Metodologia de busca, armazenamento e classificação de dados semi-estruturados na web

Methodology of search, storage and classification of semistructured data in the web

Metodología de investigación, almacenamiento y clasificación de datos semiestructurados en la Web

O objetivo deste projeto de pesquisa é o desenvolvimento de uma metodologia de armazenamento e busca em grandes massas de dados semi-estruturados, enfocando possíveis aprimoramentos, e aplicação na área de Text Mining, englobando artigos relacionados a diversas especialidades. O presente trabalho pretende contribuir na direção de fornecer uma metodologia robusta, baseada nos algoritmos de Text Mining, para melhorar a análise automatizada de documentos por meio da classificação de textos, e facilitar o armazenamento e a obtenção de informações relevantes a diversas especialidades contidas em documentos armazenados na Web.

This research project aims to develop a methodology to store and recover a great mass of semi-structured data, focusing possible improvements and applications in text mining, comprising related articles from many specialties. The present work intends to offer a strong methodology based on text mining algorithms, searching for a better automated analysis of documents by text classification and making it easy to store and acquire relevant information from documents on the Web.

El objetivo de este proyecto de investigación es el desarrollo de una metodología de almacenaje y búsqueda en grandes cantidades de datos semiestructurados, planteando posibles perfeccionamientos y aplicación en el área de Text Mining, englobando artículos relacionados a diversas especialidades. Este trabajo pretende contribuir para una metodología fuerte, basada en los algoritmos de Text Mining, con el objetivo de mejorar el análisis automatizado de documentos por medio de la clasificación de textos, y facilitar el almacenaje y la obtención de informaciones relevantes a las diversas especialidades contenidas en los documentos almacenados en la WEB.

Palavras-chave: mineração de textos na web, KNN (K-Vizinhos mais próximos), banco de dados.

Autores:

Prof. Carlos Augusto Sicsú Ayres do Nascimento

Mestrado em Engenharia de Computação pela Universidade do Estado do Rio de Janeiro, Brasil (2004), Doutorando em Engenharia (Sistemas Computacionais), Atuação em Sistemas de Computação de Alto Desempenho, Prof. Assistente da Universidade Estácio de Sá, Brasil.

E-mail: sicsu@globocom

Prof. José Luiz dos Anjos Rosa

Doutorado em Engenharia Civil (Sistemas Computacionais) pela Universidade Federal do Rio de Janeiro, Brasil (2003). Professor Titular da Universidade Estácio de Sá, Brasil.

E-mail: joseluiz.rosa@gmail.com

Prof. Nelson Francisco Favilla Ebecken

Orientador de Mestrado, Orientador de Doutorado Doutorado em Engenharia Civil pela Universidade Federal do Rio de Janeiro, Brasil (1977), Livre Docência pela Universidade Federal do Rio de Janeiro, Brasil (1990). Atuação em Estruturas, Professor Titular da Universidade Federal do Rio de Janeiro, Brasil.

E-mail: nelson@ntt.ufrj.br

Introdução

Existe uma enormidade de textos armazenados em formato eletrônico disponíveis atualmente. Esses textos estão armazenados tanto na Web, como em intranets dentro de empresas ou em

órgãos dentro das mais diferentes instituições. Com a disponibilização dessas publicações, é crescente a capacidade de se obter uma enorme quantidade de informação, mas isso não aumenta a capacidade das pessoas de ler e analisar essas informações.

Esses textos, armazenados em formato eletrônico, são muito comuns hoje em dia, e são publicados nos documentos a cada segundo. Com o crescimento contínuo no volume desses dados, novas formas de extração de conhecimento tornaram-se necessárias para buscar informações armazenadas em memorandos internos, notícias, artigos, mensagens de e-mail ou em páginas Web [1,2]. Com todo este material à disposição, a grande dificuldade está em obter documentos de interesse, uma vez que não é possível ler todos estes documentos, para, então, decidir quais são de real interesse.

