Combinando Informações Textuais e Estruturais na Recuperação de Documentos Web.

Roberta de Souza Coelho 1,2

Marcelo Nery dos Santos 2

Silvio Romero Lemos Meira 2

Resumo: A *World Wide Web* consiste numa imensa coleção de documentos distribuídos, heterogêneos e dinâmicos que são acessados por usuários inexperientes detentores de um comportamento imprevisível e longe do ideal. Estas características criam um conjunto de novos desafios para os sistemas de recuperação de informação (SRI). Diferentemente das coleções de documentos "*flat*", a Web possui informações auxiliares que vão além do conteúdo textual, tais como, a estrutura dos hiperlinks e o texto dos hiperlinks. Este trabalho propõe utilizar as informações embutidas na estrutura dos hiperlinks, juntamente com informações textuais no momento do ranqueamento dos documentos Web com o objetivo de melhorar a qualidade da resposta fornecida por um SRI.

Abstract: The World Wide Web is a dynamic, heterogeneous, massive and distributed collection of documents which is accessed by inexperienced users with unpredictable and less ideal querying behavior. The characteristics of the Web environment creates many new challenges to the Information Retrieval Systems (IRS). At the same time, unlike “flat” document collections, the WWW is hypertext and provides auxiliary information on top of the text of Web pages, such as link structure and link text. We take advantage of the link structure of the Web to produce a global "importance" ranking to every Web page indexed by a search engine, to be used in combination with textual ranking scores to improve the performance of the SRI.

1 Mobile, Av. Caxangá,5775 - Várzea - Recife - PE. CEP: 50740-000
{rsc@mobile.com.br}
2 Centro de Informática, UFPE, Caixa Postal 7851
{rsc, mns, srlm @cin.ufpe.br}

Palavras-Chaves: *Web Mining*, HITS, *PageRank*, qualidade da resposta.

1 Introdução

A *World Wide Web* (ou simplesmente Web) consiste num imenso repositório de informações que divergem em conteúdo, qualidade e tempo de vida. As características das informações presentes na Web trazem consigo novos desafios relacionados à recuperação de informações neste ambiente. Diante deste cenário, surgiram os sistemas de recuperação de informação (SRI) para Web também chamados de engenhos de busca[1] como uma alternativa para auxiliar o usuário a encontrar informações relevantes na Web.

Além das dificuldades relacionadas ao processo de busca já enfrentadas pelos sistemas de recuperação de informação tradicionais, tais como, a sinonímia (dois termos com o mesmo significado) e a polissemia (um termo com mais de um significado), os engenhos de busca enfrentam os seguintes problemas:

(i) Desafio da Precisão[3]: este desafio é enfrentado quando para uma dada consulta existe uma enorme quantidade de documentos na Web. Nestes casos, os engenhos de busca enfrentam o desafio de retornar em primeiro lugar apenas os documentos mais "importantes" para a consulta submetida.

(ii) Desafio da Cobertura: este desafio é enfrentado quando para uma dada consulta existe uma pequena quantidade de documentos na Web. Estas consultas são chamadas de consultas por tópicos específicos. Nestes casos, os engenhos de busca enfrentam o desafio de tentar encontrar os documentos que respondem à consulta entre os milhares de documentos existentes na base de dados do engenho de busca.

(iii) Técnicas de *Spamming:* consistem em técnicas utilizadas na construção de páginas Web, com o intuito de manipular as suas funções de *ranking* dos engenhos de busca [12]. Mais adiante explicaremos o papel da função de *ranking* em um engenho de busca.

Segundo pesquisas, cerca de 70% das consultas submetidas aos engenhos de busca contêm somente um termo [15], e cerca de 85,2% dos usuários visitam apenas a primeira página de resultados [6]. Tal fato nos leva a constatar que o problema da precisão é constantemente enfrentado pelos engenhos de busca.

Diferentemente de outras coleções textuais, os documentos Web são conectados por hiperlinks. Os hiperlinks possuem diferentes utilidades [7]: (i) **estrutural** - permitem ao usuário navegar pelas páginas de um site, ou de um domínio específico; (ii) **funcional -** conectam *sites* de diferentes domínios, que possuem conteúdos relacionados, e

[3] A precisão é a métrica mais utilizada para a avaliação de sistemas de recuperação de informação e consiste na fração dos documentos retornados que são relevantes.

possivelmente "importantes", segundo a opinião do autor do documento origem do link; (iii) **comercial -** links de propaganda presentes nos *banners*.

