processo de treinamento de um novo domínio não precisa ser tão especialista, uma vez que ele só rotula exemplos positivos e negativos de páginas-objeto, sem precisar selecionar características apropriadas para compor a função de *ranking*.

5 AVALIAÇÃO EXPERIMENTAL

Nesta seção são apresentados os experimentos que foram realizados com o intuito de avaliar a influência do OPIS na qualidade dos resultados de buscas-objeto. Esses experimentos são agrupados em dois momentos, de acordo com o *baseline* considerado: inicialmente, foi realizada uma experimentação comparando os resultados do OPIS aos do Google e, após, experimentos mais exaustivos foram desenvolvidos, utilizando como *baseline* o *Object Search Engine* (OSE), proposto por PHAM et al. (2010) e apresentado previamente como trabalho relacionado (Seção 3.3). A mudança de *baseline* ocorreu com o objetivo de tornar a comparação mais justa, uma vez que o OSE (o qual não se conseguiu utilizar no primeiro momento) também visa a busca-objeto, ao contrário do Google, que é de caráter genérico.

Faz-se importante salientar que os sistemas de busca adotados como *baselines* fazem uso de coleções de páginas distintas. Pode-se dizer que o Google possui em seu índice praticamente toda a Web, enquanto o OSE, assim como o Lucene, pressupõe a existência de uma coleção obtida previamente. Como o índice do Google não é de domínio público e a base utilizada pelo OSE não pode ser indexada pela API do Google, uma vez que a quantidade de documentos passível de ser indexada gratuitamente não é suficiente, foi adotada uma coleção distinta de páginas em cada momento da avaliação experimental, não sendo, dessa forma, os resultados de ambos os *baselines* comparáveis entre si.

Nas próximas seções são detalhados os experimentos realizados (incluindo os casos de estudo e a metodologia utilizada, além dos resultados obtidos), considerando essa divisão de acordo com os *baselines*.

5.1 OPIS e Google

Nesta seção é detalhada a primeira fase experimental realizada, a qual considera como *baseline* o Google, por meio do uso da biblioteca *Google Custom Search* (GCS), sem nenhuma customização. Esses experimentos foram realizados com o intuito de verificar a aplicabilidade do OPIS e, dessa forma, permitir uma avaliação da influência que ele exerce sobre os resultados de buscas-objeto, em relação a um motor de busca convencional.

A seguir, nas Subseções 5.1.1 e 5.1.2, são descritos, respectivamente, o caso de estudo e a metodologia adotada nesta experimentação. Os resultados obtidos, juntamente com suas discussões, são apresentados na Subseção 5.1.3.

5.1.1 Caso de Estudo

Para possibilitar a experimentação do OPIS, inicialmente, faz-se necessário definir um caso de estudo, o qual consiste na escolha e no treinamento de um domínio, a ser

explorado pelas buscas-objeto. Para esse caso de estudo, optou-se pelo domínio de professor/pesquisador, que foi treinado por meio do processo de realimentação de relevância, através do qual são submetidas consultas por palavras-chave e alguns dos resultados recuperados por essas consultas são rotulados como sendo ou não páginas-objeto do domínio desejado, seguindo o procedimento já explicado na Seção 3.

A fim de treinar o classificador para o domínio de professor/pesquisador, foram utilizadas as consultas "`página professor doutor`", com o objetivo de encontrar páginas institucionais e pessoais que descrevam professores/pesquisadores, e "`professor doutor currículo`", visando a recuperação de páginas que apresentem os currículos de tais profissionais. Navegando entre os resultados dessas consultas, foram rotuladas 15 páginas-objeto, relacionadas aos tipos de páginas-objeto desejados pelas consultas, e 15 páginas não objeto, como páginas relacionadas a notícias, corpos docente, concursos, disciplinas e laboratórios, que apenas mencionam ou listam objetos professores, sem descrevê-los individualmente.

O modelo de classificação resultante foi, então, considerado pelo OPIS na filtragem (identificação) de páginas-objeto do domínio de interesse deste caso de estudo (professor/pesquisador), possibilitando a realização dos experimentos apresentados, a seguir, nesta seção.

5.1.2 Metodologia

Considerando a interface do OPIS, por meio da qual pode-se habilitar a filtragem de páginas-objeto, e o modelo de classificação, treinado anteriormente para o domínio de professor/pesquisador, foi realizado um estudo acerca da viabilidade do método como um todo. Esse estudo consistiu na submissão de buscas-objeto do domínio de interesse e na avaliação da relevância dos resultados recuperados para essas consultas, tendo contado com a participação de 10 usuários. Cada usuário criou e submeteu cinco consultas por páginas-objeto, tendo também avaliado os resultados recuperados por essas consultas como sendo ou não páginas-objeto do domínio desejado.

As consultas dos usuários foram realizadas no Google (motor de busca convencional) e no Google+OPIS (com a adição do processo de filtragem). A fim de tornar o processo menos exaustivo, os usuários foram divididos em dois grupos, permitindo que cada grupo realizasse suas consultas e avaliasse os resultados em apenas um dos métodos. Após, essas consultas foram reproduzidas e avaliadas no método oposto, sem que fosse necessária uma nova participação dos usuários. Para evitar o uso de diferentes critérios de relevância durante a reprodução das consultas, os usuários detalharam o objetivo e os tipos de resultados que consideraram relevantes em cada uma de suas consultas.

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) foi novamente considerada com n de 5, 10, 15, e 20.

5.1.3 Resultados

Os resultados obtidos a partir da execução e da avaliação das consultas, conforme a metodologia apresentada na Subseção 5.1.2, para o caso de estudo do domínio de professor/pesquisador, descrito na Subseção 5.1.1, são apresentados na Tabela 5.1. Essa tabela contém as médias das precisões em n (p@n), com n de 5, 10, 15 e 20, para as cinco consultas de cada usuário, obtidas por meio do Google com (OPIS) e sem a adição da etapa

de filtragem.

Tabela 5.1: Resultados para as consultas de cada usuário.

	Média das 5 consultas de cada usuário para Google | Google+OPIS			
Usuário	**p@5**	**p@10**	**p@15**	**p@20**
1	0.080 | **0.240**	0.120 | **0.260**	0.133 | **0.213**	0.150 | **0.180**
2	0.040 | **0.120**	0.080 | **0.160**	0.133 | **0.173**	0.120 | **0.190**
3	0.000 | **0.120**	0.060 | **0.100**	0.067 | **0.145**	0.060 | **0.150**
4	0.120 | **0.160**	0.120 | **0.140**	0.133 | **0.147**	0.110 | **0.130**
5	0.080 | **0.200**	0.160 | **0.180**	0.133 | **0.148**	0.120 | **0.130**
6	0.040 | **0.160**	0.120 | **0.260**	0.160 | **0.293**	0.160 | **0.280**
7	0.000 | **0.120**	0.020 | **0.100**	0.027 | **0.096**	0.070 | **0.110**
8	0.080 | 0.080	0.100 | 0.100	0.119 | **0.120**	0.130 | 0.130
9	0.080 | **0.120**	0.080 | **0.120**	0.054 | **0.106**	0.070 | **0.120**
10	0.080 | **0.160**	0.060 | **0.220**	0.107 | **0.174**	0.120 | **0.190**

Os resultados mostram que o uso do OPIS permite a recuperação de um número maior de páginas relevantes nos níveis de corte (n) considerados, melhorando as precisões do motor de busca convencional para buscas-objeto. Os ganhos variam de zero (linha 8) a 300 (linha 6) por cento, observando p@5 e desconsiderando os casos em que o Google apresenta valores de precisão igual a zero (linhas 3 e 7), por exemplo.

Pode-se deduzir que o classificador foi capaz de identificar padrões, a partir do conteúdo das páginas, durante o treinamento e, assim, aprendeu a identificar páginas-objeto para o domínio testado, permitindo que somente páginas classificadas como páginas-objeto fossem retornadas pelas consultas submetidas ao OPIS. Considerando o exemplo introdutório de busca-objeto "`professor UFRGS`", apresentado no Capítulo 1, páginas identificadas como sendo páginas-objeto passam a substituir páginas irrelevantes, como páginas relacionadas a concursos de professores, no *ranking*, aumentando a chance das páginas retornadas atenderem as necessidades de informação do usuário (serem consideradas relevantes).

Tabela 5.2: Média das precisões para todas (50) as consultas dos usuários.

	p@5	**p@10**	**p@15**	**p@20**
Google	0.060	0.092	0.107	0.111
Google+OPIS	0.148	0.164	0.162	0.161
% de ganho	**147%**	**78%**	**52%**	**45%**

Na Tabela 5.2 são apresentadas as médias das precisões de todas as consultas para o Google e o Google+OPIS. Nota-se que os ganhos de precisão fornecidos pelo OPIS variam de 45%, considerando as 20 primeiras páginas retornadas, a 147%, considerando as 5 primeiras páginas. Esses ganhos decorrem, principalmente, do fato de muitas páginas irrelevantes (que não são páginas-objeto) retornadas pelo Google para buscas-objeto serem descartadas pelo modelo de classificação do OPIS, tornando os resultados mais precisos.