Nesse sentido, a busca dessas informações é tarefa árdua e difícil, pois as formas de pesquisa tradicionais trazem como resultado uma grande quantidade de respostas. Por exemplo, uma pesquisa realizada no Google [3] pela ocorrência *text mining* em 20/02/2007, produziu, como resposta, um conjunto com mais de quarenta milhões de resultados. O problema passa a ser o de se obter uma quantidade menor de documentos interessantes e não apenas obter documentos, o que torna necessária a criação de uma lista ordenada para determinar e listar, automaticamente, o resultado a partir dos melhores resultados encontrados nas publicações [4,5].

Deve-se considerar, ainda, que o esforço computacional para a efetivação das diversas tarefas correlacionadas é muito grande. Este óbice é resolvido pela divisão das tarefas em módulos, de tal forma que a carga de trabalho seja distribuída entre diversos computadores.

O contexto de um documento é formado pelos relacionamentos existentes (quando, onde e por quem foi escrito, além dos dados de publicação). Assim, um documento passa a exigir um contexto, pois um assunto pode não ter relevância em função do tempo de publicação. Documentos escritos em alguns locais podem ser mais relevantes do que documentos escritos em outros. Desta forma, para algumas aplicações, pode ser muito importante que se analise o contexto junto com o documento.

De acordo com Tan [6], existe um potencial maior no uso de *Text Mining* do que no uso de *Data Mining*. "... 80% das informações de uma empresa estão armazenadas em documentos de texto".

Atualmente, a automatização de processos vem crescendo em todas as áreas, não somente nas relacionadas à Tecnologia da Informação, e já é tida como fenômeno natural, inevitável e desejável, ou ainda, símbolo de modernidade. É inegável que sua aplicação gera impactos diretos em nível social, fato que desperta a atenção de muitos pesquisadores. Essa atenção tem a sua importância e contribui para o entendimento da automatização [6 VIEIRA].

A internet é uma grande fonte de dados, disponibilizada facilmente a qualquer usuário, fornecendo os mais variados assuntos. Porém, os dados fornecidos são pesquisados por meio da comparação de palavras-chave, indexadas em potentes bancos de dados dos servidores nos quais estão os sites de busca. Esses dados não contemplam o valor intangível de uma busca, mas tão somente a ocorrência pura e simples de uma palavra específica no conteúdo dos arquivos a procurar.

A automatização de um processo consiste na utilização de tecnologia, objetivando simplificar, minimizar ou eliminar a intervenção humana sobre esse processo, ou seja, torná-lo automático. A tendência constatada nos dias de hoje pode ser justificada pela necessidade de se diminuir a quantidade de documentos a serem lidos, objetivando a obtenção de um número menor de ocorrências, mas que sejam de maior interesse.

Este projeto de pesquisa será capaz de automatizar a busca por documentos relevantes, utilizando-se das técnicas a seguir.

Text Mining

Text Mining (também conhecido como *Text Data Mining* ou descoberta de conhecimento em bases de dados textuais), refere-se ao processo de extração de processos interessantes e não triviais de padrões a partir de documentos de texto com dados não estruturados. Também é visto como uma extensão de *Data Mining*, extração de conhecimento em bases de dados estruturados [6]. Atualmente, com as tarefas de *Data Mining* consolidadas, muitos esforços estão sendo aplicados à área de *Text Mining*.

Diferentemente das técnicas de *Data Mining*, as técnicas de *Text Mining* visam a efetuar buscas em dados não-estruturados ou semi-estruturados, e ainda não estão consolidadas ou não são tão eficientes quanto às de *Data Mining*. *Text Mining*, entretanto, envolve tarefas muito mais complexas do que as tarefas de *Data Mining* [6]. Muitas soluções foram propostas para agrupamento, sumarização e categorização de documentos, com o intuito de facilitar as de informação.

Preparação dos dados

Case Folding

É um processo usado para acelerar as comparações a serem feitas no processo de preparação dos dados em *Text Mining*. Visa a converter todos os caracteres a uma mesma forma, todas maiúsculas ou todas minúsculas.

