Proposta de um Framework para Visualização de Dados Agregados por Similaridade para Auxiliar Consultas durante a Navegação na Web

Caio Stein d'Agostini, Ricardo Cava, Carina Dorneles, Sergio Firmenich, Carla Freitas, Philippe Palanque, Marco Winckler

Proposta de um Framework para Visualização de Dados Agregados por Similaridade para Auxiliar Consultas durante a Navegação na Web

Caio D'Agostini[1], Ricardo Cava[2], Carina F. Dorneles[3], Sergio Firmenich*, Carla M.D.S. Freitas[2], Philippe Palanque[1], Marco Winckler[1]

[1]IRIT-Université Paul Sabatier, 118 Route de Narbonne, Toulouse cedex 9, França {caio.stein-dagostini, palanque, winckler}@irit.fr

[2]INF-Universidade Federal do Rio Grande do Sul, C.P. 15064 91.501-970 Porto Alegre, RS Brasil {racava, carla}@inf.ufrgs.br

[3]INE/CTC-Universidade Federal de Santa Catarina, C.P. 476 88049-900 – Florianópolis, SC Brasil dorneles@inf.ufsc.br

ABSTRACT

In the last decade, several specialized tools have been created upon similarity functions that, given a keyword and a context, determine the degree of similarity (or probability) that information in a dataset corresponds to the user's query. Quite often such tools are meant for experts and require training and knowledge on the application domain to be used. However, given the huge amount of information available on the Web, resolving ambiguities becomes a daily task for most users. In this paper, we present a technique for embedding into a Web browser tools for solving ambiguities between keywords that users might found while navigating the Web. A prototype illustrating such techniques has been developed as a proof of concept. The tool presents the degree of similarity directly on Web pages as a contextual help menu. The overall approach includes different datasets and similarity functions and is flexible enough to support extensions for covering additional contexts of use.

Author Keywords

Web navigation, similarity functions, contextual help.

1 INTRODUÇÃO

A busca de informações na Web, a identificação de quais fontes de dados são confiáveis e a comparação de informações originadas a partir de fontes de dados diferentes tornaram-se tarefas do cotidiano, para as quais não se tem apoio suficiente. Um dos grandes problemas criados pelo volume e pela variedade de dados disponíveis na Web é o gerenciamento do processo de deduplicação de dados [14] (i.e. dados que possuem múltiplas representações do mesmo objeto do mundo real), por exemplo, "Antonio Carlos de Azevedo" pode ocorrer como "Antonio C. Azevedo", "Antonio de Azevedo" ou, mesmo, "Antonio Azevedo". Outro exemplo corrente é o uso aleatório da acentuação, ora incluindo acentos, ora omitindo-os, sem contar que, em algumas bases de dados, as diferenças de acentos não são tratadas.

Neste contexto, ferramentas especializadas criadas a partir de funções de similaridade podem ser utilizadas para efetuar casamento aproximado de dados, indicando quais instâncias correspondem a diferentes representações do mesmo objeto do mundo real [5]. Funções de similaridade são utilizadas em motores de busca, tais como Google[1] e Lucene[2]. Porém, o grau de similaridade entre documentos e a fonte dos dados usados na busca nem sempre é mostrada ao usuário, apenas o *ranking* dos resultados. Com frequência, apenas o ordenamento não é suficiente para resolver a ambiguidade e o usuário deve continuar navegando ou realizar buscas complementares.

Existem várias ferramentas especializadas na análise da similaridade de documentos [15]. Um dos problemas que surgem em ambientes com grandes volumes de dados é como visualizar os resultados dessas funções e as alternativas que devem ser apresentadas aos usuários para uma decisão. Dado o grande volume de informações pesquisadas, a quantidade de dados apresentada é muitas

*LIFIA, Universidad Nacional de La Plata/Conicet, Argentina; e-mail: sergio.firmenich@lifia.info.unlp.edu.ar

[1] http://google.com
[2] http://lucene.apache.org

vezes excessiva, deixando muitas vezes resultados relevantes (os chamados falsos-negativos) fora do campo de visão do usuário.

As ferramentas de análise de similaridade entre dados são dedicadas a um público especializado que conhece o domínio da aplicação, a forma como as funções são implementadas e foi devidamente treinado para interpretar os resultados. Contudo, dada a grande quantidade de informações disponíveis na Web, a busca por informações é frequente para a grande maioria dos usuários Web, os quais devem realizar pesquisas complementares em motores de busca para entender (ou desambiguar) palavras encontradas durante a visita a um site Web.

Um grande número de funções de similaridade pode ser utilizado para resolver varios tipos de ambiguidade entre documentos de acordo com as necessidades do usuário. Em um exemplo simples, considerando o nome de uma pessoa encontrada ao acaso em uma página Web, funções de similaridade poderiam ajudar a responder às seguintes perguntas do usuário: *Existem nomes similares a este que eu encontrei? Este nome tem valores similares em uma base de dados nas quais eu confio (ex. Lattes, DBLP)? Este nome tem variantes em bases de dados diferentes? É possível resolver a ambiguidade de um termo apenas verificando se ele ocorre em bases de dados confiáveis?*

Para poder tornar o acesso a funções de similaridade capazes de responder a estas (e outras) perguntas dos usuários que navegam na Web, este artigo apresenta um framework que foi construído a partir da análise de tarefas de usuários Web. A proposta deste framework repousa na seguinte premissa: "a *visualização de resultados mais complexos de uma função de similaridade pode auxiliar usuários a contextualizar os resultados de busca e é possível mostrar tais resultados em qualquer página Web"*.

De modo a descrever como este framework contribui para a area de Interação Humano-Computador, o artigo está organizado da forma a seguir. A Seção 2 contem os conceitos básicos e uma breve revisão do estado da arte das áreas relacionadas ao trabalho. A Seção 3 descreve o framework propriamente dito, enquanto a Seção 4 descreve as ferramentas que foram implementadas para apoiar a abordagem. A Seção 5 ilustra o uso de ferramentas aplicadas em uma série de estudos de caso. Finalmente, a Seção 6 discute os resultados, as conclusões e aponta os trabalhos futuros.

