Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Projeto de Final de Curso
Projeto e Análise de Algoritmos

Relatório Final

Alberto Hideki Ueda

Orientador: Berthier Ribeiro Neto

Belo Horizonte
1º de dezembro de 2014

Conteúdo

1 O Problema

1.1 Contexto

Coletores de páginas da *Web* constituem o primeiro passo para a implementação de máquinas de busca modernas. De forma geral, um coletor - em inglês, *crawler* - é um sistema que faz requisições a servidores da *Web* de forma planejada e automática, coleta parte ou todo o conteúdo das páginas devolvidas pelas requisições e utiliza este novo conteúdo para realizar novas requisições [**?**]. Estima-se que hoje mais de 10% das visitas a *Websites* sejam feitas por coletores [**?**].

Hoje, mesmo as principais máquinas de busca cobrem apenas uma parte da *Web* atual. Em 2005, foi demonstrado que o nível de cobertura das principais máquinas de buscas existentes está entre 58% e 76% da *Web* [**?**]. Além disso, o custo da utilização da rede também deve ser considerado. Em 2004, tal custo foi estimado em US$ 1,5 milhão para coletores de larga escala [**?**].

1.2 Dificuldade do problema

... Desta forma solução exata para o problema é impraticável.

1.3 Objetivos do Trabalho

A ideia deste trabalho é modificar um algoritmo existente de um coletor genérico (*General Crawler*) e transformá-lo em um coletor temático (*Focused Crawler*) - que concentra-se em um único tópico de interesse - visando aumentar tanto a qualidade quanto a cobertura das páginas coletadas em relação ao algoritmo original.

Mais especificamente, este trabalho consistirá em alterações no escalonamento de longo prazo de um *General Crawler*, direcionando as requisições de *downloads* para páginas relevantes a um tópico pré-estabelecido, por meio de consultas de referência (*driving queries*) ou documentos de exemplo (como um conjunto de páginas *Web*).

1.4 Abordagem Utilizada

O objetivo deste trabalho é transformar o *general web crawler* de referência em um *focused crawler*, melhorando os resultados obtidos pelo coletor original, dado um tema específico. Para isso, modificaremos a estratégia que um coletor genérico utiliza para determinar quais são os próximos *links*

que devem ser processados. No caso do *WIRE*, devemos alterar seu escalonamento de longo-prazo (componente *manager*, conforme seção anterior), substituindo parte do código por um novo algoritmo de escolha das próximas páginas que deverão ser coletadas.

Em seguida, executaremos testes de coleta para os dois algoritmos - original e modificado - e faremos a comparação dos resultados obtidos. Dado um tópico-alvo para a coleta, espera-se que o algoritmo adaptado seja melhor em termos de espaço e memória que o original. Mais especificamente, o *focused crawler* deverá encontrar as páginas mais relevantes antes do *crawler* genérico. Parte das mesmas métricas utilizadas para avaliar o *WIRE* serão aplicadas para o nosso coletor e os resultados serão apresentados.

2 Modelagem através de um Grafo

Considerando as páginas da *Web* como vértices e os diversos *links* encontrados como arestas, pode-se aplicar diferentes algoritmos para a seleção *on-line* (em tempo de execução) dos *links* que o nosso *focused crawler* irá visitar em um nova coleta. Possíveis estratégias de escolha (*selection policy*) para o coletor baseiam-se, por exemplo, em: buscas em largura [**?**], indicador *PageRank* [**?**] ou até mesmo no uso de algoritmos genéticos [**?**].

Levando-se em conta que não é possível coletar todas as páginas da *Web*, devido à enorme quantidade de páginas e sua característica mutável, é importante encontrar as páginas mais relevantes o quanto antes. Quanto mais rápido uma página muito relevante for encontrada (de acordo com o indicador considerado), melhor a qualidade dos resultados obtidos em um dado instante de tempo da execução do coletor.

Logo, considerar a estrutura da *Web* é interessante para a implementação de um *crawler* de alto desempenho. Tal estrutura pode ser analisada sob duas perspectivas: *microscópica* e *macroscópica*.

Sob uma perspectiva microscópica, percebe-se que a estrutura dos *links* não se assemelha a um arranjo aleatório. Ao invés disso, é comum identificar *hubs* e redes *scale-free* [**?**], como ilustrado na figura abaixo.

