UMA SOLUÇÃO EM SOFTWARE LIVRE PARA BUSCA DE INFORMAÇÕES NA WEB

Gustavo Tagliassuchi

Espacio Digital Tecnologia da Informação L
gustavo@tagliassuchi.com.br

Stanley Loh

Universidade Luterana do Brasil (ULBRA)
Universidade Católica de Pelotas (UCPEL)
sloh@terra.com.br

Resumo

Este artigo apresenta um sistema de busca de informações na Web construído com tecnologia de software livre. O sistema de busca é uma meta ferramenta, utilizando o Google como recurso para encontrar páginas Web conforme palavras-chave fornecidas pelo usuário. Depois disto, o sistema gera resumos dos textos presentes nas páginas resultantes. Os resumos são guiados pelos usuários, que devem definir novas palavras-chave. O uso de resumos permite ao usuário ter uma visão detalhada das informações presentes nas páginas Web sem necessidade de acessar diversas páginas seguindo os *links* resultantes. O artigo discute as tecnologias empregadas na implementação do sistema.

Abstract

This paper presents a system to search information in Web. The system was constructed with Free Software technologies. The system acts as a meta-tool, using Google to find Web pages according to keywords given as input by Web users. After that, the system generates summaries from the texts present in the resulting pages. Summaries are created when the user chooses new keywords. The summarization technique allows a deep insight of Web pages without the user needing to access many pages. The paper discusses the technologies used in the implementation of the system.

Introdução

Com a massificação do uso da *Internet*, também cresceu o volume de informações disponíveis. Para encontrar mais facilmente as informações desejadas em meio a tamanho volume, foram criados os mecanismos de busca (*search engines*). A maioria destes mecanismos funciona de forma livre, ou seja, oferecendo os serviços sem cobrança de pagamento, o que permite que todas as pessoas com acesso à Web possam fazer uso destes serviços.

Entretanto, sempre que é feita uma pesquisa, os mecanismos retornam também um volume grande de resultados. Muitos destes não são relevantes para os interesses sendo pesquisados. O problema é que o usuário só pode realmente verificar tal irrelevância à medida que segue o link indicado pela resposta e acessa o documento apontado.

Alguns mecanismos provêm junto ao link uma espécie de resumo, chamados "*snippets*", contendo algumas partes do documento apontado, para que o usuário possa ter noção do conteúdo deste documento. Porém, os *snippets* não constituem frases completas e geralmente são gerados sem um critério muito consistente.

Este trabalho busca melhorar a qualidade das informações fornecidas pelos mecanismos de busca na Internet, oferecendo ao usuário a oportunidade de criar resumos dos documentos apontados pelos *links* fornecidos como resposta por tais mecanismos. Os resumos podem ser criados a partir de critérios definidos pelo próprio usuário. Além de permitir verificar a relevância ou não do documento resposta, o resumo permite ao usuário encontrar informações específicas sem que precise acessar o tal documento.

A idéia deste trabalho é fazer uso destes serviços, classes e códigos específicos, gratuitos e de código aberto, e também oferecer um serviço gratuito, baseado em tecnologia de software livre.

Solução Proposta

Este trabalho desenvolveu um mecanismo de geração de resumos de páginas Web, chamado de ***Google Summarizer***. A finalidade é apresentar ao usuário Web partes dos documentos encontrados como resultado de determinado mecanismo de busca (como estudo de caso, foi utilizado o mecanismo Google). O mecanismo de resumo extrai frases completas, de acordo com critérios definidos pelo usuário. Esses resumos serão criados em tempo real e confrontados pelo usuário, ao mesmo tempo em que os resultados do mecanismo de busca vão sendo disponibilizados.

Além disto, o *Google Summarizer* permite ao usuário efetuar novas buscas sobre os resumos armazenados, tornando o processo recursivo.

A extração de resumos utiliza o método "guiado pelo usuário" (*user-driven*), ou seja, o próprio usuário define novas palavras para extração dos resumos. Assim, o sistema seleciona as frases onde as palavras fornecidos pelo usuário estão contidas e exibe ao usuário uma compilação desta frases. Os termos utilizados para pesquisa aparecem em destaque no resumo.