Existem vários algoritmos na literatura, que propõem a análise da estrutura de links funcionais de uma coleção de documentos Web com o objetivo de extrair desta estrutura a opinião coletiva dos usuários da Web. O algoritmo proposto neste trabalho é baseado nos trabalhos realizados pelo projeto CLEVER da IBM [5], pelo projeto Google desenvolvido na Universidade Stanford [14] que hoje corresponde a um engenho de busca comercial, e pelo laboratório de pesquisas em Web Archaeology da Compaq [3]. Estes grupos têm demonstrado algumas das contribuições que a análise da estrutura de links pode trazer para sistemas que lidam com documentos Web. O algoritmo desenvolvido neste trabalho propõe coletar as principais características dos trabalhos acima citados, como também generalizar os resultados obtidos por estes algoritmos.

As informações extraídas através dos algoritmos de análise de links podem ser utilizadas para vários propósitos, que vão além do propósito citado acima, quais sejam: (i) identificar comunidades na Web - as comunidades são definidas por um conjunto de páginas que se auto-referenciam e abordam um tópico especifico [8]; (ii) encontrar páginas similares a uma página Web - funcionalidade "páginas parecidas" encontrada em alguns engenhos de busca [16]; (iii) identificar a reputação de uma página na Web [13]; (iv) elaborar políticas de *crawling* [18]; (v) e classificar documentos Web[6].

Este artigo está organizado como segue: na Seção 2 é mostrada uma arquitetura simplificada de engenho de busca. Na Seção 3 são mostrados os principais algoritmos de análise de links existentes na literatura. A Seção 4 detalha o algoritmo proposto neste trabalho. A Seção 5 apresenta os módulos que devem ser acrescentados à arquitetura de um engenho de busca, para permitir o armazenamento e a análise da estrutura de links. Na Seção 6 são apresentados os resultados obtidos durante os testes que avaliaram a eficácia de recuperação de estratégias de busca que utilizaram o peso de "importância", gerado pelo algoritmo proposto neste trabalho, como componente da função de *ranking*. E por fim, na Seção 7 são apresentadas as conclusões acerca dos resultados obtidos e na Seção 8 são enumerados alguns direcionamentos futuros para este trabalho.

2 Arquitetura de um Engenho de Busca

Muitos dos engenhos de busca utilizam uma arquitetura *Crawler-Indexer* centralizada [19] que é ilustrada na Figura 1 de forma simplificada.

O processo de *crawling* e indexação se inicia a partir do momento em que o Servidor de Links envia um conjunto de links para os *crawlers* – estes links, enviados inicialmente, são chamados de links semente (passo 1, Figura 1). Os *crawlers* são programas que executam localmente, enviando requisições HTTP a servidores Web remotos. As páginas Web retornadas em resposta a estas requisições HTTP são enviadas para o Servidor de Indexação (passos 2 e 3, Figura 1).

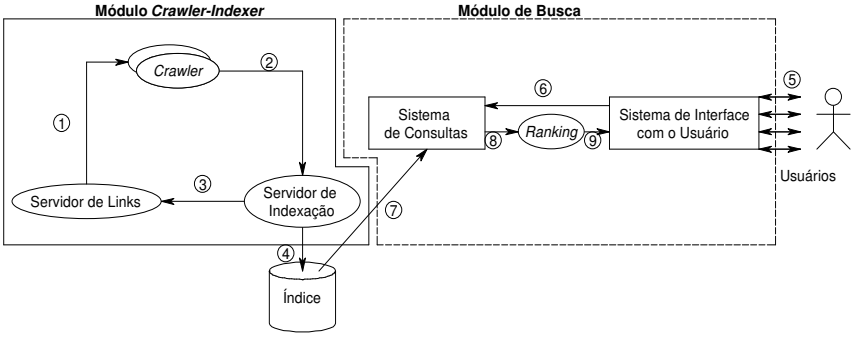

Figura 1: Arquitetura simplificada de um engenho de busca.

O Servidor de Indexação é responsável por extrair os termos e os links das páginas. Os termos são armazenados em estruturas chamadas de índice (passos 4, Figura 1). Existem várias estruturas de índice, porém a mais utilizada é a estrutura de arquivos invertidos [19]. Os arquivos invertidos são estruturas de dados que permitem encontrar de forma rápida quais documentos de uma coleção possuem um dado termo.

Os links extraídos pelo Servidor de Indexação são enviados ao Servidor de Links que irá repassá-los aos *crawlers* (passo 3, Figura 1).

O processo de busca tem início quando o usuário submete uma consulta ao Sistema de Interface com o Usuário - esta consulta corresponde a um conjunto de palavras-chaves (passo 5, Figura 1). O Sistema de Consultas é responsável por verificar nas bases de índices quais são as páginas que possuem os termos consultados (passos 6 e 7, Figura 1). As páginas encontradas, chamadas de páginas resposta, são enviadas para o sistema de *Ranking*.