Stopwords

Um dos pontos importantes da preparação dos dados é a eliminação das palavras que não possuem representatividade sobre o texto. Essas palavras, normalmente utilizadas com alta incidência, não devem ser usadas para representar diferenças ou similaridades entre documentos, uma vez que são muito comuns. O conteúdo semântico dessas palavras é insignificante, e não é relevante na análise dos documentos. Ao se retirarem essas palavras do índice, reduzimos o espaço de armazenamento e melhoramos seu desempenho durante a tarefa de *Text Mining*. Uma lista básica de palavras pode ser usada para formar uma lista de *stopwords*, chamada de *stoplist*, de forma a melhorar a tarefa de *Text Mining*. Um conjunto de 138 palavras consideradas universais para a língua inglesa forma, então, uma *stoplist* a ser usada na preparação dos dados [8].

Uma tabela de *stoplist* com palavras universais para a língua inglesa pode ser vista no apêndice A, junto com uma tabela de termos usados na Web.

A maioria dessas palavras são conjunções, preposições ou pronomes que pouco contribuem para a formação do índice [8]. De acordo com a especialidade do texto, outras listas mais apropriadas podem ser criadas, mas, neste caso, os documentos a serem trabalhados são de gêneros distintos.

Outro conjunto de palavras (tais como: *http*, *Web*, etc.) pode ser incluído à *stoplist* definida, já que levaremos em conta também documentos da *Web*. Essas palavras também pouco contribuem para a formação do índice.

Stemming

O processo de *Stemming* visa à redução ao radical (raiz), para que diferentes palavras com o mesmo significado não sejam contadas como palavras distintas. Normalmente, é usada uma palavra em uma consulta e as respostas podem trazer documentos não relevantes ao contexto exigido. Acontece que uma variante dessa palavra pode ser encontrada em outros documentos com maior relevância. Sendo assim, a consulta pode ser mais eficiente, se se trabalhar com os radicais das palavras ao invés da palavra original. Uma mesma palavra pode ter, dentre outras, variações como plurais, formas de gerúndio e acréscimo de sufixos [9,10,11,12,13]. A técnica consiste, então, em remover os sufixos e prefixos, para que palavras com o mesmo radical tenham significados similares [14,15].

O processo de *stemming* é dependente da linguagem, e as regras devem ser adaptadas para as diversas linguagens existentes, ou seja, os algoritmos devem ter suas regras modificadas de forma a atender a uma determinada linguagem. Para a língua inglesa, podemos identificar as palavras *cats*, *catlike*, *catty* como tendo o radical *cat*. Outro exemplo pode ser visto nas palavras *stemmer*, *stemming* e *stemmed* que possuem o radical *stem*. Assim, um algoritmo de *stemming* deve reduzir as palavras *fishing*, *fished*, *fish* e *fisher* à raiz *fish* [16].

A primeira publicação sobre o assunto surgiu em 1968, no trabalho sobre confluência, de Julie Beth Lovins. Em 1979, foi publicado o *stemming algorithm paper* [17], que foi a base para a criação do mais difundido algoritmo de *stemming*, o algoritmo de Porter [18]. Existem atualmente vários algoritmos de *stemming* disponíveis, alguns deles apresentados a seguir.

Classificação de dados

A classificação KNN ou K Nearest Neighbors (K vizinhos mais próximos) é considerado um método bastante simples e é baseado na identificação de grupos de dados com características similares para seu posterior agrupamento. Esse método utiliza o conceito de distância entre amostras. Para cada

amostra é criado um vetor (linha) contendo várias variáveis (colunas). A similaridade é medida através do cálculo das distâncias entre os pontos.