2 ESTADO DA ARTE

A contribuição do presente trabalho cobre várias disciplinas incluindo a análise de similaridade de dados [3], visualização de informações [21] e *design* de ajuda contextual em interfaces Web [8][9]. Nesta seção, apresentamos brevemente estes temas.

Análise de similaridade

Diferentes representações de um mesmo dado podem ser encontradas em vários domínios da Web. Em um cenário bem familiar à comunidade científica, nas bases de dados bibliográficas (e.g. DBLP, Lattes, CiteSeer, BDBComp, etc.), alguns dados como nomes de autores, por exemplo, podem aparecer com duplicações devido às diferentes formas de citação em diferentes artigos. Como não é possível assegurar com exatidão quais representações se referem ao mesmo objeto, a solução é prover um mecanismo que forneça uma medida da proximidade entre os valores. Isto é feito através da aplicação de métricas de similaridade que usam uma função $f(a1; a2) \rightarrow s$, que calcula um escore s para um par de valores $a1$ e $a2$ [5]. Quanto mais alto o valor do escore, mais similares os dois valores $a1$ e $a2$ são entre si.

Os problemas associados ao casamento de dados oriundos de fontes heterogêneas são amplamente referenciados na literatura [3, 5, 19]. Para resolver alguns destes problemas, muitos trabalhos propõem o uso de técnicas de Inteligência Artificial, tais como árvores de decisão e *Support Vector Machine* (SVM). Outra característica dos trabalhos que abordam o problema é efetuar o processo através do casamento de uma consulta representada por palavras-chave com uma coleção de valores de um conjunto, ligando cada objeto do conjunto a um escore de similaridade calculado em relação à consulta. Como resultado, é gerada uma lista ordenada dos valores da coleção de acordo com sua similaridade em relação à consulta. Por não ser uma resposta exata, cabe ao usuário decidir quais são, de fato, os objetos relevantes à sua consulta. Assim, de modo geral, é possível criar um grande número de funções de simularidade que atendem a diferentes critérios de consulta dos usários.

O uso de funções de similaridade para tratamento de dados armazenados em diferentes fontes é útil em diversos processos de gerenciamento de dados, tais como integração de dados, *data cleaning*, consulta aproximada, além de aplicações bastante variadas como, por exemplo, a análise de plágio em código-fonte [6]. De modo geral, todas estas ferramentas são reservadas a um público especializado.

Técnicas e ferramentas de visualização

Um dos problemas que surge em ambientes que fazem uso de funções de similaridade é como visualizar os resultados dessas funções e as alternativas que devem ser apresentadas aos usuários para decisão. Várias ferramentas tais como searchCrystal[3] [18], ilustrada na Figura 1, utilizam técnicas de visualização de informações para apresentar o grau de similaridade entre os resultados encontrados.

Tais técnicas são quase sempre interativas, pois objetivam, em sua grande maioria, apoiar usuários em processos de

[3] http://comminfo.rutgers.edu/~aspoerri/searchCrystal/

análise de dados [13]. São fornecidas, portanto, opções de manipulação e interação com o conjunto de dados através dessas representações visuais.

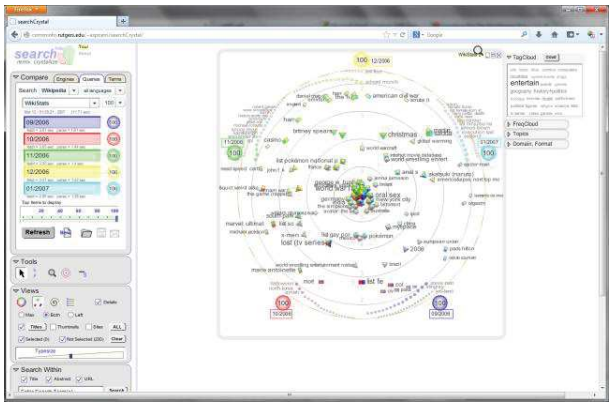

Figura 1. Visualização de resultados de busca com searchCrystal.

Algumas técnicas [2,11,12] exibem glifos ao lado de trechos dos documentos obtidos na execução de uma busca com o objetivo de identificar a sua relevância. TileBars [11] permite ao usuário perceber o tamanho relativo dos documentos recuperados, bem como a localização e a proximidade relativa dos termos da consulta. HotMap [12] utiliza uma escala de cores para representar a frequência de cada termo encontrado no respectivo documento. Chau [2] propôs um glifo, no formato de flor, para representar vários atributos de documentos web obtidos na busca. Cada conjunto de pétalas, exibido em uma cor diferente, representa um termo de pesquisa. A quantidade de elementos do conjunto identifica a frequência, o caule, o tamanho do documento e as folhas, a quantidade de links.

VisGet [4] combina a formulação de consultas interativas com um resumo visual dos resultados de busca ao longo de três dimensões: tempo, localização e tópicos. Na dimensão tempo, um gráfico de barras interativo indica a distribuição temporal dos itens permitindo a formulação de consultas temporais. A distribuição espacial dos resultados da busca é representada por círculos posicionados sobre um mapa. A composição dos tópicos do espaço de informações é sumarizada como uma *tag cloud* interativa onde o tamanho da fonte de texto utilizada indica a quantidade relativa dos itens de informação.

Algumas ferramentas permitem a exploração dos dados a partir da visualização do mapeamento resultante da aplicação de técnicas de redução de dimensionalidade. PEx-Web [16] realiza uma projeção multidimensional baseada na frequência dos termos e cria uma visualização onde cada documento, representado por um círculo, é posicionado de acordo com a similaridade do seu conteúdo em relação aos demais. PEx-Image [7], de forma similar, aplica técnicas de redução de dimensionalidade em um conjunto de imagens e realiza um posicionamento que enfatiza a semelhança do seu conteúdo. Já ProjCloud [17] mapeia uma coleção de documentos em um espaço visual, permitindo a visualização simultânea da similaridade entre os documentos e seu conteúdo por meio de *tag clouds*. Ela gera as *tag clouds* dentro de polígonos que contém documentos semelhantes ao mesmo tempo em que preserva o relacionamento semântico entre as palavras.