Analisando as páginas retornadas por cada uma das 50 consultas em ambos os métodos, notou-se que o número de páginas relevantes recuperadas pelo OPIS foi menor em apenas 6 consultas. Essas consultas eram bastante restritivas, tendo, possivelmente, um número reduzido de páginas relevantes na coleção (índice do Google). Dentre essas

consultas, uma buscava por pesquisadores da área de informática teórica que atuassem na UFRGS (0 vs. 0.05 de p@20) e outra visava encontrar pesquisadores que fizessem parte do Intituto Tecgraf da PUC-Rio (0.20 vs. 0.25 de p@20). Nesses casos, algumas páginas relevantes recuperadas pelo Google foram incorretamente classificadas como páginas não objeto, não sendo, dessa forma, retornadas pelo OPIS. Porém, para as demais buscas-objeto, o OPIS apresentou resultados equivalentes ou superiores aos do Google.

Com base no Teste-T (GOSSET, 1908), é possível afirmar que o uso do OPIS melhora significativamente (valor-p do Teste-T < 0.01) os valores de precisão para buscas-objeto em todos os níveis de corte considerados, quando comparado ao Google, para o caso de estudo utilizado. Além disso, por meio da distribuição dos resultados relevantes no *ranking*, indicada pelos níveis de precisão apresentados na Tabela 5.1, pode-se notar que o OPIS permite uma maior concentração de resultados relevantes no início do *ranking*, uma vez que os percentuais de ganho mostram-se inversamente proporcionais aos tamanhos dos níveis de corte.

5.2 OPIS e OSE

Nesta seção é detalhada a segunda fase experimental realizada, a qual considera como *baseline* o *Object Search Engine* (OSE), proposto por PHAM et al. (2010) e apresentado previamente, na Seção 3.3, como trabalho relacionado. Por ser um trabalho de busca-objeto, o OSE propicia uma comparação mais justa e direta, o que motivou seu uso em uma avaliação mais exaustiva. A implementação do OSE e uma coleção de 50 mil páginas (relacionadas aos domínios de professor/pesquisador, *notebook* e câmera digital) foram disponibilizadas por PHAM et al. (2010) e, dessa forma, utilizadas nos experimentos.

A seguir, nas Subseções 5.2.1 e 5.2.2, são descritos, respectivamente, os casos de estudo e a metodologia adotada nesta experimentação. Os resultados obtidos, juntamente com suas discussões, são apresentados na Subseção 5.2.3.

5.2.1 Casos de Estudo

A fim de permitir uma avaliação mais extensa e, dessa forma, mais confiável, foram desenvolvidos três casos de estudo, abrangendo o domínio de professor/pesquisador, já explorado nos experimentos anteriores, além dos domínios de *notebook* e câmera digital. Assim como explicado na Seção 5.1, foi realizado o treinamento desses domínios, sendo seus modelos de classificação gerados de acordo com o processo de realimentação de relevância descrito na Seção 3.

Como o OSE utiliza uma coleção de páginas em Inglês, o treinamento dos modelos de classificação também deve considerar essa linguagem, tanto para a criação das consultas por palavras-chave quanto para a lista de *stop words* a ser usada. Assim, para essa fase experimental, foi, novamente, utilizada uma lista fornecida pelo site Snowball (SNOWBALL, 2013b), mas, dessa vez, com *stop words* em Inglês.

Para o domínio de professor/pesquisador, foram utilizadas as consultas "`professor doctor homepage`" e "`professor doctor curriculum`", tendo sido rotuladas 15 páginas-objeto, incluindo páginas institucionais, pessoais e de currículos, que descrevem professores/pesquisadores; e 15 páginas não objeto, como páginas relacionadas a notícias, corpos docente, concursos, disciplinas e laboratórios, que apenas mencionam ou listam tais profissionais, sem descrevê-los individualmente.

No domínio de *notebook*, foram utilizadas as consultas "`notebook model description`" e "`notebook store product`" para rotular 15 páginas-objeto, relaci-

onadas a páginas de fabricantes e lojas virtuais, que descrevem *notebooks*; e 15 páginas não objeto, incluindo páginas de notícias, análise e listagem desses produtos, que não descrevem *notebooks* individualmente.

Finalmente, para o domínio de câmeras digitais, foram utilizadas as consultas "`digital camera description`" e "`digital camera product`" para rotular 15 páginas-objeto, incluindo páginas de fabricantes e lojas virtuais, que descrevem câmeras digitais; e 15 páginas não objeto, relacionadas a páginas de notícias, análise comparativa e listagem de câmeras digitais, as quais não apresentam esses produtos de forma individual.

Os modelos de classificação treinados, a partir das páginas rotuladas, para esses domínios compõem os casos de estudo utilizados na presente avaliação experimental, que considera os resultados obtidos pelo OPIS e pelo sistema de busca-objeto OSE, adotado como *baseline*.

5.2.2 Metodologia

Para avaliar a influência que o OPIS exerce nos resultados de busca-objeto, dessa vez comparando a outro sistema de busca-objeto (OSE), foram utilizados os casos de estudo apresentados na Seção 5.2.1, que abrangem os domínios de pesquisador, *notebook* e câmera digital.

Os experimentos consistiram na submissão de buscas-objeto e na avaliação da relevância dos resultados recuperados por essas consultas. Para isso, foram criadas, submetidas e avaliadas 51 buscas-objeto para cada um dos domínios de caso de estudo. A fim de proporcionar uma outra perspectiva de análise, as consultas foram distribuídas de acordo com a categorização proposta por POUND; MIKA; ZARAGOZA (2010) para a tarefa de recuperação de objetos na web semântica. Embora esses autores tenham estabelecido cinco categorias de consultas, como o OPIS não se insere no contexto da web semântica, nem todas as categorias propostas puderam ser adaptadas para o conceito de páginas-objeto adotado pelo OPIS, tendo sido utilizadas as seguintes:

- `Consulta de entidade` – possui a intenção de encontrar uma entidade específica, como, por exemplo, as consultas "`professor banco de dados UFRGS`", "`notebook hp pavilion dv5`", "`câmera cybershot h100`". Nesse caso, páginas que descrevem entidades (objetos) que atendem às restrições de desambiguação estabelecidas na consulta podem ser consideradas relevantes;

- `Consulta de tipo` – visa a recuperação de entidades (objetos) de um tipo ou classe particular, como, por exemplo, as consultas "`professor adjunto UFRGS`", "`notebook hp`", "`câmera sony`". Nesse caso, páginas que descrevem entidades que são instâncias do tipo desejado podem ser consideradas relevantes;

- `Consulta de atributo` – possui a intenção de encontrar valores de um atributo específico de uma entidade ou tipo, como, por exemplo, as consultas "`departamento professor banco de dados UFRGS`", "`modelo notebook hp 320gb`", "`zoom câmera sony cybershot`". Nesse caso, páginas que descrevem entidades(objetos) que atendem às restrições estabelecidas e apresentam um valor para o atributo desejado podem ser consideradas relevantes;

Como foram consideradas três categorias, atribuiu-se 17 buscas-objeto para cada categoria, considerando cada estudo de caso (domínio) individualmente, o que totaliza nas

51 consultas mencionadas anteriormente. Essas consultas foram executadas e avaliadas no OPIS e no OSE (*baseline*), proposto por PHAM et al. (2010), que forneceu sua implementação e uma coleção de 50 mil páginas (relacionadas aos domínios de pesquisador, *notebook* e câmera digital), viabilizando a comparação.

A fim de permitir que ambos os métodos usem a mesma coleção nos experimentos, o OPIS foi adaptado para utilizar o Lucene (APACHE, 2013a) como motor de busca convencional, uma vez que a API do Google, utilizada pelo OPIS, possui um índice geral próprio e não permite a indexação gratuita de uma coleção do porte da fornecida por PHAM et al. (2010). Porém, faz-se importante observar que essa adaptação foi realizada apenas para a execução dos experimentos, sem substituir a constituição e a proposta original do OPIS.

No contexto de recuperação com *ranking*, como é o caso dos motores de busca, utiliza-se, naturalmente, um conjunto dos *k* primeiros documentos recuperados para terem suas relevâncias avaliadas, devendo ser consideradas também as posições ocupadas por esses documentos no *ranking* (MANNING; RAGHAVAN; SCHÜTZE, 2008). Nos experimentos apresentados na Seção 5.1, um indicativo das posições dos documentos no *ranking* podia ser observado por meio das variações dos valores de precisão nos níveis de corte utilizados, embora a métrica de precisão em *n* não considerasse, em seu cálculo, diretamente as posições dos resultados relevantes. Na presente fase experimental, como não foram utilizados usuários na avaliação das consultas, optou-se por considerar um conjunto maior de resultados na avaliação, tendo sido utilizadas as primeiras 100 páginas recuperadas para cada consulta submetida.

Por se tornar mais difícil a análise da posição dos resultados ao se considerar um conjunto de 100 páginas com a métrica de precisão em *n* (*p@n*), optou-se pela adoção da métrica de precisão média (do Inglês, *Average Precision* – AvP), que corresponde à média das precisões obtidas após cada documento relevante ser recuperado (MANNING; RAGHAVAN; SCHÜTZE, 2008). Dessa forma, considerando r a posição de cada relevante recuperado no *ranking*, $p@r$ a precisão dos r primeiros documentos recuperados e R o total de relevantes recuperados, a precisão média é obtida por meio da fórmula:

$$AvP = \frac{\sum_r p@r}{R} \tag{5.1}$$

Como, no âmbito da busca na web, não é possível obter o total de relevantes, considerou-se R como a união dos totais de relevantes obtidos, dentre as 100 primeiras páginas avaliadas, por cada método de busca (OPIS e OSE). Além da AvP, a quantidade de resultados relevantes recuperados por cada consulta e a métrica *Mean Average Precision* (MAP), apresentada na Fórmula 5.2, também foram adotadas.

$$MAP = \frac{\sum_q AvP(q)}{Q} \tag{5.2}$$

Considerando q como sendo cada consulta de um conjunto de Q consultas, pode-se notar que a MAP consiste na média dos valores de AvP obtidos por um conjunto de consultas (MANNING; RAGHAVAN; SCHÜTZE, 2008).