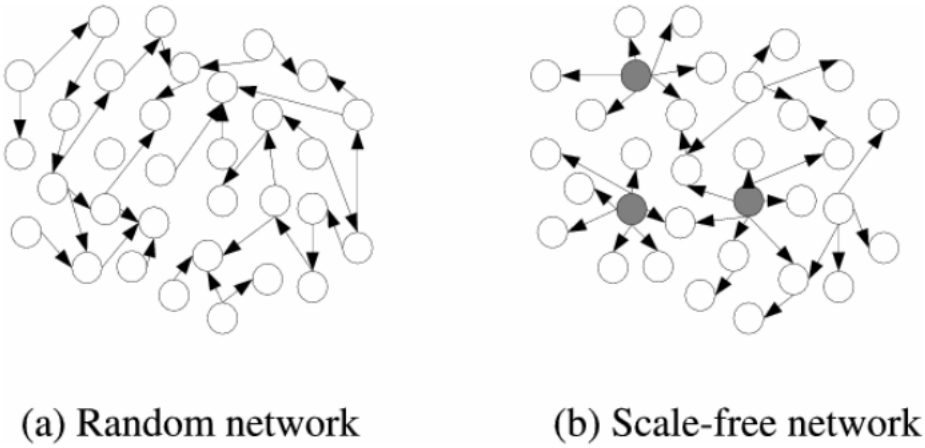

Figura 1: Exemplos de estruturas microscópicas da *Web*. Uma rede gerada aleatoriamente (à esquerda) e uma rede *scale-free* [**?**].

Já sob a perspectiva macroscópica, podemos identificar uma estrutura *bow-tie* [**?**], conforme visto em um dos trabalhos práticos da disciplina. A figura a seguir apresenta uma possível visualização desta estrutura.

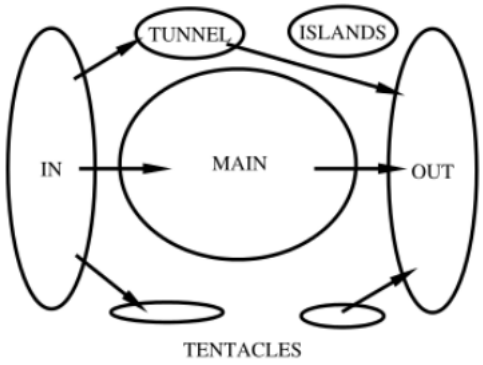

Figura 2: Estrutura macroscópia da *Web* [**?**].

Devido a estas perspectivas, muitos *crawlers* são implementados de modo a beneficiar-se destas informações. Este é o caso do WIRE [**?**], o *general web crawler* que é nosso *baseline* para este trabalho.

De forma geral, os coletores podem ser divididos em diversas categorias, conforme mostra a figura a seguir. Em cada categoria é possível identificar quais critérios são os mais importantes em uma coleta. É importante notar

que um coletor genérico deve equilibrar todos os critérios apresentados, enquanto para um coletor temático a relevância das páginas e o quanto estão atualizadas são os critérios mais importantes, podendo haver detrimento do critério de representação (o quanto a página coletada reproduz o conteúdo original).

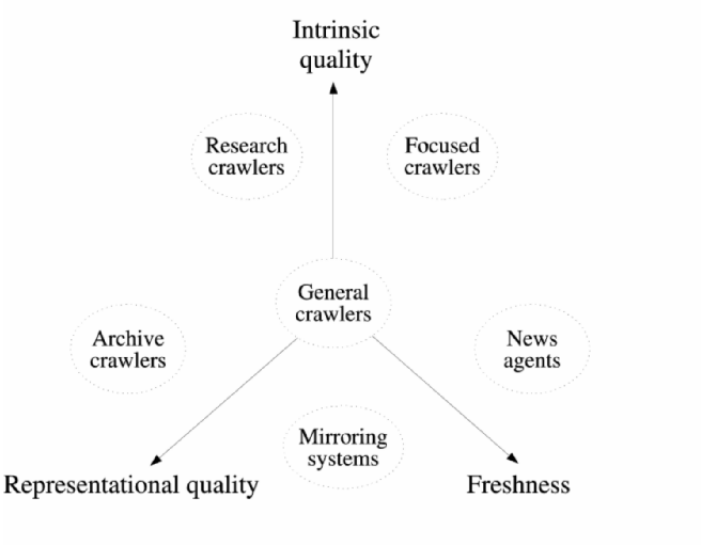

Figura 3: Tipos de coletores [**?**].

3 Baseline

O *baseline* para este trabalho é o *WIRE*, um *general web crawler* desenvolvido por Carlos Castillo em seu doutorado na Universidade do Chile [**?**]. Este coletor foi construído visando-se alto desempenho e boa escalabilidade, respeitando as políticas de restricões a visitas de robôs a servidores *Web*.

O *WIRE* foi desenvolvido para realizar integrações com máquinas de busca, como é comum no caso de coletores *web*. A figura a seguir mostra o ciclo de funcionamento de um sistema completo, desde a coleta até a busca em si.

Figura 4: Ciclo de uma máquina de busca [**?**].