Dada a natureza dos dados a serem visualizados, técnicas de visualização de grafos [20] e as de visualização de dados multidimensionais [21] são particularmente apropriadas para a representação das informações de similaridade.

De fato, graus de similaridade podem ser facilmente representados como distâncias entre os nodos de um grafo. Técnicas de visualização multidimensional permitem mostrar graus de similaridade diferentes de acordo com várias bases de dados. Contudo, tais ferramentas de visualização apresentam dois inconvenientes conhecidos: i) pressupõem um usuário treinado e capaz de entender os grafos apresentados; e ii) são ferramentas isoladas que não se integram à tarefa de navegação do usuário na Web.

Análise de tarefas de busca e navegação na Web

A navegação na Web é uma tarefa que envolve a exploração de um espaço de informação no qual o usuário apreende a informação enquanto segue as conexões entre documentos [10]. Existe um paradoxo nesta tarefa que consiste em chegar a uma informação a partir da navegação no espaço de informação. Ferramentas de buscas criam artificialmente *links* entre páginas, pois dando uma resposta imediata à uma pergunta explícita do usuário, criam a impressão de que o usuário está navegando entre duas páginas (a que contem o formulário de busca e a página resultado). De fato tal interação não corresponde a uma navegação pelo espaço de informação, onde o usuário só poderia usar os caminhos originais para navegação entre as páginas. Baeza-Yates e Neto [1] fazem uma distinção clara entre essas diferentes tarefas que um usuário pode desempenhar: (i) busca de informações ou dados e (ii) navegação através dos links.

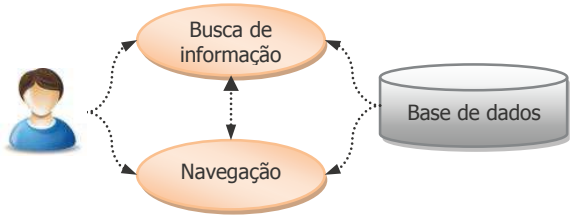

Figura 2. Modelo de interação do usuário e sistema de informação, adaptado de Baeza-Yates & Neto (1999) [1].

Como mostra a Figura 2, interfaces Web são uma tentativa de combinar estas tarefas e estender as capacidades de recuperação de informação. Em todo o caso, a combinação de recuperação de informação e navegação ainda não está bem estabelecida. Normalmente, o usuário não pode escolher de maneira transparente entre duas interações:

navegar pelos links que estão em uma página ou fazer uma busca a partir do *label* (e/ou outros atributos) do link.

Ferramentas de busca tais como Google e Quintura[4] fazem claramente uso de funções de similaridade para determinar o ordenamento de resultados. No caso de Quintura, como mostra a Figura 3, além do ordenamento dos resultados, a ferramenta implementa uma *tag cloud*, indicando conceitos próximos que auxiliam o usuário a refinar a busca e desambiguar os resultados.Enquanto ferramentas de busca são capazes de isolar um item de informação, a navegação permite criar conexões e expandir o conhecimento associado a um item de informação. Neste caso, existe o risco do usuário perder o foco de atenção durante a navegação em páginas conexas. Por esta razão, navegadores tais como Google Chrome implementam a busca contextual por palavra diretamente a partir de qualquer página Web.

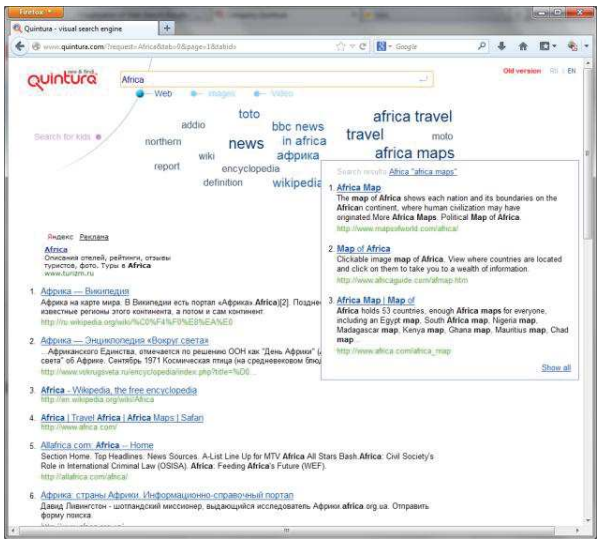

Figura 3. Quintura: resultados em lista e com *tag clouds*.

De todo modo, nenhuma ferramenta de busca indica claramente ao usuário quais fontes de dados foram utilizadas na busca e quais foram os escores encontrados. Desta maneira, embora os primeiros resultados na lista ordenada possam intuitivamente parecer os mais relevantes, eles escondem certas informações que induzem o usuário a navegar várias páginas para tomar uma decisão.

Assim, o usuário deve se desconectar-se temporariamente do contexto da tarefa principal (por exemplo, ler um artigo na Web) para resolver a ambiguidade. Ao final, para resolver um problema de ambiguidade em um texto, o usuário precisa realizar outra navegação, uma tarefa adicional, que o afasta do seu foco de interesse primário.

3 APRESENTAÇÃO DO FRAMEWORK

O objetivo principal do *framework* apresentado neste artigo é fornecer mecanismos para disponibilizar a usuários funções de busca por similaridade a partir de uma página Web qualquer, permitindo assim que os resultados (em termos de graus de similaridade) sejam visualizados nesta mesma página durante a navegação na Web. Um objetivo secundário, mas não menos importante, é permitir que o usuário final possa escolher a função de similaridade que melhor corresponde ao contexto da busca para resolução da possível ambiguidade entre palavras-chave. Convém ressaltar que o *framework* não trata *query disambiguation,* cujo objetivo é inferir o que o usuário deseja (por exemplo, se a busca é "São Paulo", o sistema tenta descobrir se é o Estado ou Time de Futebol).