5.2.3 Resultados

Nesta Subseção são apresentados os resultados experimentais obtidos a partir da submissão e da avaliação de buscas-objeto, considerando os casos de estudo (relacionados aos domínios de pesquisador, *notebook* e câmera digital) apresentados na Subseção 5.2.1 e a

metodologia descrita na Subseção 5.2.2. A fim de facilitar a análise dos resultados, permitindo uma visualização mais direta das diferenças em relação à perspectiva das categorias de consulta, não utilizada nos experimentos da Seção 5.1, optou-se pela apresentação gráfica dos resultados.

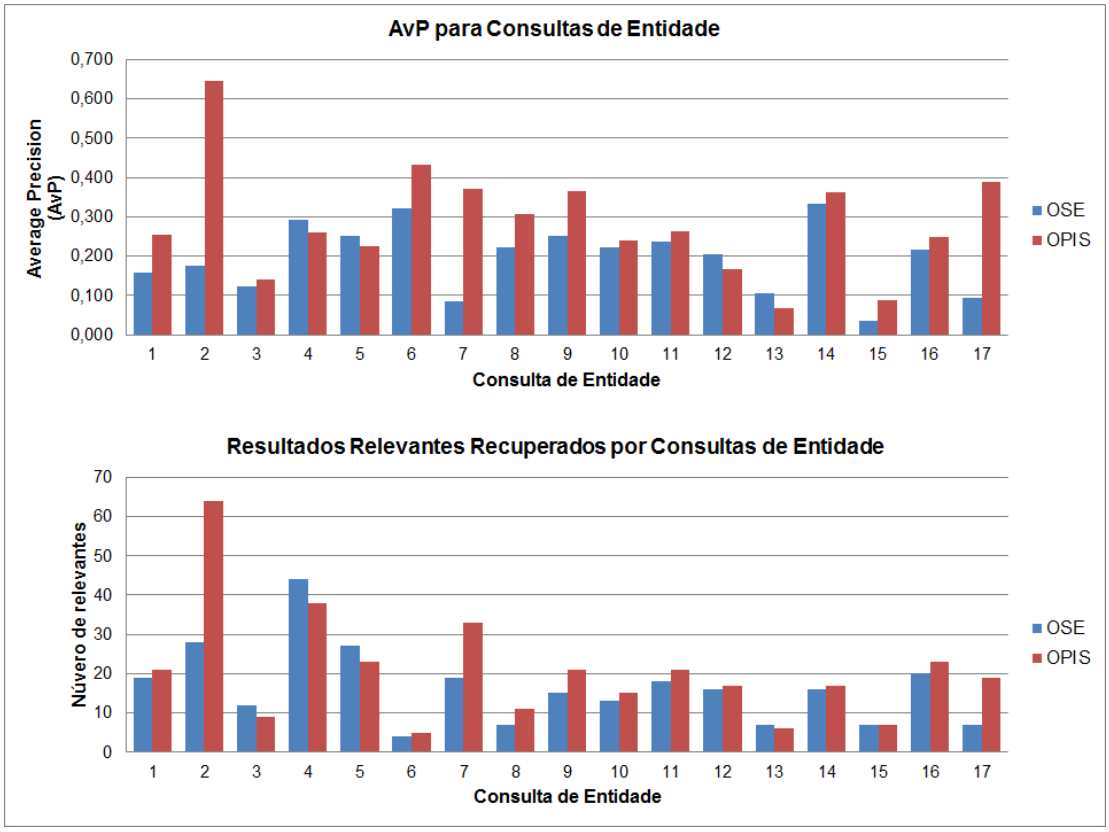

Figura 5.1: Resultados das consultas de entidade para o domínio de pesquisador.

Nas Figuras 5.1, 5.2 e 5.3 são mostrados os resultados de precisão média (AvP) e o número de páginas relevantes recuperadas, considerando o domínio de pesquisador, para cada uma das consultas realizadas nas categorias de entidade, tipo e atributo, respectivamente. Pode-se observar que, para o caso de estudo do domínio de pesquisador, o OPIS superou, quanto à AvP, o OSE em 13 consultas para as categorias de entidade e tipo e em 12 consultas para a categoria de atributo. Considerando as MAPs para as 17 consultas de cada categoria, o OPIS obteve ganhos de 45% (0.284 vs. 0.196), 34% (0.330 vs. 0.246) e 37% (0.303 vs. 0.221) para as consultas de entidade, tipo e atributo, respectivamente. Nesse caso de estudo, pode-se perceber que, em média, os valores de AvP foram maiores para as consultas de tipo, seguidas pelas de atributo e de entidade. Essa mesma ordem pode ser observada para o número de relevantes recuperados, cujos ganhos, em relação ao OSE, obtidos a partir das médias das consultas de cada categoria, foram de 26% (20 vs. 16), 17% (45 vs. 38) e 11% (33 vs. 30) para as consultas de entidade, tipo e atributo, respectivamente. Nota-se que, embora o ganho médio tenha sido maior para a categoria de entidade, essa categoria apresentou as menores quantidades médias de relevantes recuperados, tanto para o OPIS quanto para o OSE. Uma possível explicação para esse fato é a de que, em geral, consultas de entidade são mais restritivas que as de tipo, tendo, assim, um número menor de páginas relevantes associadas. Já para consultas de atributo, as páginas relevantes, além de atenderem aos valores de atributos especificados, precisam

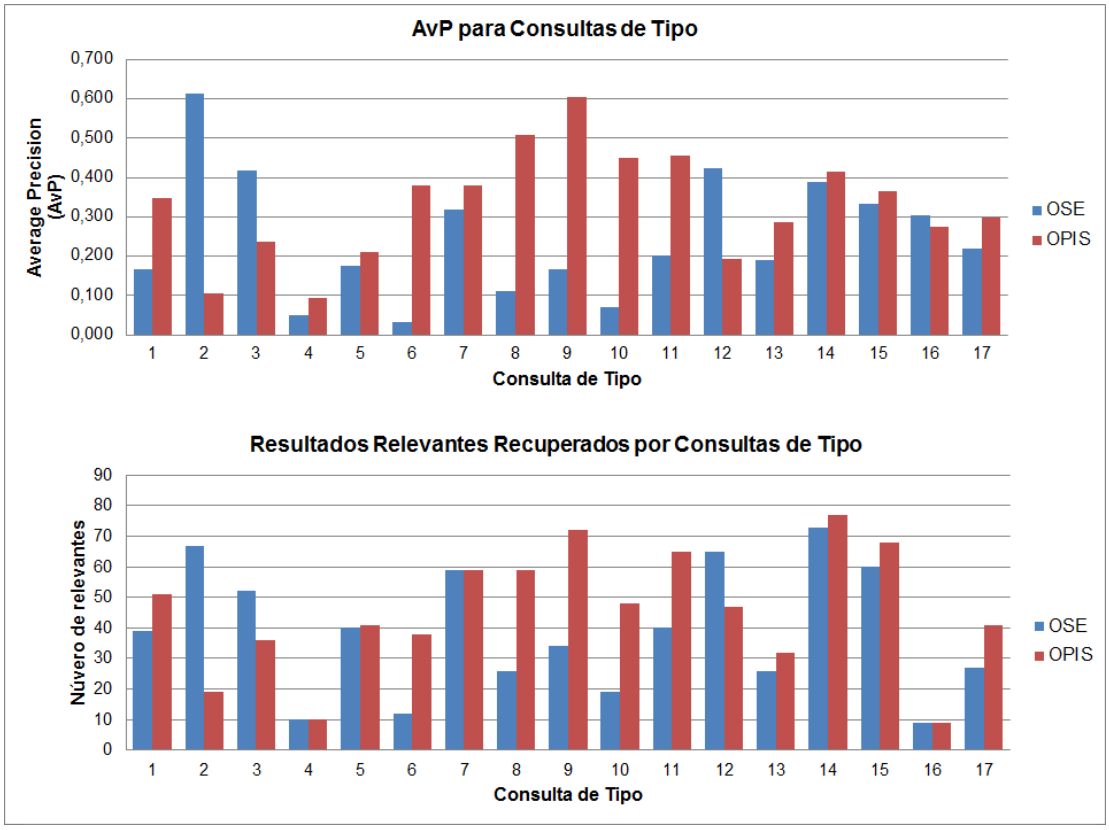

Figura 5.2: Resultados das consultas de tipo para o domínio de pesquisador.