O objetivo principal do *WIRE* é realizar a caracterização da *Web*, a partir dos dados coletados pelo *crawler*. A figura abaixo encontra-se na tese de doutorado em que o *WIRE* foi proposto. Ela mostra trabalhos futuros que podem utilizar este coletor, representados em cinza claro. Nota-se que o objetivo de nosso trabalho (construir um *focused crawler*) está entre os itens citados.

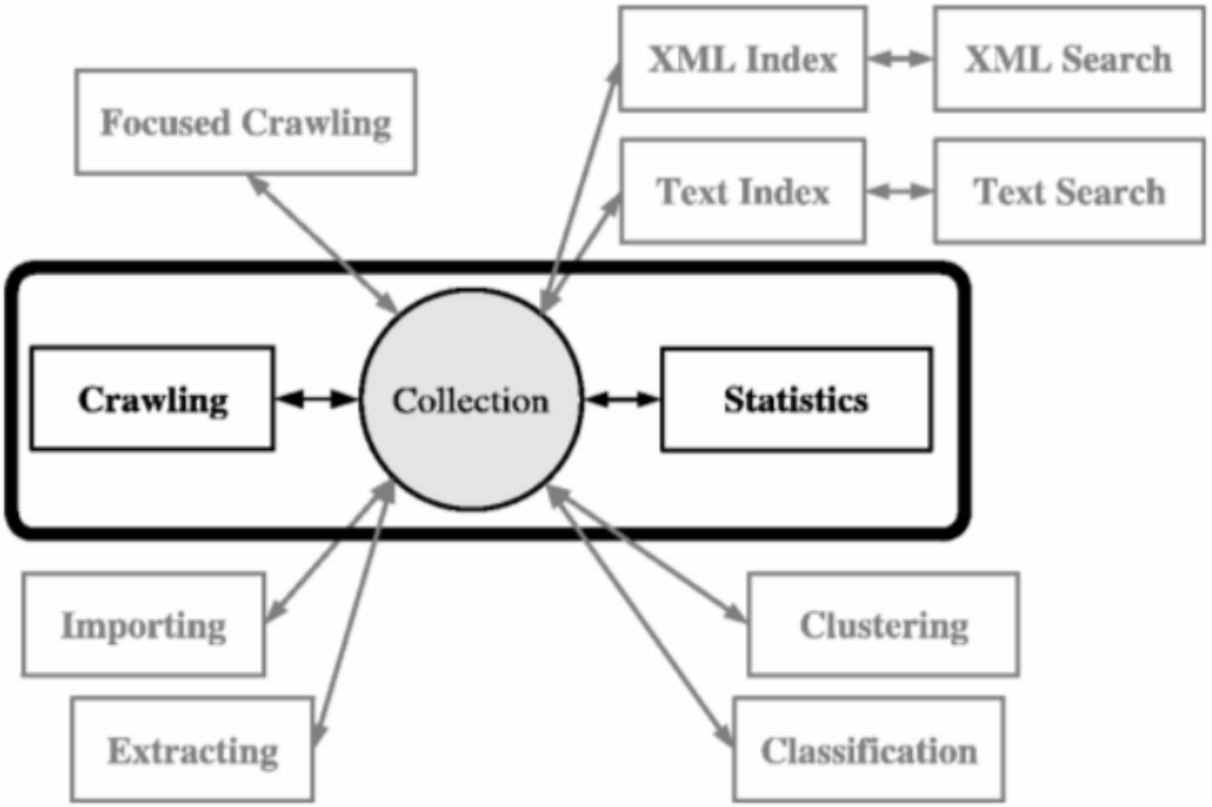

Figura 5: Áreas de atuação do *WIRE* (em preto) e trabalhos futuros (em cinza) [**?**].

3.1 Algoritmo

O algoritmo deste *baseline* é representado em alto nível pela figura a seguir.