Pode-se dizer que temos dois tipos de usuários diretos do *framework*: i) desenvolvedores de funções de similaridade que criam novos algoritmos capazes de tratar as questões de similaridade; e ii) usuários finais que se servem de tais funções durante a navegação na Web. O foco principal deste artigo são usuários finais. Assim, ainda que o *framework* proposto seja suficientemente flexível para acomodar novas funções de similaridade, este aspecto não será discutido aqui em detalhe.

De uma maneira simplificada, a arquitetura do framework é resumida na Figura 4 e descrita a seguir. O framework é distribuído entre o cliente e o servidor Web e inclui três grandes componentes: i) módulo cliente Web, *plug-in* cliente (Figura 4.A) que permite ao usuário interagir (indiretamente) com as funções de similaridade; ii) um repositório ou *broker servidor*, ou seja um intermediário de um serviço de funções de similaridade (Figura 4.B); e iii) as bases de dados que podem ser consultadas (Figura 4.C). O componente cliente (Figura 4.A) envia uma chamada de função a partir do navegador que se conecta a um servidor (Figura 4.B), que processa as funções de similaridade usando conexões com a base de dados (Figura 4.C).

Módulo cliente Web (*plug-in*) para similaridade

O módulo cliente Web (Figura 4.A) é a parte interativa do *framework*, e é o único componente visível para o usuário. Este módulo deve ser conectado ao navegador Web. Como será descrito na Seção 4, este módulo se materializa na forma de um *plug-in,* que deve ser instalado no navegador Web. As tarefas suportadas por este módulo incluem:

- Permitir ao usuário a seleção de palavras-chave a serem analisadas;
- Apresentar ao usuário a lista de funções de similaridade e invocar a função escolhida;
- Apresentar o grau de similaridade associado a cada palavra-chave selecionada pelo usuário.

Este módulo conhece as chamadas da função de similaridade, mas não armazena nenhum dado para processar as consultas. Todo o processamento ocorre do lado do servidor. Este módulo, porém, tem estreita ligação com o componente *broker* de similaridade, pois ele deve refletir, no lado do cliente, a lista de funções disponíveis.

[4] http://www.quintura.com/

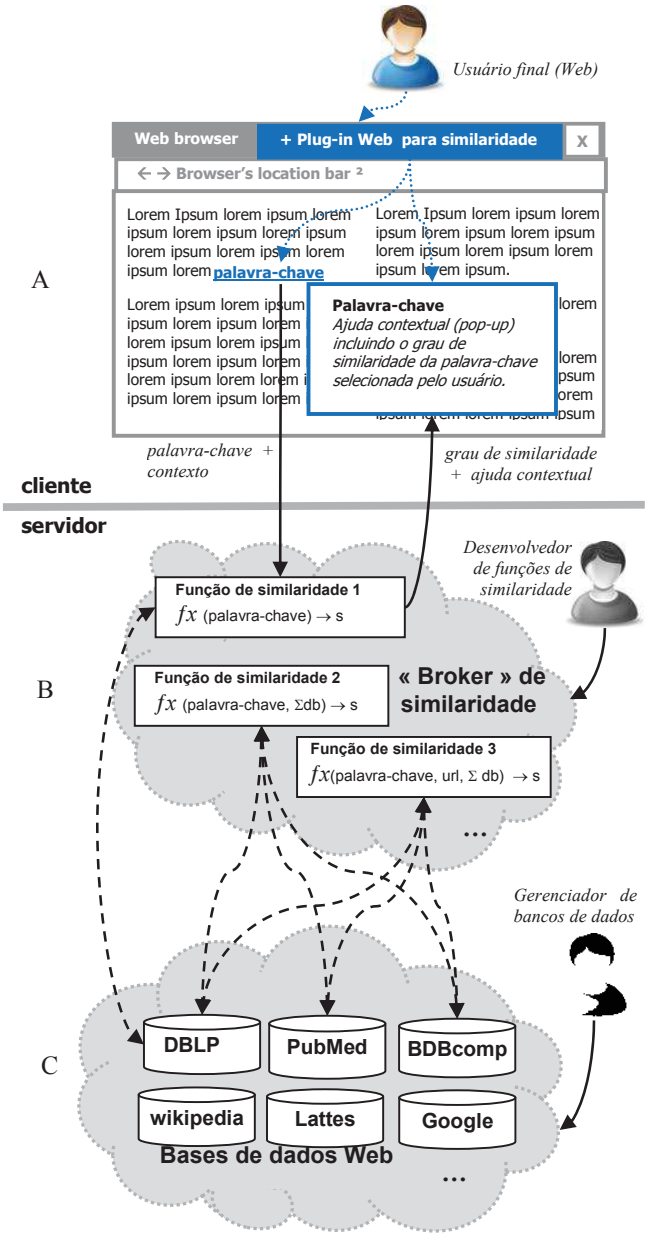

Figura 4. Arquitetura global do framework.

e acordo com a função de similaridade selecionada pelo suário, o módulo cliente deve transmitir como argumento s informações contextuais que podem ser usadas pela nção de similaridade para resolver a ambiguidade, como, or exemplo, o texto próximo da palavra-chave elecionada, a URL da página de onde o usuário lança a nsulta, o eventual link que pode estar associado à lavra-chave, etc. O resultado da função de similaridade é tornado ao cliente como um conjunto de valores, isto é, aus de similaridade, que são apresentados ao usuário por eio de uma ajuda contextual.

Módulo *broker* de similaridade

Este módulo (Figura 4.B) é a parte essencial do *framework* pois ele hospeda todas as funções de similaridade já desenvolvidas e acessíveis ao usuário através do *plug-in*. Contudo, não contém nenhuma interatividade direta com os usuários finais. Os únicos usuários que utilizam este módulo são programadores experientes que, não apenas devem conhecer os algoritmos de análise de similaridade de dados, mas também conhecem o acesso às bases de dados usadas na busca.

A arquitetura deste módulo e os detalhes da sua implementação não são descritos aqui pois estão fora do escopo deste artigo. A sincronização entre as funções disponíveis e o *plug-in* também não é descrita pois trata-se de detalhes técnicos de implementação, os quais são transparentes para o usuário final.