O sistema de *Ranking* é responsável por calcular, para cada documento, uma medida quantitativa que denota a probabilidade de relevância de cada documento em relação à consulta submetida. Após este cálculo, o sistema de *Ranking* ordena as páginas em ordem decrescente de acordo com esta medida e retorna estas páginas para o Sistema de Interface com o Usuário, que é responsável pela exibição dos links resposta. Este processo de ordenação é chamado de ranqueamento.

A maioria dos engenhos de busca analisa as páginas Web como um documento texto simples, não levando em consideração a estrutura na qual a página Web está inserida, isto é, o sistema de *Ranking* possui informações textuais como único critério de relevância no momento de se ordenar as páginas Web [4].

A seguir serão mostrados alguns algoritmos que analisam esta estrutura de links, associando um peso de "importância" as páginas Web analisadas. Este peso pode ser utilizado pelo sistema de *Ranking* juntamente com informações textuais no momento de ordenar as páginas resposta. A utilização deste peso possibilita aos engenhos de busca o

retorno de páginas que além de relevantes à consulta são consideradas importantes para tópico consultado, segundo a opinião coletiva dos usuários embutida na estrutura de links.

3 Algoritmos de Análise de Links

Os algoritmos de análise de links podem ser divididos em duas famílias: algoritmos independentes da consulta e algoritmos dependentes da consulta. A seguir serão abordados os principais algoritmos pertencentes a cada uma destas famílias.

3.1 Algoritmos Independentes da Consulta

Os algoritmos independentes da consulta ou globais associam uma medida de importância intrínseca a cada página pertencente à base de índices de um engenho de busca, sem levar em consideração informações textuais das páginas ou informações originadas a partir de uma consulta específica.

PageRank

Page et al. [14] definiram um algoritmo de análise de links que associa um *ranking* global às páginas Web, chamado PageRank, o qual tenta capturar a noção de "importância" de uma página. Por exemplo, a página da folhaonline é intuitivamente mais procurada que a página de um grupo de pesquisas em Mecânica Celeste da UFPE. Esta diferença pode estar refletida no número de outras páginas que apontam (*backlinks*) para estas duas páginas.

O algoritmo PageRank leva em consideração além do número de *backlinks* que uma página possui, a importância das páginas que apontam para esta página. Logo, uma página que é apontada pela folhaonline receberá maior importância, do que outra página que é apontada por uma página desconhecida. Uma definição simples do PageRank (r) que captura a idéia mostrada acima é apresentada na fórmula (1):

$$r(i) = d * \sum_{j \in B(i)} r(j) / N(j) + (1-d)/m, \qquad (1)$$

Onde:

- d representa uma variável calculada empiricamente;
- B(i) representa o conjunto de *backlinks* da página i;
- N(j) representa o número de links presentes no corpo da página j (*forwardlinks*);
- m é o número total de páginas existentes no subconjunto da Web analisado.

O Google utiliza o valor PageRank juntamente com valores gerados por algoritmos de busca textuais para responder as consultas por palavras-chaves submetidas ao engenho.

3.2 Algoritmos Dependentes da Consulta

Esta família de algoritmos associa uma medida de importância a cada página pertencente ao conjunto de documentos Web retornados em resposta a uma consulta.

Hypertext Induced Topic Search (HITS)

O algoritmo PageRank baseia-se na noção de que páginas importantes são aquelas que são apontadas por outras páginas importantes. Em muitos casos, porém, os relacionamentos existentes entre as páginas Web sugerem um diferente tipo de modelo. Se considerarmos, por exemplo, as home pages dos principais engenhos de busca. Apesar de todas elas serem importantes, elas não se auto referenciam. Tal fato decorre da simples razão de que os engenhos de busca competem entre si. O algoritmo HITS consiste em uma maneira de reconhecer a importância destas home pages, através da análise de páginas anônimas que contém listas de links que apontam para elas (*bookmarks)*.

O algoritmo HITS, proposto por Kleinberg [11], associa a cada página dois valores de *ranking* dependentes da consulta: o peso de autoridade e o peso de *hub*.

A idéia básica do algoritmo HITS consiste em identificar um pequeno subgrafo (S) da Web - subgrafo formado pelas URLs retornadas em resposta a uma consulta, e por páginas que apontam e são apontadas por páginas deste conjunto - e aplicar o algoritmo de análise de links a este subgrafo de forma a identificar as páginas *hubs* e autoridades deste subgrafo. Uma página é considerada boa autoridade se é apontada por muitos bons *hubs*, e uma página é considerada bom *hub* se aponta para muitas boas autoridades. A Figura 2 representa graficamente os conceitos de *hub* e autoridade.

Figura 2: Representação gráfica de uma página *hub* e uma página autoridade.