Segundo Rosa [19] "... se duas classes A e B possuem vários pontos em seus domínios, dado um ponto desconhecido x, este ponto será classificado em função da quantidade de pontos cujas distâncias forem as menores possíveis em relação às classes A e B. O cálculo da distância entre a amostra desconhecida em relação às amostras conhecidas pode ser feito através de algum dos seguintes métodos [19,20,21]:

- Distância de Manhattan

$$d(x, y) = \sum_{i=1}^{n-1} |x_i - y_i| \quad \text{Fórmula 1}$$

- Distância de Minkowski

$$d(x, y) = \sqrt[q]{\sum_{i=1}^{n-1} w_i |x_i - y_i|^q} \quad \text{Fórmula 2}$$

- Distância Euclidiana

É um caso particular da distância de Minkowski. Onde: $q = 2$ e $w_i = 1$

$$d(x, y) = \sqrt{\sum_{i=1}^{n-1} (x_i - y_i)^2} \quad \text{Fórmula 3}$$

O valor de K identifica o número de vizinhos mais próximos que serão usados para identificar a nova amostra em relação a todas as amostras. Sendo assim, quanto mais amostras de uma determinada classe se encontram próximas à amostra desconhecida, maior é a probabilidade de que esta amostra pertence àquela determinada classe. O cálculo mais usado é o da distância Euclidiana. Na figura 1, é apresentado o algoritmo do método de classificação KNN [19,21].

FIGURA 1

Algoritmo do método classificação KNN

```

Início
  Ler as n amostras de classes conhecidas (ac)
  Ler a amostra de classe desconhecida (ad)
  i ← 1
  Repita
    Calcular a distância de ad para aci
    vetor de distâncias ← a distância e a classe de aci
    i ← i + 1
  Até que (calcular todas as distâncias, isto é, i > n);
  Ordenar o vetor de distâncias pelos valores das distâncias
  Selecionar os K primeiros elementos do vetor de distâncias
  i ← 1
  classe0 ← 0
  classe1 ← 0
  Repita
    Comparar a classe
    Se (classe igual a 0)
      Então classe0 ← classe0 + 1
    Senão classe1 ← classe1 + 1
  Fim Se;
  i ← i + 1
  Até que (calcular todas as K amostras, isto é, i > K);
Fim.
```

Arquitetura

A arquitetura é baseada em portal para a internet. A construção da aplicação enfocará a separação das diferentes tarefas em três módulos, oferecendo a possibilidade de distribuição das mesmas entre diferentes servidores. O servidor principal atenderá aos usuários por meio de um portal para a internet. O usuário pedirá seu credenciamento no sistema e obterá um login. De posse deste login, o usuário criará, então, o seu perfil através da inclusão do resumo de uma tese de seu interesse. Esse perfil será usado para a classificação das publicações disponíveis no servidor, com auxílio dos resumos das respectivas publicações e posterior criação da lista ordenada de publicações relevantes. O sistema se incumbirá de efetuar a classificação e de gerar a lista, além de e atualizar a página de consulta do usuário, sempre que isso se fizer necessário, de forma automática.

A aplicação será responsável pela busca dos documentos no portal da CAPES, de duas diferentes formas: automática e manual. A forma automática que é o principal objetivo deste projeto é usada para obter novas publicações disponibilizadas. Já a forma manual, quando necessária, será apenas para a inclusão de novos periódicos, até então não contemplados, ou para uma aquisição forçada de documentos, tais como no caso do início do processo, obtendo uma massa de dados para a inicialização do processo.

A aplicação foi dividida em três diferentes módulos, sendo eles a busca de dados na internet feita pelo Módulo Buscador, a preparação e armazenamento de dados semi-estruturados feita pelo Módulo Preparador e, por fim, a pesquisa de documentos de maior interesse feita pelo Módulo Pesquisador.

O Módulo buscador

Um dos grandes problemas é a busca de documentos na internet. Estes documentos geralmente estão armazenados em diferentes locais e necessitam de muito tempo de busca. O Buscador irá acessar uma base de dados contendo os principais locais para a busca de publicações e trará de forma automática os arquivos encontrados e ficará em execução constante, buscando novas publicações sempre que forem incluídas no portal da CAPES. Essa ferramenta é formada por um navegador de internet preparado para acessar uma base de dados contendo os locais de busca, ao fazer acesso a esses locais e trazer o conteúdo do documento, este não precisará ser lido e seu conteúdo será armazenado na máquina servidora.