A principal vantagem do *Google Summarizer* é desonerar o usuário de ter que seguir os *links* resultantes de um mecanismo de busca tradicional para então ler os textos presentes na página Web apontada e aí sim verificar se a informação desejada encontra-se nesta página ou não. Além disto, o usuário terá que seguir todos os *links* resultantes, sendo que muitos deles não apresentam a informação desejada ou mesmo apontam para páginas não mais existentes.

Outra vantagem do *Google Summarizer* é que o sistema permite a busca de informações mais específicas, através dos resumos. Por exemplo, se uma pessoa deseja saber o endereço de um restaurante, primeiro utiliza o *Google Summarizer* para encontrar páginas sobre o restaurante desejado e então utiliza o mesmo mecanismo para extrair resumos das novas páginas resultantes, contendo textos onde pode estar a informação desejada (frases contendo as palavras "endereço" ou "rua" ou "avenida").

Tecnologia Utilizada

Para a criação do *Google Summarizer*, foi desenvolvida uma aplicação em linguagem *PHP* [3] que importa os dados dos mecanismos selecionados (através de *templates* específicos), e posteriormente, após a criação dos resumos, faz o armazenamento em um banco de dados *MySQL* [5].

Para a utilização do mecanismo *Google* e acesso direto ao seu banco de dados (através de sua *Web API – Application Program Interface* [4]), foi necessário ainda se incluir rotina para criação de objeto *SOAP* no aplicativo. O *SOAP* é um paradigma cliente-servidor, construído sobre tecnologias de *Internet*, para simplificar tarefas envolvendo procedimentos e acessando objetos através de uma rede [2]. Ele utiliza o *XML* para codificar procedimentos de solicitações (e decodificar suas respostas) num pacote ideal para transmissões através de redes, via http.

A *Google Web API* nada mais é do que uma licença gratuita para se utilizar a base de dados do mecanismo, desde que seja para uso pessoal, onde são disponibilizadas ferramentas e metodologias para troca de dados entre um software e a *API*.

Essa *API* foi disponibilizada pelo mecanismo Google como forma de manter seu desenvolvimento e comprometimento com seus usuários, e tornando sua base disponível no futuro a este tipo de utilização, massificar seu uso e se manter como referência em inovação quando se fala em busca de informações na Internet.

Para utilização da *Web API* do Google foi necessário implementar a sua interface *SOAP (Simple Object Access Protocol)* para na linguagem *PHP*, pois o *Google* só disponibilizou versões para *Java* e para a arquitetura *.NET*. O *SOAP* nada mais é que um protocolo "leve" para troca de informações, em ambientes descentralizados e distribuídos. Ele é baseado no protocolo *XML*, que define o seu *framework*, ou a forma como as mensagens serão trocadas e suas informações transportadas na utilização do protocolo. Em síntese, o *SOAP* é um protocolo baseado em *XML* que permite aos aplicativos trocarem informações através do protocolo *http*.

Ao final de uma conexão, o servidor *SOAP* recebe a solicitação SOAP contendo as chamadas dos procedimentos, as decodifica, executa suas funções, encapsula a resposta e envia o pacote *SOAP* de volta ao cliente que fez a requisição inicial. O cliente decodifica a resposta e utiliza o retorno da

maneira mais conveniente. O processo todo é de certa forma simples, pois é totalmente baseado em padrões já existentes e solidificados, o que o torna facilmente utilizável e compreensível.

A utilização do *SOAP* e da *API* do *Google*, permitiu a criação dos blocos *XML* de troca de informações. Porém além da criação de uma interface em *PHP* para este fim, conforme citado anteriormente, foi necessário o cadastramento no site do mecanismo a fim de se obter o registro e a "*license key*". Essa chave é utilizada para validar a permissão da utilização da base de dados do mecanismo.

Ainda, sua utilização é restrita, permitida somente para ambientes domésticos, e tendo limitações técnicas como: possibilidade de se executar apenas 1000 buscas/dia na base do mecanismo, somente retorna 10 resultados de cada busca, não permite codificar os textos de entrada e saída conforme o idioma utilizado, entre outras. A limitação dos 10 resultados foi contornada.

Por outro lado, a solução é maleável, aceitando as chaves de busca avançadas utilizadas no mecanismo *Google* normalmente, permitindo selecionar a busca por idioma, filtrar resultados para não duplicar as referências e ainda utilizar filtros para conteúdo adulto.