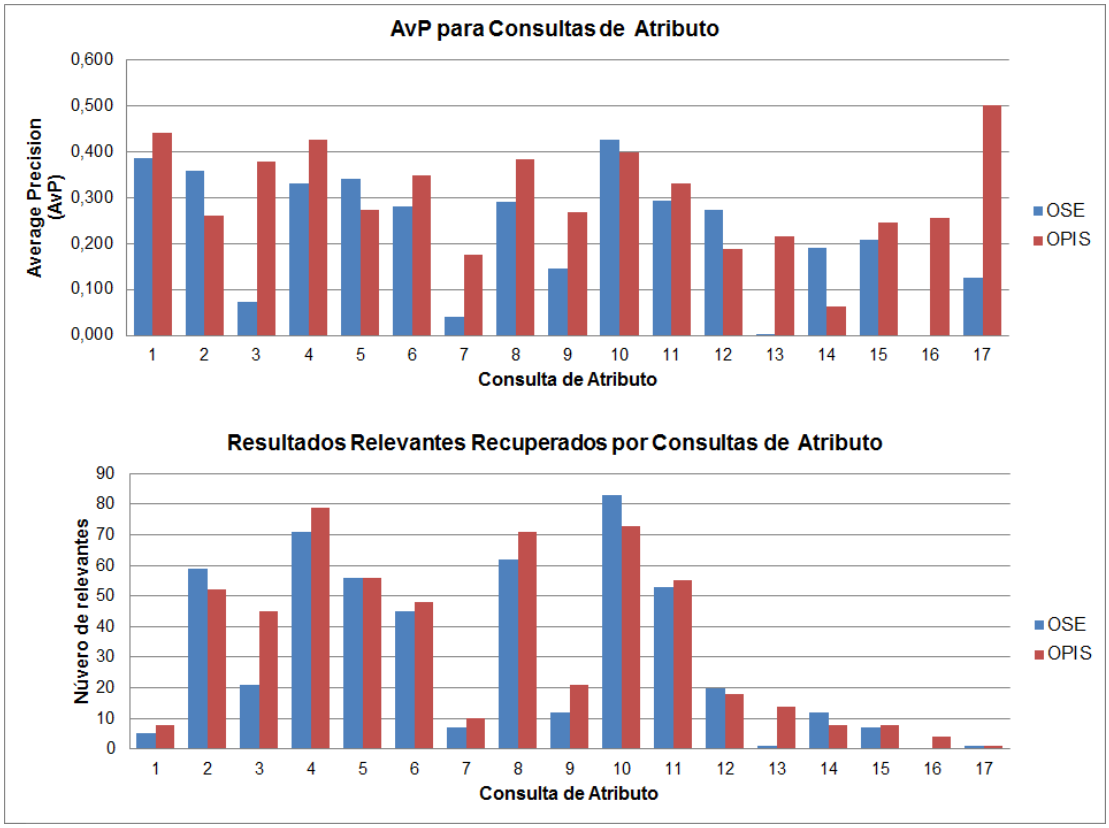

Figura 5.3: Resultados das consultas de atributo para o domínio de pesquisador.

possuir um valor não definido para um atributo de desejo, restringindo mais ainda a noção de relevância.

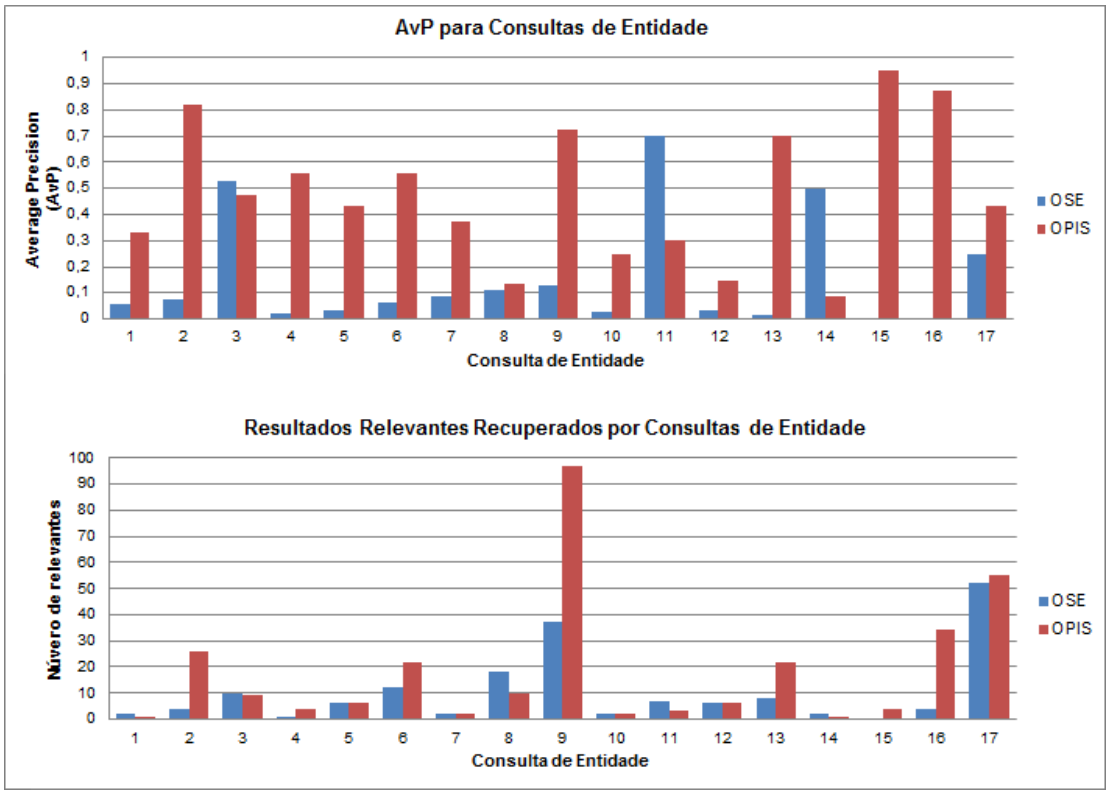

Figura 5.4: Resultados das consultas de entidade para o domínio de *notebook*.

Nas Figuras 5.4, 5.5 e 5.6 são apresentados os gráficos para o estudo de caso do domínio de *notebook*. Pode-se observar que, para a categoria de entidade, o OPIS obteve resultados de AvP maiores ou equivalentes aos do OSE na maioria das consultas, perdendo apenas em três casos. Já nas consultas de tipo, o OPIS foi superado pelo *baseline* em 12 das 17 consultas, o que não ocorreu na categoria de atributo, na qual o OPIS voltou a mostrar-se melhor, tendo superado o OSE em 10 consultas e se igualado em outras três. Uma possível explicação para o fato de o OPIS ter sido superado na maioria das consultas de tipo pode estar relacionada à existência de seções de recomendação em páginas de comércio virtual. Para a consulta de tipo "`notebook sony 750 gb`", por exemplo, uma página relacionada a um *notebook* da marca Sony com HD de 320 gb e que apresente, na seção de recomendação, um *notebook* Sony com HD de 750 gb, poderá ser recuperada, uma vez que o OPIS realiza consultas por palavras-chave e os termos da consulta estão presentes nessa página. Uma vez recuperada, essa página dificilmente será descartada na etapa de filtragem, já que, mesmo não atendendo às necessidades do usuário, ela apresenta as características de uma página-objeto. Ao fazer parte do *ranking*, essa página possibilita que outras páginas, realmente relevantes para a consulta, tenham seus posicionamentos prejudicados, reduzindo o valor da AvP. Esse problema não ocorre com o *baseline*, uma vez que ele trabalha com consultas estruturadas, restringindo mais o posicionamento e a apresentação dos valores dos atributos nas páginas. Considerando as MAPs obtidas por categoria, o OPIS apresentou ganhos de 209% (0.479 vs. 0.155) e 84% (0.431 vs. 0.235) nas consultas de entidade e atributo, respectivamente, tendo perdido em 29% (0.340 vs. 0.482) nas consultas de tipo. Os aumentos relativos à média de relevantes recuperados

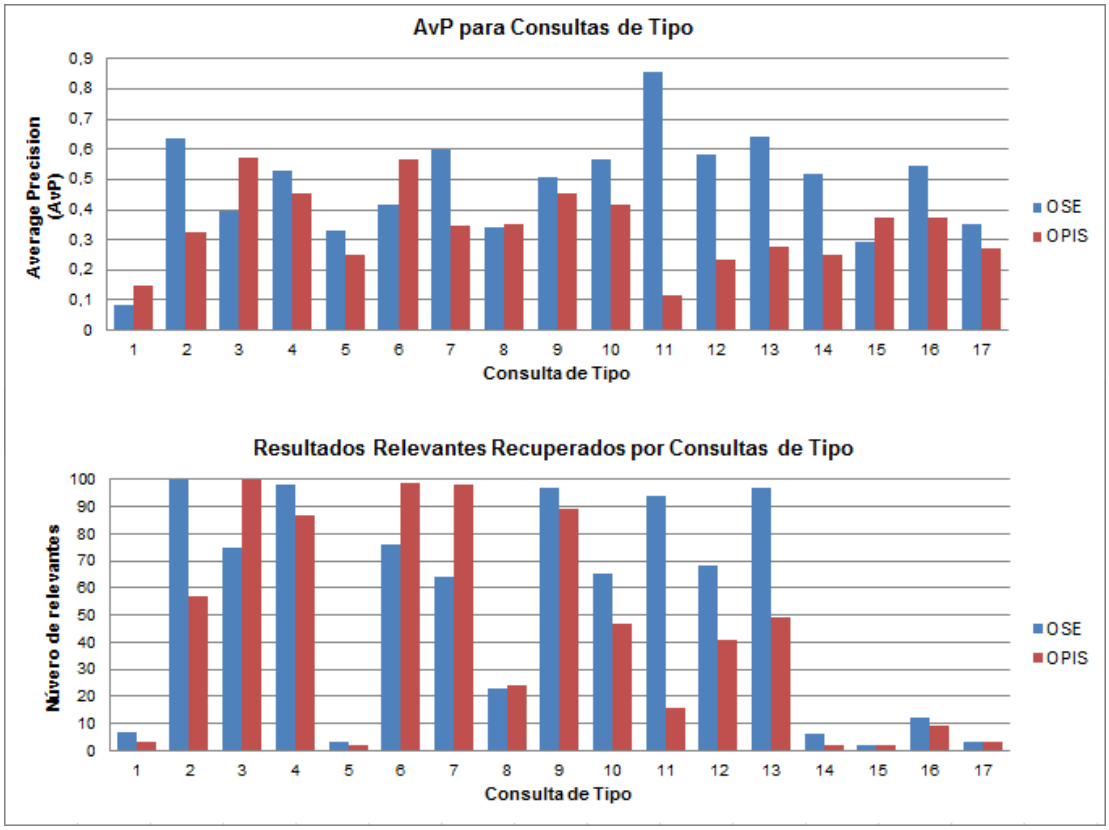

Figura 5.5: Resultados das consultas de tipo para o domínio de *notebook*.