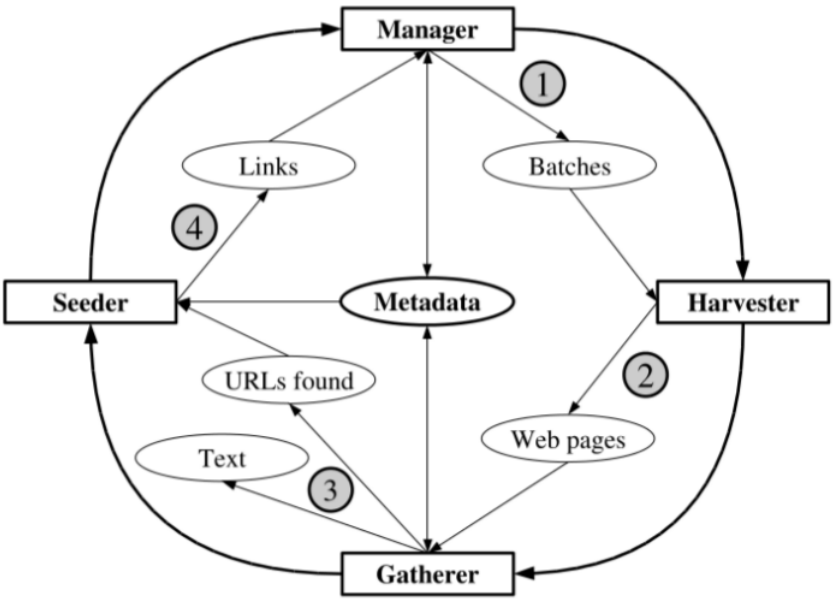

Figura 6: Representação do funcionamento do coletor *WIRE* [**?**].

Cada componente é descrito a seguir:

- *Manager*: realiza os cálculos da relevância de cada página coletada e as atualizações no escalonamento de longo-prazo;
- *Harvester*: responsável pelo escalonamento de curto prazo e pelas coletas de páginas;
- *Gatherer*: faz o *parse* e a extração de *links*;
- *Seeder*: responsável pela *resolução* de *URLs* e atualizações no grafo de *links*;

O algoritmo utiliza listas de adjacências para representar a estrutura de *links* encontrada entre as páginas, podendo ser descrito da seguinte forma:

1. (*Seeder*) A partir de um arquivo inicial de *URLs*, gera-se a lista das próximas páginas que devem ser coletadas;
2. (*Manager*) Se uma página da lista gerada já foi visitada recentemente, a mesma é ignorada;

3. O valor intrínseco de cada página da lista é calculado, por meio de um indicador. Por exemplo, o *PageRank* [**?**];

4. Calcula-se o quanto as páginas estão atualizadas;

5. Com o valor intrínseco e o nível de atualização das páginas, calcula-se a qualidade geral de cada página;

6. Extrai-se então da lista as k páginas de maior qualidade, k determinado em um arquivo de configuração;

7. (*Harvester*) Determina-se como as k páginas serão coletadas (observando regras de "bom comportamento"para coletores);

8. Coleta-se todas as páginas de acordo com as regras determinadas no item anterior;

9. (*Gatherer*) As informações coletadas são tratadas, armazenando-se o que é relevante e descartando o restante do material coletado;

10. São enviadas para o *Seeder* as novas *URLs* encontradas nas páginas coletadas e reinicia-se o processo, agora sem a necessidade de arquivos de inicialização;

A condição de parada depende do objetivo do usuário, que pode priorizar cobertura (continuar até que não haja páginas no domínio de interesse, demandando potencialmente um tempo inviável) ou por critério de tempo (encerra-se quando atingir um nível determinado de cobertura). O estudo do *WIRE* mostrou que a partir de uma coleta com 50% de uma grande coleção de páginas, já é possível atingir 80% do valor total do *PageRank* dessa coleção [**?**].

4 Heurística Implementada

5 Análise de Complexidade dos Algoritmos

5.1 Complexidade de Tempo

5.2 Complexidade de Espaço

6 Análise Experimental dos Algoritmos

6.1 Tempo de Execução

6.2 Uso de Memória

6.3 Qualidade da Resposta

6.4 Melhor Caso

6.5 Pior Caso

7 Trabalhos Futuros

8 Conclusão

Referências

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The concepts and technology behind search.* Pearson Education Limited, 2nd edition, 2011.

[2] Albert-Laszlo Barabasi. Linked the new science of networks, 2002.

[3] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, June 2000.

[4] Carlos Castillo. *Effective Web Crawling.* PhD thesis, School of Engineering, Santiago, Chile, November 2004.

[5] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through url ordering. In *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, pages 161–172, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[6] Nick Craswell, Francis Crimmins, David Hawking, and Alistair Moffat. Performance and cost tradeoffs in web search. In *Proceedings of the Australasian Database Conference ADC2004*, pages 161–170, Dunedin, New Zealand, January 2004. `http://research.microsoft.com/users/nickcr/pubs/craswell_adc04.pdf`.

[7] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, pages 902–903, New York, NY, USA, 2005. ACM.

[8] Judy Johnson, Kostas Tsioutsiouliklis, and C. Lee Giles. Evolving strategies for focused web crawling. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 298–305, 2003.

[9] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the Tenth Conference on World Wide Web*, pages 114–118, Hong Kong, May 2001. Elsevier Science.

[10] J. Nielsen. Statistics for traffic referred by search engines and navigation directories to useit. `http://www.useit.com/about/searchreferrals.html`, 2004.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.