Entretanto, cabe enfatizar que este módulo pode conter um número grande de funções de similaridade. Tais funções podem levar em conta não apenas diferentes parâmetros de entrada fornecidos pelo usuário (tais como palavras-chave, URLs, palavras próximas, etc.) como também fornecer diferentes resultados, na forma de texto, tabelas ou grafos.

Bases de dados

Este elemento corresponde a qualquer conjunto de dados ao qual as palavras-chave do usuário são comparadas para estabelecer o grau de similaridade. Como ilustrado na Figura 4.C, tais bases de dados podem ser estruturadas (por ex., DBLP) ou semi-estruturadas (por ex., Wikipedia). A função de similaridade escolhida pelo usuário depende do tipo da base de dados utilizada na busca.

4 IMPLEMENTAÇÃO DA FERRAMENTA

Nesta seção é apresentada a ferramenta cliente que foi desenvolvida na forma de um *plug-in* para o navegador Firefox. O *plug-in* é a única parte do *framework* visível para o usuário final e será a única parte descrita neste artigo, a título de detalhamento.

A implementação do *plug-in* foi feita reutilizando componentes do *framework* CSN[5]. Este *framework* permite criar facilmente funções (chamadas de *augmenter),* as quais suportam adaptações na página Web que está sendo visualizada pelo usuário. Essas adaptações *aumentam* os elementos de interação da página Web, permitindo mudar a cor do texto da página e criar janelas *pop-up*, por exemplo, dando flexibilidade ao desenvolvedor e potencializando as funcionalidades das ferramentas.

O *plug-in* é disponibilizado na forma de um arquivo Javascript (.js), que contém um *augmenter* para o *framework* CSN. Tudo o que o usuário deve fazer para utilizá-lo é fazer *download* de um arquivo .xpi a partir do

[5] http://www.megadll.net/csn/

navegador Firefox que se encarregará de extrair os arquivos e instalar o *plug-in*.

Uma vez instalado, as funções de similaridade tornam-se acessíveis ao usuário final a partir de menu contextual do navegador, como mostra a **Figura 5**.

As tarefas suportadas pelo *plug-in* incluem:

- **Seleção de palavra(s)-chave**: A seleção de uma palavra chave é feita com funcionalidade de seleção (clicar e arrastar) do próprio *browser*. Quando esta seleção é realizada, o *plug-in* altera a página visitada (no *browser*) e adiciona marcações para que seja possível, mais adiante, na interação com o usuário, localizar a origem da consulta e onde mostrar os resultados.
- **Invocar uma função de similaridade**: O *augmenter* conhece um ou mais repositórios de funções de similaridades. Para cada função, uma entrada no menu da ferramenta é adicionada (**Figura 5**). O nome de cada uma destas novas entradas é uma breve descrição do funcionamento da função. Ao selecionar uma função, a mesma é invocada para a(s) palavra(s)-chave que o usuário tem atualmente selecionadas ("José Silva" na **Figura 5,** através da funcionalidade de selecionar texto do *browser*). Caso a função exija outros parâmetros além das palavras selecionadas, estes também são passados neste momento (por exemplo, a URL da página Web). O protótipo descrito neste trabalho apresenta somente quatro funções de similaridade, porém outras poderiam ser adicionadas, através de uma nova versão do *plug-in*, adaptada para invocar as novas funções com os parâmetros apropriados.
- **Apresentar os valores de similaridade**: Os valores retornados pela função de similaridade são apresentados em uma tabela (**Figura 6**). O título da tabela corresponde à(s) palavra(s)-chave para a(s) qual(is) o usuário deseja verificar a similaridade. Cada linha da tabela corresponde a um elemento com a qual a similaridade foi calculada, sendo a primeira coluna um identificador da fonte de dados, enquanto a segunda coluna indica o grau de similaridade que varia entre 0 (nenhuma similaridade) e 1 (similaridade máxima). O *plug-in* também indica ao usuário a que se refere à tabela de similaridade – apesar da tabela ter um título, este pode se referir, potencialmente, a diversos elementos da página (por exemplo, uma palavra que aparece diversas vezes em um texto).

Interação com a ferramenta

O modo de interação com a ferramenta é relativamente simples: o usuário deve selecionar uma palavra-chave no texto de uma página Web e invocar a função de similaridade a partir do menu contextual, como mostra a **Figura 5**. A invocação das funções é feita clicando com o botão direito e escolhendo a opção "Desambiguar" do menu. Em seguida, a lista de funções de similaridade disponíveis é exibida.

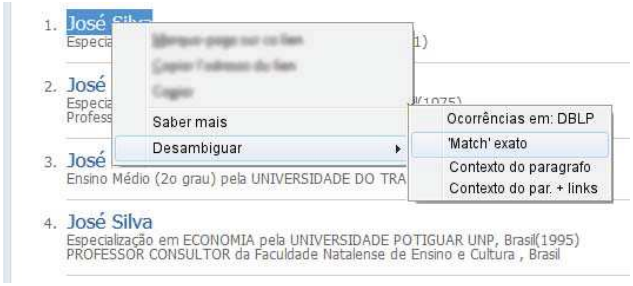

Figura 5. Menu contextual do navegador com funções de desambiguação disponíveis através do *plug-in*.

A seleção de um função da lista provoca a invocação da mesma, que retorna uma tabela de valores contendo os graus de similaridade encontrados. Como indica a **Figura 6**, esta tabela é mostrada próxima à palavra-chave. A visualização da tabela é ativada por um evento do tipo *"mouse-over"* e desaparece quando o usuário retira a seleção da palavra-chave.

Figura 6. Exemplo de ajuda contextual mostrando graus de similaridade obtidos a partir de funções de similaridade disponíveis no *plug-in*.

As Figuras **Figura 5** e **Figura 6** ilustram a interação com o *plug-in* a partir de uma página de consulta na plataforma de currículos Lattes do CNPq[6]. Contudo, cabe ressaltar que as funções de similaridade são disponíveis a partir de qualquer página Web.