O HITS associa um peso de autoridade *a* e um peso de *hub h* para cada página pertencente ao subgrafo S. Inicialmente os pesos de *hub* e autoridade são inicializados com pesos arbitrários. O algoritmo de análise de links é um algoritmo iterativo que desempenha dois tipos de operação em cada iteração, a operação O, e a operação I. Na operação I, o peso de autoridade de cada página pertencente ao grafo é atualizado de forma a receber o somatório dos pesos de *hub* de todas as páginas que apontam para ela. Na operação O, o peso de *hub* de cada página recebe o somatório dos pesos das páginas autoridade que são apontadas por ele. Estas operações são representadas pelas Fórmulas (2) e (3), onde B(i) representa o conjunto de *backlinks* da página i e F(i) representa o número de *forwardlinks* da pagina i.

$$a(i) = \sum_{j \in B(i)} h(j), \quad (2)$$

$$h(i) = \sum_{j \in F(i)} a(j), \quad (3)$$

Pode-se perceber que uma página pode ser ao mesmo tempo um bom *hub* e uma boa autoridade. O algoritmo repete iterativamente as operações I e O, realizando a cada iteração a normalização dos pesos de autoridade e *hub*, até que os pesos de *hub* e autoridade cheguem a convergir, isto é, até que estes pesos não variem acima de um valor pré-determinado.

Todo este processamento, que leva cerca de alguns minutos, é realizado no momento da consulta. Por esta razão o HITS não atende ao requisito de tempo imposto pelos engenhos de busca comerciais que é de poucos segundos. Algumas extensões do algoritmo HITS que adicionaram análise de conteúdo à análise puramente estrutural foram desenvolvidas pelo projeto CLEVER da IBM [18], e pelo grupo de pesquisas em Web Archaeology da Compaq [3].

4 *Algoritmo Proposto*

O algoritmo proposto neste trabalho consiste em uma extensão do algoritmo HITS. Contrariamente ao HITS, o algoritmo proposto se aplica uma única vez para todo o grafo da Web indexada, associando pesos globais de autoridade e *hub* para cada página do grafo Web. Isto o torna mais resistente às técnicas de *spamming* baseadas em links, detalhadas em [7]. Este algoritmo adiciona novas características ao HITS tais como a adição de vetores de personalização às operações I e O, e a aplicação de um conjunto de heurísticas de limpeza ao processo de geração do grafo base. Tomando como base as características enumeradas acima este algoritmo foi denominado de Global Hybrid HITS (GHHTIS).

Como, no momento da consulta, os pesos de *hub* e autoridade já foram pré-computados, o engenho de busca não requerer nenhum tempo adicional para a análise de links, obedecendo ao requisito de tempo imposto a estes sistemas que é de poucos segundos. Os pesos de *hub* (h) e autoridade (a) são calculados de acordo com a relação de reforço mútuo, através de sucessivas iterações das equações mostradas a seguir.

$$a(i) = InitAut(i) + \sum_{j \in B(i)} h(j) * aut\_wt(j,i), \quad (4)$$

$$h(i) = InitHub(i) + \sum_{i \in F(i)} a(j) * hub\_wt(i,j), \quad (5)$$

Onde:
- (i,j) representa um link partindo da página i e chegando a página j;
- B(i) corresponde ao conjunto de *backlinks* da página i;
- F(i) corresponde ao conjunto de *forwardlinks* de i;
- aut_wt e hub_wt são pesos atribuídos aos links com o objetivo de minimizar a relação de reforço mútuo entre os *hosts*. A utilização destes pesos foi elaborada pela extensão do algoritmo HITS desenvolvida pela Compaq [2];
- InitAut e InitHub são vetores de personalização.

Os pesos aut_wt e hub_wt ponderam a influência de cada *host* no cálculo dos pesos de *hub* e autoridade. A Figura 3 mostra dois cenários, no Cenário 1 é atribuído um peso aut_wt = ¼ aos arcos, que serão responsáveis por reduzir a influência do *host* A no cálculo do peso de autoridade da página existente no host B. O Cenário 2 mostra uma situação similar para o cálculo do peso de hub de uma página do host A.

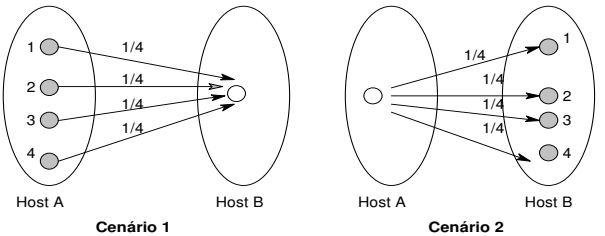

Figura 3: Associando pesos aos arcos.