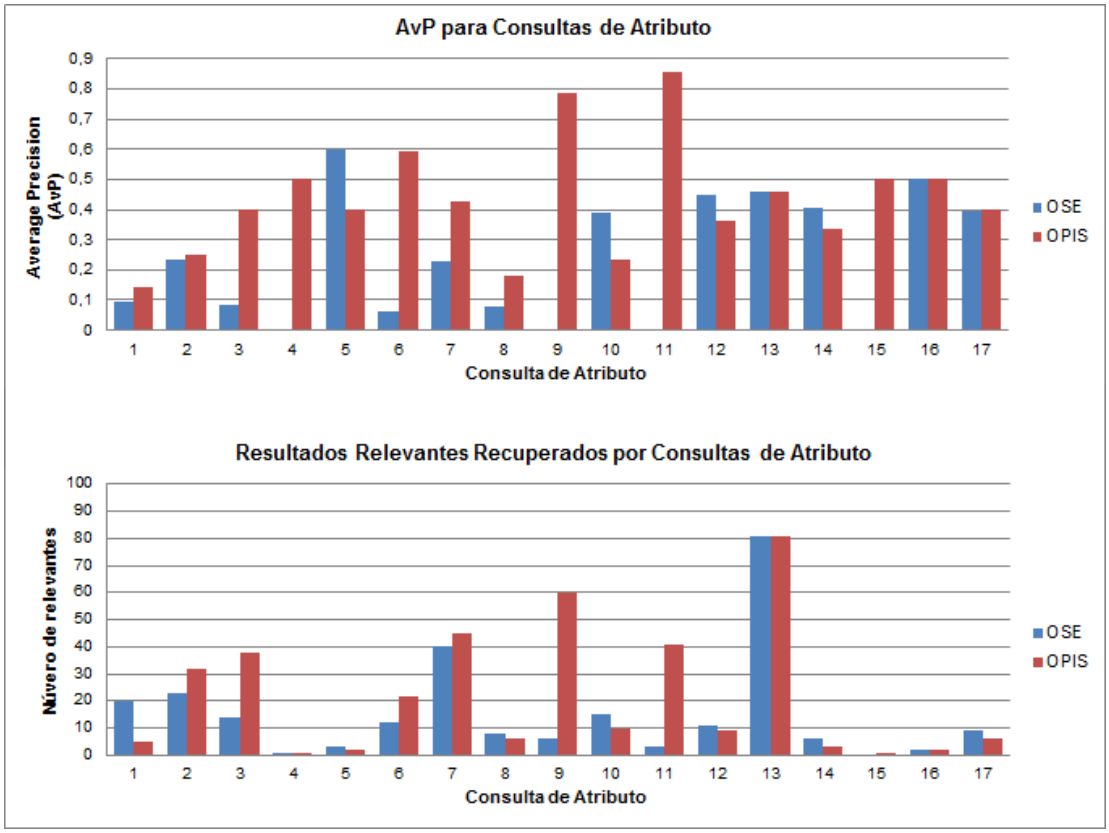

Figura 5.6: Resultados das consultas de atributo para o domínio de *notebook*.

por categoria mantiveram esse comportamento, tendo o OPIS obtido ganhos, em relação ao OSE, de 76% (17 vs. 10) e 43% (21 vs. 14) nas consultas de entidade e atributo, respectivamente, e queda de 18% (42 vs. 52) nas consultas de tipo. Em relação ao número de relevantes recuperados, os resultados mostraram-se semelhantes aos obtidos no caso de estudo anterior, no qual a categoria de tipo (42 vs. 52) apresentou médias maiores, tanto para o OPIS quanto para o OSE, sendo seguida pelas categorias de atributo (21 vs. 14) e entidade (17 vs. 10).

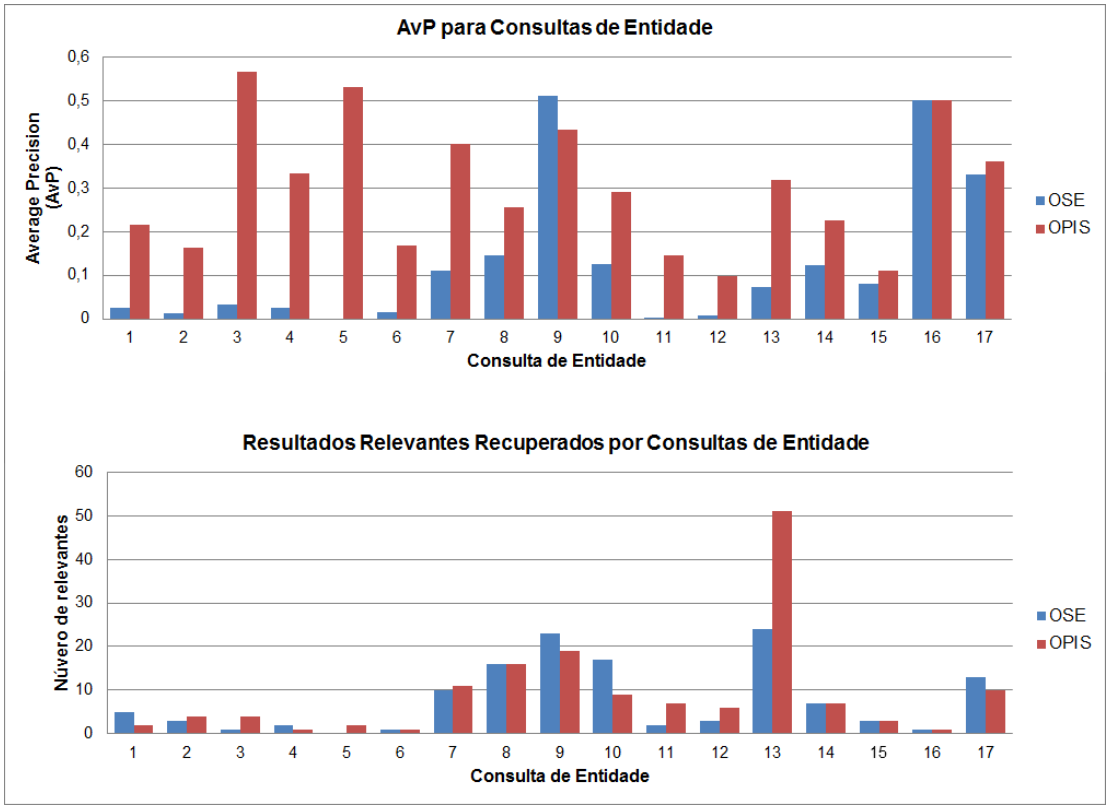

Figura 5.7: Resultados das consultas de entidade para o domínio de câmera digital.