O tempo necessário para obter os resultados depende de vários fatores, tais como, a complexidade da função de similaridade, o tamanho da base de dados consultada e a sobrecarga do servidor.

5 ESTUDO DE CASO

Para ilustrar o uso do *framework* e das ferramentas desenvolvidas, esta seção descreve um estudo de caso que contém um cenário que envolve vários casos de busca usando funções de similaridade distintas, com o objetivo de desambiguar uma palavra-chave em diferentes contextos.

[6] http://lattes.cnpq.br/

Para tanto, considere um usuário que necessita encontrar informações sobre um indivíduo. No nosso exemplo, o usuário trabalha com bioinformática aplicada à medicina e encontra, por acaso, um artigo *online*, como o ilustrado na **Figura 7**, o qual menciona o trabalho de um outro pesquisador chamado "Jose Silva", sobre o qual ele deseja mais informações.

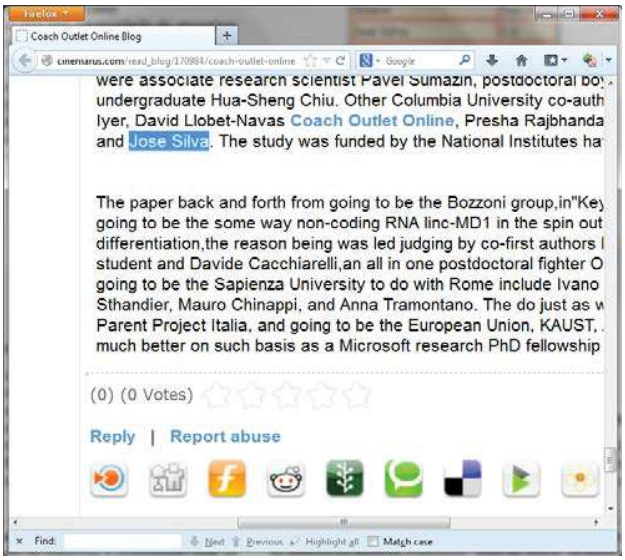

Figura 7. Página Web inicial [7] a partir da qual o usuário decide usar funções de similaridade para tomar uma decisão.

Entretanto, o nome procurado "Jose Silva" é muito comum, motivo pelo qual todas as buscas realizadas pelo usuário retornam muitos resultados, de diferentes contextos (por exemplo, professores, estudantes, médicos, cientistas da computação, escritores, etc.). Mesmo buscando em bases de dados de domínios mais específicos, os resultados ainda podem ser muito ambíguos, como mostra a **Figura 8**.

Para fins de ilustração, nosso usuário não busca apenas por qualquer "José Silva", mas sim um "José Silva" pesquisador que trabalha na área de bioinformática aplicada à medicina. Usando um motor de busca, o usuário poderia encontrar informações em:

- bases de informática por resultados que também tratem de medicina;
- resultados que também referenciem medicina em bases de informática;
- resultados que tenham referência à medicina e à informática no currículo de registros de bases como a plataforma Lattes.

[7] Em: http://cinemarus.com/read_blog/170984/coach-outlet-online

Figura 8. Resultados de uma busca por "Jose Silva" usando ferramentas disponíveis na plataforma Lattes.

Usando qualquer ferramenta de busca atual, o usuário terá que navegar por diversas abas e/ou janelas de navegadores e identificar a correspondência entre o conteúdo de cada navegação e a relevância do conteúdo para o tão procurado "Jose Silva".

Para mostrar como o *framework* pode auxiliar a busca do usuário, apresentamos, a seguir, vários cenários contendo os resultados de funções de similaridade diferentes. Estes diferentes cenários servem para mostrar que a função a ser utilizada depende do contexto do tipo de ambiguidade que o usuário está tentando resolver. Para tanto, vamos considerar quatro tipos de busca por similaridade:

- **Busca 1 = Sintático, nome com ocorrências de uma base de dados**. Esta função consulta somente uma fonte de dados e retorna o grau de similaridade da palavra buscada com possíveis *matchs* da base de dados.
- **Busca 2 = Sintática, nome com múltiplas bases de dados**. Precisa haver a mesma palavra chave; se for composta, ter a mesma ordem e grafia (como buscar por "José Silva", com aspas, em um buscador).
- **Busca 3 = Contextual**. Inclui todas as palavras no mesmo parágrafo. Esse conjunto de palavras é comparado com as palavras em cada uma das outras bases (como recuperar em um buscador por "José Silva Columbia bioinformatic", onde Columbia e bioinformatic são palavras do artigo que despertaram interesse do usuário).
- **Busca 4 = Contextual + *Links***. Funciona tal qual a Busca 2; porém, se há *links* no parágrafo, as palavras contidas no *link* também são enviadas como parâmetro

(como se, além de adicionar palavras-chave retiradas do texto (busca 2), o usuário também as buscasse em todos os *links* do artigo).

Execução de cenários de busca usando o *framework*

O restante desta seção descreve como o *framework* e as ferramentas desenvolvidas auxiliam o usuário a realizar as buscas previamente descritas.

Busca 1: Sintático, nome com ocorrências de uma base

Considere o mesmo problema descrito anteriormente, porém, agora, o usuário tem à sua disposição a ferramenta proposta neste artigo. Como o usuário quer procurar informações sobre outras produções científicas, ele busca por "José Silva" em uma base como DBLP ou BDBComp (informática), PubMed (medicina) ou Lattes (todas as áreas). O usuário, para ter uma ideia da dificuldade da busca (por exemplo, saber se basta 'José Silva' ou se precisa de mais informações), pode verificar a similaridade da palavra consultada com ocorrências em uma base de dados, como mostrado na **Figura 9**, que lista diversos possíveis arranjos do nome. Usando o *plug-in* aqui descrito, esta função pode ser invocada diretamente do artigo *online* (**Figura 7**) ou de qualquer conteúdo no formato HTML.

