Trabalho Prático 1 - Recuperação da Informação

Matheus Aquino Motta[1]

[1]Bacharelado em Matemática Computacional
DCC - Universidade Federal de Minas Gerais

matheusmotta@dcc.ufmg.br

Abstract. *In this report we will briefly discuss the implementation of the assignment 1 of the subject Information Retrieval. The problem consisted into implementing a simple crawler algorithm to collect URLs and its respective titles using the Chilkat library.*

Resumo. *Nesse relatório iremos discutir brevemente a implementação do Trabalho Prático 1 da disciplina Recuperação da Informação. O problema consistia em desevolver um simples coletor de páginas web para encontrar URLs e seus respectivos títulos utilizando a biblioteca Chilkat.*

1. Introdução

O problema proposto no Trabalho Prático 1 consistia no desenvolvimento de um simples *Crawler*, que deveria ser implementado a partir do uso da biblioteca *Chilkat*, onde seriam dados dois parâmetros de entrada: uma *URL* s e um número inteiro n.

Assim, deveriamos desenvolver um algoritmo que a partir da página inicial s imprimisse $n + 1$ *URLs* de enlaces, *links*, presentes na página incluindo o enlace inicial passado como entrada e seus respectivos títulos, e ao final imprimisse o tempo de *Crawling* médio μT gasto para pesquisar cada um dos enlaces encontrados.

2. Implementação

De um modo geral a implementação do algoritmo consistiu basicamente em um *loop* de repetição que para cada iteração era realizado o *Crawling* do próximo potencial *link URL*.

Desse modo, seja k o número de páginas coletadas com sucesso e t_i o tempo gasto para realizar o *Crawling* da i-ésima página, realizamos o seguinte: Caso seja encontrado um novo enlace l_i com sucesso, imprimimos o seu título e sua respectiva *URL*, e além disso, incrementamos o número de páginas pesquisadas com êxito e armazenamos o tempo de execução do *Crawling*. Caso contrário, imprimimos uma mensagem de erro relativa ao evento ocorrido, i.e, caso não hajam mais páginas a serem pesquisadas ou um erro foi detectado.

Destarte, ao final da execução do *loop* de coleta principal, imprimimos o tempo médio em segundos μT de *Crawling* para cada enlace pesquisado com sucesso.

$$\mu T = \frac{1}{k} \sum_{i=0}^{k-1} t_i$$

Valor esse que é ajustado por um fator de 10^{-6}, haja vista que a função utilizada da biblioteca *chrono* para computar o tempo, considera a execução em microsegundos.

3. Resultados

Foram realizados múltiplos testes para ilustrar o funcionamento do algoritmo em diferentes cenários.

Primeiro cenário, URL inicial *"www.bbc.com"* e $n = 20$,

Successfully crawled links: 21
Total crawling execution time(s): 29.1106
Average crawling execution time(s): 1.38622

Segundo cenário, URL inicial *"www.goodreads.com"* e $n = 20$,

Successfully crawled links: 21
Total crawling execution time(s): 54.0347
Average crawling execution time(s): 2.57308

Terceiro cenário, URL inicial *"www.ufmg.br"* e $n = 20$,

Successfully crawled links: 21
Total crawling execution time(s): 8.4789
Average crawling execution time(s): 0.403757

A diferença no tempo de execução do *Crawling* para os diferentes cenários pode ser explicada por características particulares dos três sítios. O primeiro e segundo sítio possuem hospedagem em domínios não brasileiros, o que leva a uma latência e tempo de requisição superior ao do terceiro que possui uma hospedagem local.

Além disso, é notável que o primeiro sítio possui abrangência e número de acessos superiores ao segundo, haja vista que é um jornal mundialmente conhecido, enquanto o segundo consiste em uma rede social de nicho. Assim, o primeiro sítio potencialmente possui um servidor mais adaptado e otimizado para o acesso de usuários.

Os *outputs* completos com as *URLs* coletadas e seus respectivos títulos podem ser encontrados na pasta *Resultados*, assim como o código fonte comentado utilizado na pasta principal.

4. Conclusões

A partir desse trabalho foi possível ter um primeiro contato com coletores *Web*, que irão dar base para atividades futuras, e que por sua vez são parte fundamental de máquinas de busca.

Ademais, entender e analisar resultados para diferentes sítios e cenários foi interessante para perceber a complexidade e o comportamento de *Crawlers* na *Web* em variados contextos.

5. Apêndice

```cpp
#include <CkSpider.h>
#include <iostream>
#include <chrono>
#include <string>

using namespace std;

void pageDisplay(int idx, string url, string title){
    if (idx == 0) cout << "\n";
    cout << "Page " << idx << "\n";
    cout << "Title: " << title << "\r\n";
    cout << "URL: " << url    << "\r\n" << "\r\n";
}

int main(){
    string firstURL; cin >> firstURL;
    int n; cin >> n;

    CkSpider spider;
    spider.Initialize(firstURL.c_str());
    spider.AddUnspidered(firstURL.c_str());

    double exeTime = 0, k = 0;

    for (int i = 0; i < n+1; i++){
        auto initialExeTime = chrono::high_resolution_clock::now();
        bool found = spider.CrawlNext();
        auto finalExeTime = std::chrono::high_resolution_clock::now();

        if (found){
            pageDisplay(i, spider.lastUrl(), spider.lastHtmlTitle());
            exeTime += chrono::duration_cast<std::chrono::microseconds>(finalExeTime - initialExeTime).count();
            k++;
        }
        else {
            if (!spider.get_NumUnspidered()){
                cout << "There're no more pages to be crawled" << endl;
                break;
            }
            else
                cout << spider.lastErrorText() << "\r\n" << endl;
        }

        spider.SleepMs(1000);
    }

    double avgExeTime = (exeTime*1e-6)/k;

    cout << "Successfully crawled links: " << k << endl;
    cout << "Total crawling execution time(s): " << exeTime*1e-6 << endl;
    cout << "Average crawling execution time(s): " << avgExeTime << endl;

    return 0;
}
```

Figure 1. Código fonte utilizado (não comentado).

```
Page 18
Title: UFMG - Universidade Federal de Minas Gerais - Eventos
URL: http://ufmg.br/comunicacao/eventos/quarta-17h-live-educacao-e-politica
atina

Page 19
Title: UFMG - Universidade Federal de Minas Gerais - Eventos
URL: http://ufmg.br/comunicacao/eventos/boaventura-flavia-biroli-haddad-e-p
-pandemia-eleioes-e-o-futuro-da-democracia

Page 20
Title: UFMG - Universidade Federal de Minas Gerais - Eventos
URL: http://ufmg.br/comunicacao/eventos

Successfully crawled links: 21
Total crawling execution time(s): 8.4789
Average crawling execution time(s): 0.403757
matheusmtta@matheus:~/Desktop/Information-Retrieval$
```

Figure 2. Exemplo de parte do output completo.