Nas Figuras 5.7, 5.8 e 5.9 são apresentados os gráficos para o estudo de caso do domínio de câmera digital. Pode-se observar que o OPIS obteve resultados de AvP maiores ou equivalentes aos do OSE na maioria das consultas da categoria de entidade, perdendo apenas em um caso (consulta 9). Já nas consultas de tipo e de atributo, o OPIS foi superado pelo *baseline* em oito e seis consultas, respectivamente. Porém, em muitos desses casos, os valores de AvP foram muito próximos, não caracterizando perdas perceptíveis para o usuário. Além disso, nas três categorias, mas principalmente nas de entidade e atributo, é possível notar que, em muitas consultas superadas pelo OPIS, as diferenças de AvP são bastante expressivas, mostrando que o OSE apresentou dificuldade ao atender tais consultas de forma satisfatória. Uma possível explicação para isso se encontra no fato de o domínio de câmera compartilhar alguns atributos (como marca, modelo e preço) com o de *notebook*. Para uma consulta sobre o atributo marca, que restrinja os campos de preço para $999.00, por exemplo, uma página relacionada a uma câmera digital profissional pode ser recuperada, prejudicando o *ranking* e a AvP da consulta para o *baseline*. Considerando as MAPs das três categorias, o OPIS obteve ganhos de 141% (0.300 vs. 0.125), 28% (0.378 vs. 0.294) e 60% (0.303 vs. 0.189) para as consultas de entidade, tipo e atributo, respectivamente. Nota-se que, embora o ganho médio tenha sido menor para

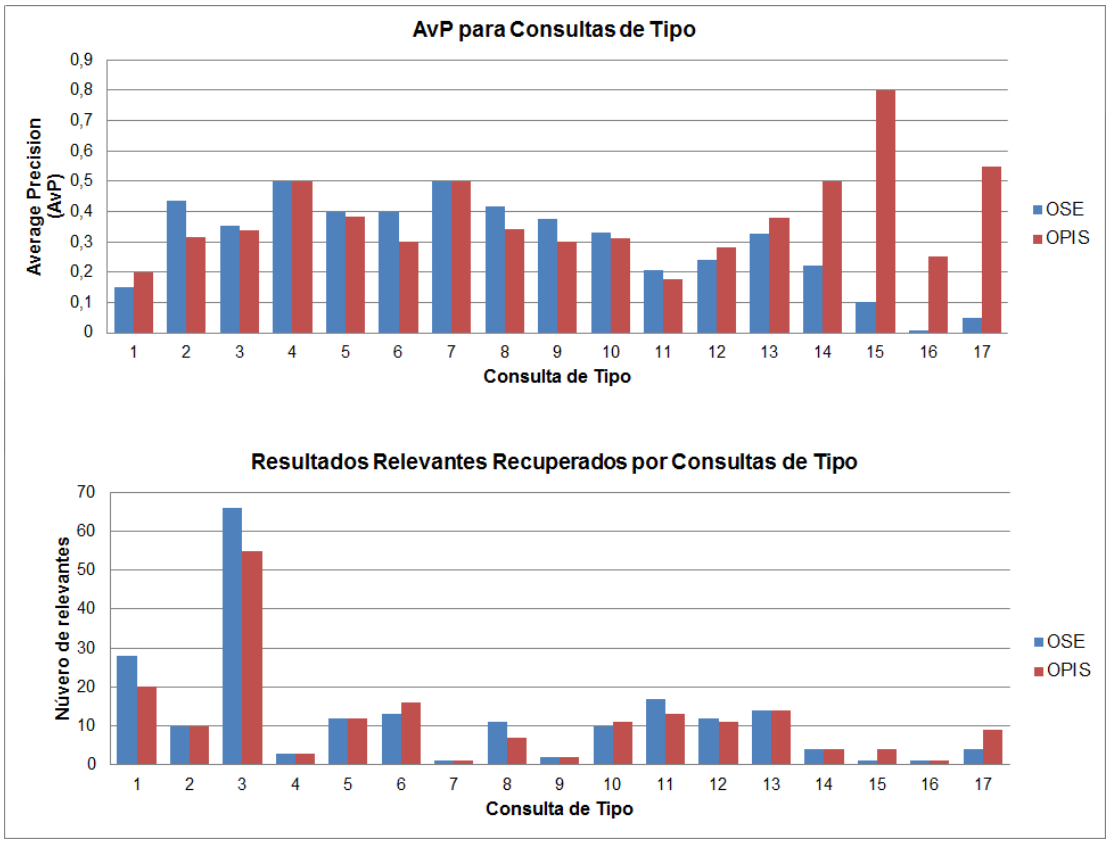

Figura 5.8: Resultados das consultas de tipo para o domínio de câmera digital.

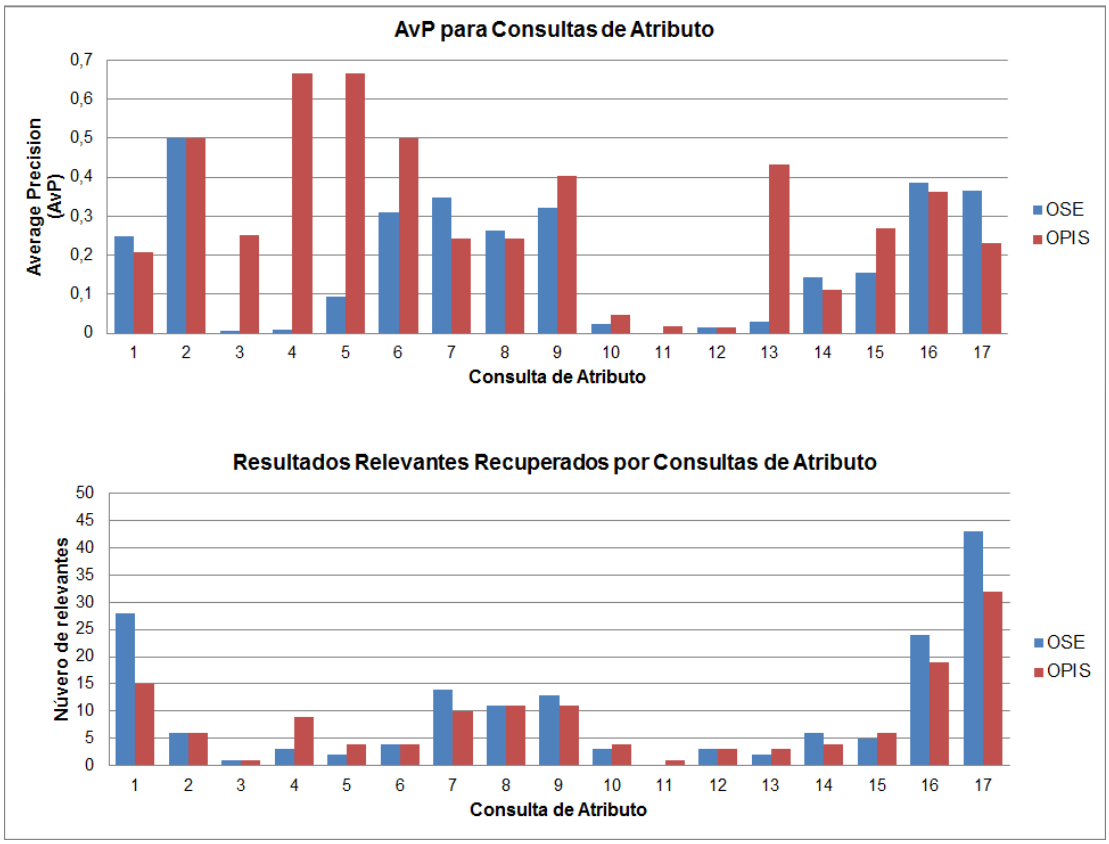

Figura 5.9: Resultados das consultas de atributo para o domínio de câmera digital.

a categoria de tipo, foi nessa categoria, novamente, que as consultas obtiveram precisões maiores, atendendo de forma mais satisfatória as necessidades do usuário ao apresentar páginas relevantes melhor posicionadas no *ranking*. Quanto ao número de relevantes retornados, a média do OPIS foi maior apenas para a categoria de entidade, na qual obteve um ganho de 18% (9 vs. 7), tendo apresentado perdas de 8% (11 vs. 12) e 15% (8 vs. 9) nas categorias de tipo e atributo, respectivamente, as quais não representam, exatamente, perdas, uma vez que a diferença de apenas um relevante pode ser ocasionada por um falso negativo de página-objeto, totalmente aceitável ao se considerar o uso de classificação no processo de filtragem.

Analisando de forma mais geral os resultados, percebe-se que não há uma variação direta entre a quantidade de relevantes retornados e a precisão média obtidas pelas consultas, uma vez que a precisão média não depende apenas da quantidade de relevantes retornados, mas também do posicionamento destes no *ranking*. Consultas que recuperam poucos relevantes podem ter uma AvP alta, o que indica que esses poucos relevantes apresentaram-se bem posicionados, da mesma forma que consultas que recuperam muitos relevantes podem apresentar queda na AvP, indicando que alguns desses relevantes foram prejudicados no *ranking*, o que não compromete, obrigatoriamente, o desempenho da busca.

Tabela 5.3: MAPs para as 51 consultas de cada domínio.

	Pesquisador	***Notebook***	**Câmera Digital**
OSE	0.221	0.291	0.203
OPIS	0.305	0.417	0.327
% de ganho	**38%**	**43%**	**61%**

Faz-se importante observar também, de acordo com a Tabela 5.3, que o OPIS melhorou a precisão média da maioria das consultas para todos os domínios de casos de estudo. Considerando as MAPs de cada domínio separadamente (conjunto de 51 consultas), o OPIS apresenta melhoras de 38% (0.305 vs. 0.221), 43% (0.417 vs. 0.291) e 61% (0.327 vs. 0.203) para os domínio de pesquisador, *notebook* e câmera digital, respectivamente, quando comparados ao *baseline*. Por meio do Teste T, aplicado às consultas de cada domínio, pode-se afirmar que o OPIS melhora significativamente (valor p do Teste T < 0.01) a precisão média das buscas-objeto em todos os domínios, quando comparado ao OSE. Isso significa que o OPIS conseguiu aprender os domínios utilizados por meio das páginas rotuladas (15 para exemplos positivos e 15 para exemplos negativos de páginas-objeto). Dessa forma, a etapa de filtragem permitiu que somente páginas classificadas como páginas-objeto para os domínios considerados fossem mantidas no *ranking*, melhorando suas posições e, consequentemente, atendendo de forma mais satisfatória às necessidades dos usuários para esse tipo de busca.

Tabela 5.4: MAPs para as 51 consultas de cada categoria.