A computação de *hubs* e autoridades termina quando a ordem induzida pelo vetor de pesos de autoridade ao longo de 10 iterações varia abaixo de um limite pré-estabelecido. Tal critério de parada foi estabelecido empiricamente. A ordem induzida pelo vetor de pesos também é utilizada como critério de convergência pelo algoritmo PageRank [14] o qual, da mesma forma que o GHHITS, é aplicado uma única vez a todo o grafo da Web indexada.

Data Cleaning

As heurísticas de *Data Cleaning*, também chamadas de heurísticas de limpeza, atuam extraindo os links não funcionais do subgrafo Web analisado. Os links não funcionais podem provocar ruídos que podem levar à degradação da eficácia de recuperação do engenho que utilize pesos calculados pelo GHHITS.

Neste trabalho, foi utilizada uma heurística para evitar a autopromoção, isto é, páginas que conferem peso de autoridade para páginas do mesmo site. Esta heurística descarta os links intrínsecos durante a construção do grafo Web. De acordo com esta heurística, duas páginas são consideradas do mesmo site (intrínsecas) quando: os *hosts* são similares, isto é, quando o domínio contido nas URLs é o mesmo, ou quando os IPS são similares. Existem

três casos de similaridade entre IPS, quais sejam: (i) quando as duas páginas pertencem às classes IP A ou B e os dois octetos mais significativos coincidem; (ii) quando as duas páginas pertencem à classe IP C e os 3 octetos mais significativos coincidem; e (iii) quando as duas páginas pertencem à classe IP D e todos os octetos coincidem.

5 Arquitetura Estendida de um Engenho de Busca

Para verificarmos o impacto na qualidade da resposta causado pela utilização de informações estruturais em conjunto com as informações textuais, foi necessária a implementação de um sistema que viabilizasse o armazenamento e a análise da estrutura de links da Web. Chamamos este sistema de SAAL – Sistema para Armazenamento e Análise de Links. O SAAL estende a arquitetura *Crawler-Indexer* centralizada a partir da inclusão de um módulo para armazenamento e análise de links. A Figura 4 ilustra esta arquitetura estendida.

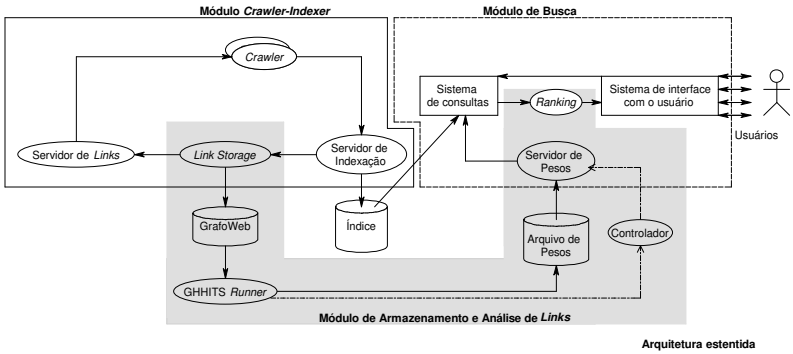

Figura 4: Arquitetura estendida de um engenho de busca.

Uma visão mais detalhada sobre cada componente pertencente ao módulo de análise e armazenamento de links é apresentada a seguir.

Link Storage

O Link Storage é responsável por receber e armazenar informações sobre a estrutura de links existente entre as páginas Web indexadas e por aplicar a heurística de limpeza.

Grafo Web

O Grafo Web contém a estrutura de links da Web indexada. Estes links são armazenados em estruturas similares aos arquivos invertidos [19], concebidos para armazenar as bases de índices dos engenhos de busca. A Figura 5 mostra, de forma

simplificada, como estão estruturadas as duas estruturas responsáveis pelo armazenamento de links. O arquivo de *forwardlink* faz o mapeamento entre uma URL e os links apontados por ela, e o arquivo de *backwardlink* realiza o mapeamento entre uma URL e as URLs que apontam para ela.

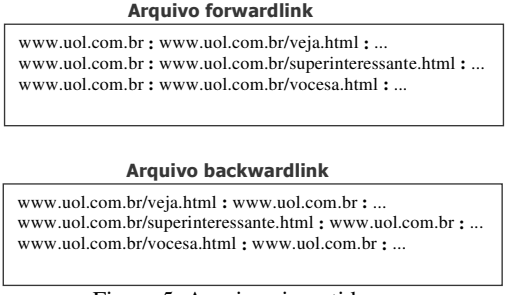

Figura 5: Arquivos invertidos.

GHHITS Runner

Este componente implementa o algoritmo GHHITS, sendo responsável por executá-lo sobre um subconjunto da Web indexada (Grafo Web).