José Silva

Source	Similarity
José Silva	1.0
Francisco José Silva Silva	0.53
José Fernando Silva	0.52
José Luis Silvan-Cardenas	0.48
Paulo José Silva Silva	0.48
José C. Silva	0.41
José Silva Matos	0.40
Douglas José Silva	0.39
Elton José Silva	0.39
José Augusto Silva	0.39
José Luís Silva	0.39
José Machado Silva	0.39
José Reinaldo Silva	0.39
José Silvestre Silva	0.39
José A. Silva	0.38
José F. Silva	0.38
...	...

Figura 9. Resultados da ocorrência de nomes similares a "Jose Silva" na DBLP.

Como as possibilidades para "José Silva" são muito amplas, o usuário decide continuar a busca em uma base que auxilie a contextualizar os resultados. Neste caso, o usuário busca na plataforma Lattes, tal como ilustrado na **Figura 8**, a qual fornece informações sobre formação, área de atuação, etc.

Apesar dos resultados apresentarem uma classificação, não é clara a diferença entre alguns resultados (por exemplo, resultados entre 1 a 4). Em vez de abrir os *links* de cada resultado, um por um, o usuário utiliza a ferramenta para tentar esclarecer as diferenças entre os diferentes "José Silva", tal como mostra a **Figura 5**.

Busca 2 : Sintático, nome com múltiplas bases

Quando o usuário passa o cursor sobre o texto selecionado para desambiguação, uma pequena tabela apresenta que fontes de dados estão cadastradas na ferramenta (ver coluna 'Source' da **Figura 6**) e similaridade do conteúdo buscado ("José Silva") com o conteúdo da base ('Similarity'). Como "José Silva" é um nome comum, ele está presente em todas as bases. Como a função encontra *matchs* exatos em todas as bases, a tabela resultante não ajuda o usuário. Porém, caso o usuário buscasse um nome menos frequente (exemplo José Hyppolito Silva), a tabela resultante seria bem diferente já que a única base de dados com este nome é a PubMed – o que indicaria ao usuário que o resultado se refere, provavelmente, a um médico. Este exemplo é ilustrado pela **Figura 10**.

José Hyppolito da Silva

Source	Similarity
DBLP	0.0
PubMed	1.0
DBBComp	0.0

Figura 10. Função de similaridade do tipo 2 "sintático" aplicada a nome "José Hypolito da Silva".

Busca 3 : Contextual

Ao escolher a função 3 (**Figura 5**), o *browser* envia como parâmetro da função, além da palavra(s) selecionada(s), outras informações que podem ajudar a contextualizar a palavra-chave, por exemplo, todo o conteúdo próximo à palavra-chave (no caso do Lattes, o parâmetro seria o parágrafo contendo um pequeno resumo sobre colocação profissional e formação do indivíduo correspondente a cada resultado). A similaridade de um resultado com uma base de dados é, então, calculada, levando também em consideração a frequência com que essas palavras ocorrem na base. Os resultados obtidos são mostrados na **Figura 11**.

Conforme visível pelos valores, a informação contextual extra passada para a função permite ao usuário ver que:

- a primeira ocorrência (**Figura 11**.a) selecionada provavelmente se trata de alguém que trabalha com computação (valor alto para DBLP) e não em medicina (valor baixo para PubMed). Porém, ainda há dúvidas, já que o valor para BDBComp é reduzido, apesar de também se tratar de computação.
- A segunda ocorrência (**Figura 11**.b) provavelmente se trata de um médico, devido ao alto valor de PubMed
- A terceira ocorrência (**Figura 11**.c) tem um valor não muito elevado, porém aproximado, para todas as bases de dados. Isto pode indicar que se trata de alguém que trabalha com informática na medicina (que é exatamente o que o usuário procura)

José Silva

Source	Similarity
DBLP	0.75
PubMed	0.05
DBBComp	0.10

a)

José Silva

Source	Similarity
DBLP	0.05
PubMed	0.80
DBBComp	0.01

b)

José Silva

Source	Similarity
DBLP	0.6
PubMed	0.67
DBBComp	0.59

c)

Figura 11. Desambiguação de 3 ocorrências de 'José Silva' selecionadas em 3 parágrafos diferentes dentre os resultados da busca na plataforma Lattes.

É importante chamar a atenção que esses valores foram obtidos a partir do contexto obtido pela função extraindo informações do *site* Lattes. Se a busca tivesse sido originada em outro *site*, por exemplo, DBLP (**Figura 12**), os resultados seriam outros (afinal, a informação na página obviamente não seria mais a mesma), devido ao contexto diferente. Por exemplo, a quantidade de informação no parágrafo de um resultado ("Matos") é bem menor que no Lattes.

Figura 12. Tabela gerada com a função 3 para uma ocorrência.

Busca 4 : Contextual + Links

O resultado obtido usando esta busca é semelhante ao da busca 3, porém agregando ainda mais conteúdo no conjunto de parâmetros passados. No *site* do Lattes, por exemplo, esta função pode acessar o conteúdo do perfil/currículo de cada 'José Silva'. Logo, poderia utilizar informação detalhada sobre formação, publicações passadas, etc.

Novamente, o resultado indica (tal como a Função 3) que a terceira ocorrência (**Figura 11**.c e **Figura 13**.c) se refere a alguém cujo trabalho é pertinente tanto à computação quanto à medicina.

A escolha de um cenário com extremos (um nome bastante comum com um domínio consideravelmente específico (bioinformática na medicina) foi feita por ilustrar mais facilmente como relações de similaridade (de forma, conteúdo, etc.) entre diferentes bases de dados podem ser exploradas por usuários leigos (quanto a cálculo de similaridade, recuperação de informação, etc) desde que estas sejam apresentadas de uma maneira adequada.

José Silva

Source	Similarity
DBLP	0.57
PubMed	0.72
DBBComp	0.61

a)

José Silva

Source	Similarity
DBLP	0.12
PubMed	0.55
DBBComp	0.06

b)

José Silva

Source	Similarity
DBLP	0.70
PubMed	0.07
DBBComp	0.08

c)

Figura 13. Tabela gerada com função 3 para uma ocorrência.