Os dados retornados pelo Google, em formato *XML*, recebidos em um *array* específico, são armazenados em um banco de dados *MySQL* de tabela única, com a seguinte estrutura:

```
cod,int(11),,PRI,NULL,auto_increment
query,varchar(200),,,,
title,varchar(200),YES,,NULL,
URL,varchar(200),,MUL,,
snippet,text,YES,MUL,NULL,
cachedSize,varchar(20),,,0,
results,text,YES,,NULL,
sec,int(11),,,0,
total,int(11),,,0,
resultsfull,text,YES,,NULL,
```

Onde pela ordem são armazenados os códigos de retorno (índice), a palavra-chave em questão, o título da página armazenada no mecanismo, a *URL*, o texto disponível, o tamanho da página, dois campos de controle, o que armazena o resumo e o que armazena a quantidade de palavras encontradas e o último campo, "resultsfull" que armazena todo o conteúdo (já limpo, sem *tags*) da página pesquisada.

Para uma melhor visualização se faz necessário a limpeza do código das páginas encontradas. São removidas *tags* desnecessárias de linguagens como *HTML, PHP, ASP, Javascript, CSS* entre outras, de forma que apenas o texto resultante seja armazenado e contabilizado para a criação do resumo. Para este trabalho se utilizaram algoritmos para limpeza de códigos, comandos do próprio PHP e ainda alguns otimizados pelo autor, pois novas *tags* são criadas com alguma freqüência, o que implica em novas mudanças nas funções de limpeza de código. Poderia ainda se observar a remoção de todas as *tags* e sinais, armazenando apenas texto puro, mas em alguns casos não é o ideal. A função de *fetching* [1] dos *links* fornecidos pelo *Google* foi desenvolvida como base de uma classe em PHP que simula o comportamento de um *web browser*, recuperando o conteúdo da página, e foi adaptada para este fim.

O sistema encontra-se disponível para uso nos endereços Web http://www.fiapo.com.br/tcc/ e também em http://www.edw3.net/tcc/

Experimento

Um exemplo de busca foi executado para demonstrar as potencialidades do *Google Summarizer*. A partir de um teste prático executado entre diversos participantes de várias instituições acadêmicas em todo o mundo [6], foram fornecidas 10 perguntas para que pessoas no mundo todo procurassem as respostas na Web, utilizando técnicas, recursos e estratégias diferentes. O objetivo era observar o tempo de busca além da qualidade das respostas e das estratégias utilizadas. Para o teste prático do sistema *Google Summarizer* foi selecionada uma pergunta, em inglês originalmente: *I need a map showing the location of the Penfold's winery in Australia.* (Eu preciso um mapa exibindo a localização da vinícola *Penfold* na Austrália).

Executou-se a busca pela palavra chave "*Penfold's winery australia*" e foi extraído o primeiro resumo com a palavra-chave "map". O resultado foi encontrado facilmente graças à habilidade da criação de resumos. Para exemplo de comparação, o teste efetuado na cidade de *Iowa* levou 8 horas para ser concluído, após 10 buscas distintas.

Conclusões

Este trabalho demonstra que várias aplicações podem beneficiar-se das tecnologias de software livre, como as empregadas neste sistema. Apesar dos benefícios do sistema desenvolvido, há ainda algumas limitações nos módulos oferecidos como "*free*" por alguns fornecedores, os quais podem inviabilizar seu uso.

Fica como trabalho futuro, a idéia de aprofundar os estudos e oferecer uma ferramenta mais robusta e para ser utilizada por um número maior de usuários, em diferentes ambientes, que não apenas o de testes e o acadêmico. Existe ainda a possibilidade de se oferecer esta tecnologia a empresas estabelecidas como forma de melhorar suas tecnologias.

Referências Bibliográficas

[1] Zmievski, Andrei. Web Client Class for PHP - http://snoopy.sourceforge.net/
[2] Ayala, Dietrich. NuSoap References - NuSoap Documentation. - http://dietrich.ganx4.com/nusoap/
[3] Apache Software Foundation. PHP Documentation - http://www.php.net/docs.php
[4] Google Services. Google Web API Service - http://www.google.com/apis/
[5] MySQL AB. MySQL Documentation - http://www.mysql.com/documentation/
[6] International ACM-SIGIR Conference on Research and Development in Information Retrieval. **Proceedings...** 1998.