Arquivo de Pesos

O Arquivo de Pesos, gerado pelo algoritmo de análise de links, corresponde a um arquivo binário que contém o peso de autoridade e o peso de *hub* para cada página participante da computação do GHHITS.

Servidor de Pesos

O Servidor de Pesos é um servidor remoto que atende às requisições do engenho de busca. Estas requisições solicitam o peso de autoridade e/ou *hub* de uma dada página.

Controlador

Este componente é responsável por receber informações de controle e repassá-las ao Servidor de Pesos. Estas informações de controle compreendem as informações de *start*, *stop* e *reload* do Servidor de Pesos. A diretiva *reload* é utilizada para alertar ao Servidor de Pesos que um novo Arquivo de Pesos foi gerado.

Ranking

Para que os pesos de autoridade passassem ser usados pelo engenho de busca no momento do ranqueamento da resposta, foram necessárias alterações no componente de

Ranking deste sistema. A fórmula de ranqueamento presente no componente de Ranking foi alterada, passando a combinar os paradigmas textual e estrutural. Estes paradigmas estão representados na fórmula de ranqueamento através da medida de similaridade textual e da medida de importância respectivamente. Estas medidas são descritas a seguir:

(i) Medida de similaridade: obtida através do algoritmo de busca baseado no paradigma textual. Este algoritmo procura associar um peso de similaridade entre um documento e uma consulta. No engenho de busca Radix, utilizado como estudo de caso, este algoritmo utiliza o modelo booleano estendido [19] acrescido de heurísticas de proximidade entre os termos. A representação dos documentos é feita a partir de todos os termos do conteúdo, do título e da URL do documento, excluindo-se as stop-words - as palavras que aparecem com muita freqüência nos documentos da coleção (ex.: artigos e preposições).

(ii) Medida de importância: Corresponde ao peso de autoridade[4] calculado através do algoritmo GHHITS.

A fórmula de ranqueamento passou a ser representada através da seguinte fórmula de fusão de informação:

$$\text{Ranking} = (a * \text{Norm\_SimTextual} + b*\text{Norm\_AUTORIDADE}) \quad \textbf{(6)}$$

Onde:
- Ranking corresponde ao valor final atribuído ao documento.
- Norm_SimTextual é o valor de similaridade originado pelo algoritmo de busca baseado no paradigma textual, normalizado pelo maior valor de similaridade que pode ser retornado por este algoritmo.
- Norm_AUTORIDADE: corresponde ao peso de autoridade originado pelo algoritmo GHHITS, baseado no paradigma estrutural, normalizado pelo maior valor de autoridade que pode ser retornado por este algoritmo.
- a e b correspondem a constantes cujos valores foram calculados empiricamente.

6 Resultados do Experimento

O algoritmo GHHITS foi aplicado a um subconjunto da Web contendo 3.742.983 páginas (11,14% da Web brasileira em janeiro de 2002), que corresponde a um subconjunto das páginas indexadas pelo engenho de busca Radix - www.radix.com.br.

Este subconjunto de páginas indexadas deu origem a um grafo composto por 85.648.933 nós, que correspondem às páginas efetivamente indexadas e as páginas que são referenciadas no corpo destas páginas. Após a aplicação das heurísticas de limpeza sobre este grafo, ele ficou reduzido a um subconjunto contendo 2.882.853 nós. A partir da análise deste

[4] Experimentos prévios não obtiveram melhoria na qualidade da resposta com a utilização dos pesos de *hub*. Por esta razão, o peso de *hub* não foi utilizado neste experimento.

subgrafo e dos pesos de autoridade gerados pelo GHHITS quando aplicado a este subgrafo, puderam ser percebidas as seguintes características:

- 1.384.398 páginas (27,69 % das páginas) possuíam links externos ao seu site;
- 35,13% das páginas possuíam *backlinks* de sites diferentes do site ao qual pertenciam;
- As páginas com maior peso de autoridade são páginas que possuem baixa profundidade no site, geralmente, são as páginas *home* do site;
- As páginas com maior peso de autoridade no subgrafo Web analisado são páginas ".com". O site www.cin.ufpe.br/~rsc/GHHITS/cemMais.html traz uma lista contendo os 100 maiores *hubs* e autoridades encontrados neste experimento.

Este experimento teve como objetivo avaliar as estratégias de busca enumeradas a seguir. Estas estratégias utilizam a fórmula de *ranking* descrita na Seção 5, com variações nos valores dos coeficientes a e b: Estratégia Textual (TEXT); Peso de Autoridade (AUT); 0.85 x Estratégia Textual + 0.15 x Peso de Autoridade (0.85TEXT_0.15AUT); 0.75 x Estratégia Textual + 0.25 x Peso de Autoridade (0.75TEXT_0.25AUT); 0.65 x Estratégia Textual + 0.35 x Peso de Autoridade (0.65TEXT_0.35AUT).