6 COMENTÁRIOS FINAIS E TRABALHOS FUTUROS

O trabalho apresentado neste artigo integra conhecimento de várias áreas tais como análise de similaridade, tecnologia Web, visualização de informações, mas acima de tudo de projeto centrado no usuário. De fato, todas as técnicas empregadas neste trabalho são colocadas a serviço de usuários finais de modo a ajudar a resolver problemas de ambiguidades entre instâncias de dados encontrados durante a navegação na Web. Com volume crescente de informações na Web, novas técnicas baseadas em funções de similaridade podem tornar a busca por informações mais eficiente. Embora este trabalho trate os detalhes técnicos das funções de similaridade sem apresentá-los totalmente aos usuários, ele visa resolver dois problemas básicos dos usuários Web com ferramentas de buscas atuais:

- Apresentar explicitamente os graus de similaridade de buscas e as fontes de informações de uma consulta, que atualmente são escondidas atrás do ordenamento de resultados;
- Facilitar a visualização de resultados de busca por similaridade diretamente a partir do contexto da página, evitando uma navegação por múltiplas páginas.

Este trabalho é um primeiro passo em direção a prover transparência em ferramentas de buscas. Os resultados ainda são preliminares mas encorajantes. Os princípios de funcionamento do *framework* foram apresentados e devidamente ilustrados através de cenários típicos. Estudos experimentais com usuários são os próximos passos, de modo que sejam confirmadas as conclusões preliminares retiradas dos estudos de caso apresentados.

O *framework* introduz uma nova abordagem para algoritmos de busca e de visualização de resultados pois torna evidente os graus de similaridade. Como ilustrado neste artigo, existe um grande potencial para se criar funções de similaridade que sejam adaptadas a vários tipos de busca, permitindo a experimentação de uma maneira simples e direta de testar tais funções. O *framework* permite, ainda, investigar como usuários interpretam e se apropriam dos resultados de busca. Embora se considere que os graus de similaridade brutos (variando de 1 a 0) sejam relativamente intuitivos, testes com usuários são necessários para determinar se, de fato, as pessoas são capazes de interpretar corretamente esses escores. Na mesma linha de estudo, é necessário investigar se os

usuários apreciam a busca contextual e até que ponto eles selecionam as funções de similaridade apropriadas para os diferentes critérios de busca possíveis.

Outros trabalhos futuros incluem prover formas alternativas de representação e de visualização de graus de similaridade e a investigação de técnicas que coletam interativamente a opinião do usuário sobre o grau de similaridade apresentado como resultado. Esta opinião, ou *feedback*, poderá, então, ser usada para refinar o cálculo usado em funções de similaridade de modo a produzir resultados ainda mais confiáveis.

AGRADECIMENTOS

À CAPES, financiadora do projeto CAPES/COFECUB 735/12 entre UFRGS, UFSC e IRIT.

REFERÊNCIAS

1. Baeza-Yates, R., Neto, B. R. Modern Information Retrieval. New York, ACM Press Books / Addison-Wesley (1999).
2. Chau, M. Visualizing Web search results using glyphs: Design and evaluation of a flower metaphor. *ACM Transactions on Management Information Systems*. 2(1). ACM (2011). Article 2.
3. Chen, F. R., Farahat, A. O., Brant, T. Multiple similarity measures and source-pair information in story link detection. In *HLT/NAACL* (2004), 313-320
4. Dork, M.; Carpendale, S.; Collins, C.; Williamson, C. VisGets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Trans. on Visualization and Computer Graphics*, 14(6). IEEE (2008), 1205–1212.
5. Dorneles, C. F., Gonçalves, R., dos Santos Mello, R. Approximate data instance matching: a survey. *Knowledge Information Systems,* 27(1): 1-21 (2011).
6. Đurić, Z., Gašević, D. A Source Code Similarity System for Plagiarism Detection. *Computer Journal*, 56(1) (2013), 70-86.
7. Eler, D. M.; Nakazaki, M.; Paulovich, F. V.; Santos, D. P.; Oliveira, M. C. F.; Batista Neto, J. E. S.; Minghim, R. Multidimensional visualization to support analysis of image collections. In *SIBGRAPI 2008*. IEEE (2008), 289-296.
8. Firmenich, S., Gaits, G., Gordillo, S., Rossi, G., Winckler, M. Supporting Users Tasks with Personal Information Management and Web Forms Augmentation. ICWE 2012: 268-282.
9. Firmenich, S., Winckler, M., Rossi, G. A Framework for Concern-Sensitive, Client-Side Adaptation. In *ICWE* Springer (2011), LNCS 6757,198-213.
10. Fleming, J. Web Navigation: Designing the User Experience. O'Reilly, 1998. 264 p.
11. Hearst, M. A. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *CHI*. ACM (1995).
12. Hoeber, O., Yang, D. X. A comparative user study of web search interfaces: HotMap, Concept Highlighter, and Google. In *IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE(2006).
13. Keim, D. A. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1). IEEE (2002), 1-8.
14. Navarro, G. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1). ACM(2001), 31-88.
15. Nichele, C. M., Becker, K. Clustering Web Sessions by Levels of Page Similarity. In *PAKDD*'2006. Springer (2006), 346-350.
16. Paulovich, F. V., Pinho, R., Botha, C. P., Heijs, A., Minghim, R. PEx-Web: Content-based visualization of web search results. In *IV'08*. IEEE (2008), 208-214.
17. Paulovich, F. V. , Telles, G. P., Toledo, F. M. B., Minghim, R., Nonato, L. G. Semantic Wordification of Document Collections. *Computer Graphics Forum*, 31(3). Eurographics (2012), 1145-1153.
18. Spoerri, A. Visual Mashup of Text and Media Search Results. In *IV'07*. IEEE (2007).
19. Tejada , S.; Knoblock, C.A., Minton, S. Learning object identification rules for information integration, *Information Systems*, 26(8). (2001), 607-633.
20. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J.J., Fekete, J.-D. and Fellner, D.W. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6), Eurographics (2011), 1719–1749.
21. Ward, M., Grinstein, G., Keim, D. *Interactive Data Visualization: Foundations, Techniques, and Applications*. Nantick, A K Peters/CRC Press (2010). 513 p.