A aplicação permitirá que documentos avulsos possam ser inseridos, bastando que eles sejam armazenados na área de armazenamento e sejam cadastrados na base de dados. Informações como a data de inclusão e o local de origem fazem parte do contexto do documento publicado. Haverá, também, a possibilidade de se incluir novos locais de busca com a inclusão de novas localizações na base de publicação dos periódicos. Atualmente, já foram armazenados 322.008 artigos referentes à Ciência da Computação para avaliação de desempenho da aplicação.

O Módulo preparador

O Módulo Preparador é uma ferramenta de Text Mining, capaz de monitorar a área de armazenamento de documentos alimentado pelo Módulo Buscador. Suas principais funções são a identificação de novos arquivos disponíveis e a preparação dos dados para posterior processamento pelo Módulo Pesquisador. Esta fase visa a aumentar a velocidade de processamento, uma vez que para cada usuário haverá um processo Pesquisador diferente e este não necessitará preparar os dados, uma vez que eles

já se encontrarão prontos. Um documento, ao chegar ao servidor principal, será armazenado em um determinado diretório. Todos os documentos recebidos serão submetidos à etapa de preparação dos dados, que será dividida em três diferentes tarefas. Antes da primeira tarefa, o texto que foi armazenado será transferido do diretório de chegada para um diretório de trabalho. No diretório de trabalho, o texto é, então, submetido a um algoritmo capaz de extrair as palavras relevantes, eliminando todo tipo de dados que não formem palavras. O resultado será um documento XML contendo a lista de palavras, já eliminadas as duplicidades, mas, informando o número de ocorrências de cada uma dessas palavras no texto. O documento gerado é documentado na base de dados, com informações de data de aquisição; origem da publicação; localização do documento original e um *status* de leitura, com valor inicial de *não*. Ao ser lido pelo usuário, este, o *status* de leitura, é então alterado para *sim* e sua intenção é informar ao usuário se uma publicação já foi lida ou não. O resultado da preparação é armazenado em um outro diretório, no qual estará disponível para a realização das pesquisas pelo próximo módulo.

O Preparador realizará as tarefas de *Text Mining*:

- *Case Folding* — todas as letras de todas as palavras são transformadas em maiúsculas;
- *Stopwords* — o texto resultante é submetido ao algoritmo capaz de extrair as palavras, eliminando todo tipo de dados que não formem palavras, além de eliminar todas as palavras que estiverem na lista de *stopwords*, palavras essas que são irrelevantes para o contexto.
- *Stemming* — É tarefa final da fase de preparação dos dados e consiste na tarefa de *stemming* do documento, por meio do uso do algoritmo de *stemming* de Porter[22].

O Módulo pesquisador

Após a fase de preparação, o arquivo XML criado contém as ocorrências que têm importância para a classificação e a quantidade de ocorrências respectivas. Isto facilita a construção dos vetores de classificação (compostos das quantidades de ocorrências encontradas) e que serão comparados com o vetor criado a partir do perfil do usuário criado durante o seu cadastramento no sistema, por meio da técnica de classificação por KNN. O Módulo Pesquisador é responsável pelo processamento final e cria a lista de publicações de maior relevância para o usuário. A lista poderá ser parametrizada de forma a atender melhor as necessidades do usuário. Os principais itens a serem configurados são:

1. Número de documentos na lista de publicações de maior relevância;
2. Se o aplicativo irá verificar em todas as publicações armazenadas ou se o usuário irá selecionar aquelas que considera serem mais importantes para si;
3. Período de busca aos documentos pela data de aquisição, especificando período livre, por data de início e fim, ou ainda, a partir de quando e até a presente data.

A lista também é ajustável quanto à retirada de documentos já lidos e facilita, também, a localização deles, já que o usuário poderá acessar diretamente através de um *link* a publicação de seu interesse.