A coleção de testes, utilizada neste experimento, correspondeu ao subconjunto das páginas indexadas pelo Radix contendo 3.742.983 documentos, descrito acima.

Para avaliarmos a utilização do peso de autoridade no ranqueamento, foi utilizada a técnica de julgamento de relevância cego [5]. Nesta técnica, (i) o usuário, através de um sistema de avaliação, submete a consulta ao mesmo tempo para todas as estratégias de busca participantes da avaliação, (ii) as respostas retornadas pelas estratégias são armazenadas em uma única lista de URLs que são ordenadas randomicamente, e submetidas para avaliação, (iii) nesta avaliação, o usuário classifica as páginas como relevantes ou irrelevantes, sem saber qual estratégia foi responsável por retornar a página avaliada.

Neste processo de avaliação foram utilizadas 100 consultas testes que foram coletadas dos *logs* de acesso do engenho de busca Radix. Este processo necessitou de 10 voluntários, que foram responsáveis por avaliar as páginas retornadas por 10 consultas deste conjunto. O julgamento dos voluntários foi armazenado na forma de uma tripla (página, julgamento, consulta) para posteriormente ser utilizado no cálculo das métricas: precisão@10, PMTS@10. A precisão média TREC Style (PMTS@n) [10] é calculada dividindo-se o somatório dos valores da precisão - calculados em cada ponto onde um documento relevante é retornado - por *n*, que corresponde ao número de elementos que estão sendo avaliados para uma dada consulta. A precisão relativa (precisão@n) [9] consiste na precisão calculada para um conjunto contendo os *n* primeiros documentos retornados na lista de links resposta.

As Tabelas 1 e 2 mostram os valores das métricas PMTS@10 e precisão@10 obtidos para cada uma das estratégias durante este experimento e os valores da comparação relativa (CR-TEXT) e da comparação absoluta (CA_TEXT) com relação à estratégia TEXT.

Tabela 1: Valor da métrica PMTS@10 para as cinco estratégias avaliadas.

Estratégia	PMTS@10	CA-TEXT	CR-TEXT
TEXT	0,332	-	-
AUT	0,166	-16,6%	-50,05%
0.85TEXT_0.15AUT	0,368	3,6%	10,74%
0.75TEXT_0.25AUT	0,381	4,9%	14,64%
0.65TEXT_0.35AUT	0,366	3,4%	10,13%

Tabela 2: O Valor da precisão@10 para as cinco estratégias avaliadas.

Estratégia	precisão@10	CA-TEXT	CR-TEXT
TEXT	0,478	-	-
AUT	0,284	-19,4%	- 40,52%
0.85TEXT_0.15AUT	0,516	3,9%	8,13%
0.75TEXT_0.25AUT	0,523	4,5%	9,42%
0.65TEXT_0.35AUT	0,509	3,2%	6,65%

Ao se analisar os valores das métricas PMTS@10 e precisão@10 percebe-se que as estratégias que utilizaram a fusão de paradigmas, 0.85TEXT_0.15AUT, 0.75TEXT_0.25AUT, 0.65TEXT_0.35AUT obtiveram uma significativa melhora com relação à estratégia TEXT. As três estratégias enumeradas acima não chegaram a atribuir um coeficiente maior que 50% (b>=0,5) para o peso de autoridade, pois o objetivo desta fusão de paradigmas foi utilizar o GHHITS como estratégia auxiliar ao *ranking* baseado no paradigma textual, e não com o intuito de substituir este paradigma.

A única estratégia analisada na qual foi atribuído um valor de b maior que 0,5 foi a estratégia AUT (b=1, a=0). Por utilizar as informações estruturais contidas nos links como única fonte de informação, desprezando o conteúdo da página. Esta estratégia mostrou quedas expressivas nos valores das métricas coletadas com relação à estratégia TEXT. Este resultado já era esperado uma vez que ao se ranquear as páginas apenas por critérios estruturais, perdem-se características importantes das páginas, tais como: o *tfidf* [17]; a distância entre os termos; e a distância entre o termo consultado e o início do documento.

A estratégia 0.75TEXT_0.25AUT foi responsável pelas melhorias mais expressivas nos valores das métricas analisadas. Esta estratégia representa, portanto, o equilíbrio entre as medidas de importância e de provável relevância da fórmula de fusão no cenário avaliado neste experimento. Portanto, o objetivo deste trabalho foi atingido, pois se conseguiu alterar a função de *ranking* de um engenho de busca de forma que este passasse a retornar páginas que além de relevantes eram consideradas "importantes" para o tópico consultado.