	Entidade	**Tipo**	**Atributo**
OSE	0.159	0.341	0.215
OPIS	0.354	0.349	0.346
% de ganho	**123%**	**3%**	**61%**

Considerando os resultados de MAPs para as consultas de cada categoria, apresentados na Tabela 5.4, o OPIS obteve ganhos em todos os casos, sendo as melhoras mais

expressivas nas categorias de entidade e atributo. Isso é compreensível uma vez que, como foi visto na Figura 5.5, o OSE superou o OPIS na categoria de tipo do domínio de *notebook*, o que influenciou na média das AvPs dessa categoria, tornando-a menor. Antes da realização dos experimentos, esperava-se que o OPIS apresentasse perda na categoria de atributo, uma vez que o OSE executa busca estruturada sobre os atributos dos domínios, enquanto o OPIS realiza busca por palavras-chave. Porém, os resultados do OPIS se mostraram positivamente competitivos, o que pode ser explicado pelo fato de que, mesmo que as palavras-chave de uma consulta apareçam em páginas irrelevantes, se essas páginas não apresentarem conteúdo similar ao de páginas-objeto, elas serão descartadas pelo OPIS. Além disso, as características usadas em uma função de *ranking* pelo OSE podem ser similares às características de outros domínios, o que pode ocasionar a recuperação de páginas de domínios indesejados. Isso é mais difícil de ocorrer com o OPIS, uma vez que ele considera quase todos os termos das páginas e não apenas um conjunto limitado de características, o que reduz a ambiguidade dos domínios.

Considerando a MAP de todas as consultas (conjunto de 153 consultas), independente de domínio e categoria, o ganho obtido pelo OPIS em relação ao OSE é de 47% (0.350 vs. 0.238). Por meio do Teste T, pode-se afirmar novamente que o OPIS obteve ganho de precisão significativo (valor p do Teste T < 0.01) em relação ao *baseline*, o que indica que, em geral, o OPIS atendeu de forma mais satisfatória às necessidades dos usuários, considerando buscas-objeto.

6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou o OPIS, um método de identificação e busca de páginas-objeto, proposto a fim de atenuar o problema apresentado pelos motores de busca convencionais ao responder buscas-objeto. Para a identificação, o OPIS adota técnicas de realimentação de relevância, pré-processamento de texto e aprendizagem de máquina na tarefa de classificação baseada em conteúdo de páginas web. Um modelo de classificação é criado para cada novo domínio através da ajuda de um usuário, que fornece exemplos de páginas-objeto, mas não precisa selecionar um subconjunto de características para representar o domínio, o que reduz seu esforço e nível de especialidade. O OPIS não descarta o uso de GSEs e, ao invés disso, em sua etapa de busca, propõe a integração de um classificador a um GSE, adicionando uma etapa de filtragem ao processo de busca convencional. Essa simples abordagem permite que somente páginas identificadas (classificadas) como páginas-objeto sejam recuperadas pelas consultas dos usuários, atendendo, assim, mais satisfatoriamente suas necessidades quanto a buscas-objeto. As principais contribuições desse método estão na melhoria da precisão média dos resultados de buscas-objeto e na redução do esforço gasto pelos usuários para este fim.

Experimentos foram realizados com a finalidade de avaliar a influência que o OPIS exerce sobre os resultados de buscas-objeto. Nessa avaliação, foram considerados dados reais, pertencentes os domínios de pesquisador, *notebook* e câmera digital, e o *baseline* OSE, que também visa a melhoria dos resultados de buscas-objeto. Foram criadas, para cada domínio, 51 buscas-objeto, divididas entre as categorias de entidade, tipo e atributo. Essas consultas foram submetidas e tiveram suas 100 primeiras páginas recuperadas avaliadas, tanto no OPIS quanto no *baseline*. Os resultados mostram que o OPIS forneceu um ganho médio de 47% de MAP, em relação ao OSE, obtendo, mais especificamente, os ganhos de 38%, 43% e 61% para os domínios de pesquisador, *notebook* e câmera, respectivamente. Por meio do Teste T, pode-se afirmar que o OPIS melhora significativamente a precisão média das buscas-objeto, quando comparado ao *baseline*, atendendo de forma mais adequada às necessidades dos usuários para esse tipo de consulta. Além disso, faz-se importante ressaltar que o OPIS reduz o nível de especialidade do usuário e seus esforços durante o treinamento dos domínios.

Ao longo deste trabalho, foram realizadas duas publicações, listadas a seguir:

1. COLPO; MANICA; GALANTE. **OPIS: Um método para a identificação e a busca de páginas-objeto**. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 28., Recife, PE. Proceedings... SBC, 2013. p.103-108. (SBBD).

2. COLPO; MANICA; GALANTE. **OPIS: Um método para identificação e busca de páginas-objeto apoiado por realimentação de relevância e classifi-**

cação de páginas web. In: WORKSHOP DE TESES E DISSERTAÇÕES EM BANCO DE DADOS, 12., Recife, PE. Proceedings... SBC, 2013. p.8-14. (WTDBD/SBBD).

O primeiro artigo apresenta uma proposta inicial do OPIS, na qual o processo de filtragem era realizado antes da tarefa de indexação, e de experimentos preliminares. O amadurecimento dessa proposta é apresentado no segundo artigo, acompanhado da experimentação apresentada na Seção 5.1. A proposta descrita no último artigo é a mesma apresentada nesta dissertação, tendo o processo se tornado mais genérico, por meio da aplicação da filtragem em tempo de consulta e da introdução de realimentação de relevância no treinamento dos domínios, o que também reduziu a participação do usuário.

Um artigo completo, abrangendo os resultados finais da dissertação, os quais consideram o OSE como *baseline*, foi escrito com o objetivo de ser submetido ao periódico SIGMOD Record.

Como trabalhos futuros, podem ser introduzidas melhorias ao OPIS a partir dos seguintes caminhos:

- Incorporar aprendizagem ativa ao treinamento, permitindo que o *ranking* de páginas recuperado pelas consultas de treinamento seja atualizado, de modo a ofertar os resultados com, possivelmente, maior informatividade primeiro e, assim, acelerar os processos de rotulação e aprendizagem. Uma abordagem semelhante pode ser encontrada no trabalho de DAL BIANCO et al. (2013), sendo, porém, aplicada ao contexto de deduplicação;

- Utilizar mineração de uso para, a partir do comportamento de *clicks* do usuário, induzir o rótulo das páginas de treinamento, de forma indireta e transparente ao usuário. Uma página posicionada no início do *ranking* de uma busca-objeto e que não é clicada pelo usuário, por exemplo, pode ser adicionada ao conjunto de treinamento como um página não objeto, uma vez que se pode inferir que o usuário, ao ler o título e a pequena descrição dessa página, já a tenha considerado irrelevante;

- Desenvolver um método para detectar quais consultas por palavras-chave são efetivamente buscas-objeto, de forma similar ao que é feito pelos motores de busca para detectar consultas dos tipos navegacional, transacional e informacional, o que possibilitaria que as buscas-objeto fossem tratadas, automaticamente, de forma especial;

- Introduzir métodos mais robustos na extração do conteúdo das páginas, como os de redução de ruído e de ponderação orientada à estrutura da página.

REFERÊNCIAS

APACHE. **Apache Lucene**. Disponível em: <http://www.lucene.apache.org>. Acesso em: novembro 2013.

APACHE. **Apache Tomcat**. Disponível em: <http://www.tomcat.apache.org>. Acesso em: novembro 2013.

ASSIS, G. T. de et al. A Genre-Aware Approach to Focused Crawling. **World Wide Web**, [S.l.], v.12, n.3, p.285–319, May 2009.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. Boston: Addison-Wesley, 1999. 513p.

BENNETT, P. N.; SVORE, K.; DUMAIS, S. T. Classification-Enhanced Ranking. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 19., Raleigh, USA. **Proceedings...** New York: ACM Press, 2010. p.111–120.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific American**, [S.l.], v.284, n.5, p.34–43, May 2001.

BLANCO, L. et al. Supporting the automatic construction of entity aware search engines. In: ACM WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT, 10., New York, USA. **Proceedings...** ACM Press, 2008. p.149–156. (WIDM '08).

CAMPOS, R. N. T.; DIAS, G. **Agrupamento Automático de Páginas Web Utilizando Técnicas de Web Content Mining**. 2005. Dissertação (Mestrado em Ciência da Computação) — Departamento de Informática, Universidade da Beira Interior, Covilhã.

CHAKRABARTI, S.; BERG, M. van den; DOM, B. Focused crawling: a new approach to topic-specific web resource discovery. In: WORLD WIDE WEB, New York, USA. **Proceedings...** Elsevier North-Holland: Inc., 1999. p.1623–1640. (WWW '99).

CHANG, C.-C.; LIN, C.-J. LIBSVM: a library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, New York, USA, v.2, n.3, p.27:1–27:27, May 2011.

CHANG, C.-C.; LIN, C.-J. **LIBSVM – A library for support vector machines**. Disponível em: <http://www.csie.ntu.edu.tw/ cjlin/libsvm/>. Acesso em: novembro 2013.

CHENG, G.; QU, Y. Integrating Lightweight Reasoning into Class-Based Query Refinement for Object Search. In: ASIAN SEMANTIC WEB CONFERENCE, 3., Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2008. p.449–463. (ASWC '08).