Considerações finais

A metodologia pretende fornecer ao usuário, meios de organizar e automatizar as tarefas de pesquisa em periódicos, aumentando consideravelmente a sua produção e permitindo a maximização do tempo despendido para esta pesquisa. Sua divisão em

módulos favorece, além do processamento distribuído, que novos algoritmos de preparação ou de classificação sejam incorporados às suas funcionalidades, aumentando, assim, as opções de condicionamento de tarefas e ações a serem executadas.

Referências

- ZANASI A., Ebecken N., Brebbia C.A.: Web Mining through the Online Analyst. Conference on Data Mining 2000, pp. 3-14, Cambridge, July WIT Press 2000.
- LOPES, M. C. S.: Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português., Rio de Janeiro, 2004.
- Google. Disponível em WWW por <http://www.google.com.br>.
- AGICHTEIN E., ZHENG Z.: Identifying "Best Bet" Web Search Results by Mining Past User Behavior. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press New York, NY, USA, 2006.
- JOACHIMS T., GRANKA, L., PANG, B., HEMBROOKE H. and Gay G.: Accurately Interpreting Clickthrough Data as Implicit Feedback. In the Proceedings of SIGIR, 2005.
- TAN A.H.: Text Mining the State of the Art and the Challenges. In Proceedings of the PAKDD 1999. Workshop on Knowledge Discovery from Advanced Databases, pages 65-70, Beijing, China, April 1999.
- VIEIRA, P. A.: A história da automatização do trabalho. Disponível em: <http://orbita.starmedia.com/~novosdebates/textos/t10.htm>. Acesso em 25 de outubro de 2005.
- KONCHADY M.: Text Mining Application Programming. 1a Edition, Charles River Media, Boston, Massachusetts, 2006.
- Maria Célia Santos Lopes, M. C. S.: Mineração de dados textuais utilizando técnicas de clustering para o idioma português. COPPE/ UFRJ, Rio de Janeiro, 2004.
- Chaves, M. S.: Um estudo e apreciação sobre algoritmos de stemming para a língua portuguesa. IX Jornadas Iberoamericanas de Informática. Cartagena de Indias - Colômbia, 11-15 agosto de 2003.
- FRANKS, W.B., BAEZA-YATES, R.: Readings in Information Retrieval: Data Structured Algorithms. Ed. Upper Saddle River, NJ: Prentice Hall, 1992.
- SPARK-JONES, K., WILLET, P.: Readings in Information Retrieval. San Francisco: Morgan Kaufmann, 1997.
- BAEZA-YATES, R.: Modern Information Retrieval. New York, N.Y.: Addison-Wesley, 1999.
- MARTHA, A. S. BARRA, P.S.C., CAMPOS, C. J. R.: Recuperação de Informações em Textos Livres de Prontuários do Paciente. Disponível em WWW por <http://www.sbis.org.br/sbis/arquivos/636.pdf>.
- WIVES, L.K.: Indexação de Documento.s Textuais. Disponível em WWW por <http://www.inf.ufrgs.br/~wives/publicacoes/IDT.pdf>.
- LOVINS J. B.: DEvelopment of a Stemming Algorithm. Macmechanical Translation and Computational Linguistics, 11, 22-31, 1968.
- PORTER, M. F.: Stemming Algorithm Parser. Computer Laboratory Cambridge England, 1979.
- PORTER, M. F.: The Porter Stemming Algorithm.. Disponível em WWW por <http://www.tartarus.org/~martin/PorterStemmer/index.html>.
- ROSA, J. L. A.: Classificação de Dados Através da Otimização do Método KNN-Fuzzy em Ambiente de Computação Paralela. COPPE/ UFRJ, Rio de Janeiro, 2003.
- MILLER, G.: Wordnet: An online lexical database, International Journal of Lexicography, 3(4):235-312, 1996.
- HAN J.: KAMBER, M., Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.
- PORTER, M. F.: An Algorithm for Suffix Stripping. Program, 14(3), 130-137, 1980.

Artigo enviado em 26/09/2008, aceito em 03/11/2008