7 Conclusões

A análise da estrutura de links de um subconjunto contendo 11,4% da Web brasileira nos permitiu detectar que as maiores autoridades da Web brasileira são páginas .com. Este comportamento nos fornece um indício para concluirmos que as páginas .com são consideradas mais "importantes" pelos usuários da Web brasileira do que muitas páginas do domínio.br. Os resultados obtidos neste trabalho demonstraram que a análise das informações contidas nos links, através do GHHITS, pode ser utilizada em conjunto com uma estratégia puramente textual para melhorar a eficácia de recuperação do engenho de busca. Os bons resultados obtidos até o momento servem de motivação para o estudo sobre soluções mais complexas que utilizem heurísticas de limpeza mais sofisticadas e que levem em consideração o conteúdo dos links no momento do cálculo dos pesos de *hub* e autoridade.

8 Direcionamentos Futuros

Os trabalhos futuros que podem ser desenvolvidos a partir deste envolvem: (i) a comparação entre a qualidade da resposta gerada pelo algoritmo GHHITS aplicado a todo o grafo da Web indexada e o algoritmo HITS aplicado ao grafo resposta - caso a qualidade da resposta gerada pelo HITS se mostre muito melhor que a gerada pelo GHHITS, o HITS poderia ser utilizado para a construção de uma *cache* contendo as consultas submetidas com mais freqüência ao engenho de busca; (ii) a personalização dos vetores de *hub* e autoridade gerados por este algoritmo - os vetores InitAut e InitHub existentes no algoritmo GHHITS para representar a opinião dos usuários, possuíram todas as entradas iguais a 0 em todas as estratégias avaliadas, propõe-se como trabalho futuro, a utilização de diferentes valores de entradas para os vetores InitAut e InitHub de forma a conferir menos ou mais importância a algumas páginas Web segundo a opinião de um usuário ou de uma comunidade de usuários da Web.

Referências

[1] A. Arasu, H. Garcia-Molina, J. Cho, A. Paepcke, and S. Raghavan, Searching the Web, *ACM Transactions on Internet Technology*, 1(1), 2--43, 2001.

[2] C. Buckley and E. Voorhees, Evaluating evaluation measure stability, *Proc. of the 23rd International ACM SIGIR*, pp. 33--40, 2000.

[3] K. Bharat and M. R. Henzinger, Improved Algorithms for Topic Distillation in Hyperlinked Environments, *Proc. of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR), 104--111, 1998.

[4] Search Engine Watch: www.searchenginewatch.com

[5] S. Chakrabarti, B. Dom, D. Gibson, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, Experiments in topic distillation, *ACM SIGIR workshop on Hypertext Information Retrieval on the Web*, 1998.

[6] C. Silverstein, M. Henzinger, H. Marais and M. Moricz, Analysis of a very large Web search engine query log, *SIGIR Forum*, 33(1), 6-12, 1999.

[7] B. D. Davison, Recognizing nepotistic links on the Web, *Artificial Intelligence for Web Search Technical Report*,WS-00-01, 23--28, 2001.

[8] G. W. Flake, S. Lawrence, and C. L. Giles, Efficient identification of Web communities, *Proc. of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000)*, 150--160, 2000.

[9] M. Gordon, P. Pathak, Finding information on the world wide Web: The retrieval effectiveness of search engines, *Information Processing and Management*, 35 (2), 141--180, 1999.

[10] D. Hawking, N. Craswell, P. Bailey, K. Griths, Measuring search engine quality, *Journal of Information Retrieval*, 4, 33--59, 2001.

[11] J. Kleinberg, Authoritative sources in a hyperlinked environment, *Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 668--677, 1998.

[12] R. Lempel and S. Moran, The stochastic approach for link-structure analysis (salsa) and the tkc effect, *Proc. of the 9th World Wide Web Conference*, 387--401, 2000.

[13] A. O. Mendelzon, D. Rafiei, What do the Neighbours Think?, *Computing Web Page Reputations IEEE Data Engineering Bulletin*, 23(3):9--16, 2000.

[14] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the Web, *Stanford Digital Library Working Paper*, SIDL-WP-1999--0120, 1998

[15] S. Lawrence and C. L. Giles, Accessibility of information on the Web, *the journal Nature*, 400, 107-109, 1999.

[16] J. Dean and M. R. Henzinger, Finding Related Web Pages in the World Wide Web, *Proc. of the 8th International World Wide Web Conference (WWW8)*, 389--401, 1999.

[17] K. Yang, Literature Review, *School of Information and Library Science University of North*, 2001

[18] S. Chakrabarti, M. van den Berg, and B. Dom, Focused crawling: a new approach to topic specific Web resource discovery, *Proc. of 8th WWW Conference*, 31 (11-16), 1623--1640, 1999.

[19] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, United States of America, 1999.