CHOI, B.; YAO, Z. Web Page Classification. In: CHU, W.; LIN, T. Y. (Ed.). **Foundations and Advances in Data Mining**. [S.l.]: Springer-Verlag, 2005. p.221–274. (Studies in Fuzziness and Soft Computing, v.180).

COLPO, M. P.; MANICA, E.; GALANTE, R. OPIS: Um método para a identificação e a busca de páginas-objeto. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 28., Recife, PE. **Proceedings...** SBC, 2013. p.103–108. (SBBD).

COLPO, M. P.; MANICA, E.; GALANTE, R. OPIS: Um método para identificação e busca de páginas-objeto apoiado por realimentação de relevância e classificação de páginas web. In: WORKSHOP DE TESES E DISSERTACÕES EM BANCO DE DADOS, 12., Recife, PE. **Proceedings...** SBC, 2013. p.8–14. (WTDBD/SBBD).

DAL BIANCO, G. et al. Tuning Large Scale Deduplication with Reduced Effort. In: INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT, 25., New York, NY, USA. **Proceedings...** ACM, 2013. p.18:1–18:12. (SSDBM).

GOOGLE. **Google Custom Search API**. Disponível em: <https://developers.google.com/custom-search>. Acesso em: outubro 2013.

GOSSET, W. S. The probable error of a mean. **Biometrika**, [S.l.], v.6, n.1, p.1–25, 1908.

GUO, X. et al. On the Class Imbalance Problem. In: INTERNATIONAL CONFERENCE ON NATURAL COMPUTATION, 2008., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2008. p.192–201. (ICNC '08, v.4).

HANANI, U.; SHAPIRA, B.; SHOVAL, P. Information Filtering: overview of issues, research and systems. **User Modeling and User-Adapted Interaction**, Hingham, USA, v.11, n.3, p.203–259, Aug. 2001.

HRESKO, J. Web Page Multi-label ClassifiCation for Filtering Content from the Web. In: RUSSIR YOUNG SCIENTISTS CONFERENCE, 6., Yaroslavl, Russia. **Proceedings...** [S.l.: s.n.], 2012. p.16–23. (RuSSIR 2012).

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. **A Practical Guide to Support Vector Classification**. Taipei 106, Taiwan: Department of Computer Science – National Taiwan University, 2010. Disponível em: <http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf>. Acesso em: novembro 2013.

ISKE, P.; BOERSMA, W. Connected brains: question and answer systems for knowledge sharing - concepts, implementation and return on investment. **Journal of Knowledge Management**, [S.l.], v.9, n.1, p.126–145, 2005.

JI, L. et al. ExSearch: a novel vertical search engine for online barter business. In: ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 18., New York, USA. **Proceedings...** ACM Press, 2009. p.1357–1366. (CIKM '09).

JOACHIMS, T. Text Categorization with Suport Vector Machines: learning with many relevant features. In: EUROPEAN CONFERENCE ON MACHINE LEARNING, 10., London, UK, UK. **Proceedings...** Springer-Verlag, 1998. p.137–142. (ECML '98).

KANG, C. et al. Learning to rank with multi-aspect relevance for vertical search. In: ACM INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING, New York, USA. **Proceedings...** ACM Press, 2012. p.453–462. (WSDM '12).

KAPTEIN, R. et al. Entity ranking using Wikipedia as a pivot. In: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 19., New York, USA. **Proceedings...** ACM Press, 2010. p.69–78. (CIKM '10).

KLYNE, G.; CARROLL, J. J. **Resource Description Framework (RDF)**: Concepts and Abstract Syntax. [S.l.]: World Wide Web Consortium (W3C), 2004. Disponível em: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Acesso em: novembro 2013.

LOPEZ, S.; SILVA, J. A New Information Filtering Method for WebPages. In: WORKSHOPS ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, 2010., Washington, USA. **Proceedings...** IEEE Computer Society, 2010. p.32–36. (DEXA '10).

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. New York, USA: Cambridge University Press, 2008.

MARINHO, L. B.; GIRARDI, R. Mineração na Web. **Revista Eletrônica de Iniciação Científica**, [S.l.], v.3, n.2, June 2003.

METEREN, R.; SOMEREN, M. Using Content-Based Filtering for Recommendation. In: ECML/MLNET WORKSHOP ON MACHINE LEARNING AND THE NEW INFORMATION AGE, Barcelona, Espanha. **Proceedings...** [S.l.: s.n.], 2000. p.47–56.

MIKLÓS, Z. et al. From web data to entities and back. In: ADVANCED INFORMATION SYSTEMS ENGINEERING, 22., Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2010. p.302–316. (CAiSE'10).

NAKATANI, M.; JATOWT, A.; TANAKA, K. Adaptive ranking of search results by considering user's comprehension. In: INTERNATIONAL CONFERENCE ON UNIQUITOUS INFORMATION MANAGEMENT AND COMMUNICATION, 4., New York, USA. **Proceedings...** ACM Press, 2010. p.27:1–27:10. (ICUIMC '10).

NAVADIYA, D.; PATEL, R. Web Content Mining Techniques –A Comprehensive Survey. **International Journal of Engineering Research & Technology**, [S.l.], v.1, n.10, p.1–6, Dec. 2012.

NETSCAPE. **Open Directory Project**. Disponível em: <http://www.dmoz.org/>. Acesso em: outubro 2013.

NIE, Z. et al. Web object retrieval. In: WORLD WIDE WEB, 16., New York, USA. **Proceedings...** ACM Press, 2007. p.81–90. (WWW '07).

ORACLE. **JavaServer Pages Technology**. Disponível em: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>. Acesso em: novembro 2013.

OSWALD, D. et al. **HTML Parser**. Disponível em: <http://htmlparser.sourceforge.net>. Acesso em: outubro 2013.

PAGE, L. et al. **The PageRank Citation Ranking**: bringing order to the web. [S.l.]: Stanford InfoLab, 1999. Technical Report, Previous number = SIDL-WP-1999-0120. (1999-66).

PALME, J. **Information Filtering**. Disponível em: <http://www.dsv.su.se/ jpalme/select/information-filtering.pdf>. Acesso em: outubro 2013.

PHAM, K. C. et al. Object search: supporting structured queries in web search engines. In: NAACL HLT 2010 WORKSHOP ON SEMANTIC SEARCH, Stroudsburg, USA. **Proceedings...** Association for Computational Linguistics, 2010. p.44–52. (SS '10).

POUND, J.; MIKA, P.; ZARAGOZA, H. Ad-hoc Object Retrieval in the Web of Data. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 19., New York, NY, USA. **Proceedings...** ACM, 2010. p.771–780. (WWW '10).

QI, X.; DAVISON, B. D. Web page classification: features and algorithms. **ACM Computing Surveys**, New York, USA, v.41, n.2, p.1–31, Feb. 2009.

RAJAN, S. et al. A large-scale active learning system for topical categorization on the web. In: WORLD WIDE WEB, 19., New York, USA. **Proceedings...** ACM Press, 2010. p.791–800. (WWW '10).

RIBONI, D. Feature Selection for Web Page Classification. In: EURASIA-ICT WORKSHOP ON WEB CONTENT MAPPING: A CHALLENGE TO ICT, 1., Tehran, Iran. **Proceedings...** Australian Computer Society, 2002.

SANTOS, R. D. C. Conceitos de Mineração de Dados na Web. In: MINICURSOS DO XV SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB, Fortaleza, CE. **Anais...** [S.l.: s.n.], 2009. v.1, p.41–80.

SNOWBALL. **Snowball Portuguese Stop Word List**. Disponível em: <http://snowball.tartarus.org/algorithms/portuguese/stop.txt>. Acesso em: outubro 2013.

SNOWBALL. **Snowball English Stop Word List**. Disponível em: <http://snowball.tartarus.org/algorithms/english/stop.txt>. Acesso em: dezembro 2013.

UNIVERSITY OF WAIKATO. **Weka Data Mining Software API**. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka>. Acesso em: outubro 2013.

WANG, H. et al. Personalized ranking model adaptation for web search. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 36., New York, USA. **Proceedings...** ACM Press, 2013. p.323–332. (SIGIR '13).

WEB TRANSLATOR JAVA. **Web Translator Java API**. Disponível em: <http://sourceforge.net/projects/webtranslator>. Acesso em: novembro 2013.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining**: practical machine learning tools and techniques. 3.ed. [S.l.]: Morgan Kaufmann, 2011.

WIVES, L. K. **Técnicas de descoberta de conhecimento em textos aplicadas à inteligência competitiva**. 2002. Exame de Qualificação — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

YAMAMOTO, T.; NAKAMURA, S.; TANAKA, K. Reranking and Classifying Search Results Exhaustively Based on Edit-and-Propagate Operations. In: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, 20., Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2009. p.855–862. (DEXA '09).

YANG, H.; CHUA, T.-S. Web-based list question answering. In: COMPUTATIONAL LINGUISTICS, 20., Stroudsburg, USA. **Proceedings...** Association for Computational Linguistics, 2004. (COLING '04).

ZAREH BIDOKI, A. M. et al. A3CRank: an adaptive ranking method based on connectivity, content and click-through data. **Information Processing and Management**, Tarrytown, USA, v.46, n.2, p.159–169, Mar. 2010.

ZHENG, Z. et al. A regression framework for learning ranking functions using relative relevance judgments. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 30., New York, USA. **Proceedings...** ACM Press, 2007. p.287–294. (SIGIR '07).