CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Repository Open Access to Scientific Information from Embrapa

6º Congresso Interinstitucional de Iniciação Científica - CIIC 2012
13 a 15 de agosto de 2012– Jaguariúna, SP

EVOLUÇÃO DO MECANISMO DE BUSCA DO AINFO-CONSULTA COM USO DE THESAURUS AGROPECUÁRIO

IGOR J. P. **MARINHO**[1]**;** HENRIQUE T. M. **CARDONE**[2]**;** GLAUBER J. **VAZ**[3]

Nº 12610

RESUMO

Este trabalho propõe analisadores para sistemas de recuperação da informação (SRI) que exploram as relações de equivalência contidas em um thesaurus. Um protótipo com estes analisadores foi construído para o Ainfo-Consulta, um SRI que permite a realização de consultas à produção bibliográfica da Embrapa. Os analisadores propostos, que também podem ser utilizados em outros SRIs, visam a obter, principalmente, resultados com maior cobertura.

ABSTRACT

This paper proposes an analyzer for information retrieval systems (IRS) that exploits the equivalence relations contained in a thesaurus. We built a prototype with these analyzers for Ainfo-Consulta, an IRS which allows queries to the bibliographic production of Embrapa. The proposed analyzers can be used in other IRSs and aims to reach mainly results with greater recall.

[1]Estagiário: Graduação em Análise e Desenvolvimento de Sistemas, FT/UNICAMP, Limeira-SP, igorj27@gmail.com.

[2]Estagiário: Graduação em Análise e Desenvolvimento de Sistemas, FT/UNICAMP, Limeira-SP, htcardone@gmail.com.

[3] Orientador: Embrapa Informática Agropecuária, Campinas-SP, glauber@cnptia.embrapa.br.

INTRODUÇÃO

Um Sistema de Recuperação de Informação (SRI) é um sistema capaz de catalogar e recuperar documentos relevantes à consulta do usuário (KOWALSKI, 1997). Este trabalho trata da evolução do SRI Ainfo-Consulta, módulo do sistema Ainfo de gestão de acervos impressos e digitais de bibliotecas, que inclui todas as fases do fluxo de tratamento da informação, desde o registro das publicações até a exibição de resultados de consultas feitas pelos usuários (EMBRAPA INFORMÁTICA AGROPECUÁRIA, 2012a).

Um SRI pode ser dividido em duas grandes fases: indexação e busca. A primeira consiste em processar os dados originais e gerar um índice, uma estrutura de dados que permite rápido acesso às palavras armazenadas dentro dela. A indexação pode ser dividida em quatro etapas: obtenção de conteúdo, construção dos documentos, análise e indexação dos mesmos (HATCHER et al., 2010).

A primeira etapa consiste em definir e reunir o conteúdo que precisa ser indexado. No caso do Ainfo-Consulta, todas as informações da produção bibliográfica da Embrapa já estão armazenadas em um banco de dados.

Uma vez obtido o conteúdo, é necessário estabelecer as unidades que serão utilizadas pelo motor de busca. Na tecnologia selecionada, esta unidade é chamada de documento. No Ainfo-Consulta, cada documento construído corresponde a uma obra registrada, que contém dados como título, autoria, resumo, palavras-chaves e ano de publicação.

Durante a análise dos documentos, o motor de busca divide o texto em uma série de elementos atômicos (*tokens)* para depois indexá-los. Os analisadores tratam de diversas questões, como palavras compostas, palavras de baixa relevância (*stopwords*), uso de sinônimos e normalização de letras maiúsculas e acentos, entre outras.

A última etapa consiste na adição dos documentos nos arquivos de índice, que é realizada de forma transparente pela tecnologia utilizada. Estes arquivos são construídos para tornar as buscas mais eficientes.

Busca é o processo de procurar termos em um índice para encontrar documentos onde eles apareçam. Também pode ser dividido em quatro etapas: interface de usuário, construção e realização da consulta, e exibição dos resultados (HATCHER et al., 2010).

A interface de usuário oferece meios para se realizar as buscas. O Ainfo-Consulta (EMBRAPA INFORMÁTICA AGROPECUÁRIA, 2012b) é um sistema *web* que provê, além de um campo de entrada para consultas, opções para buscas

avançadas e outros recursos. O protótipo resultado deste trabalho também é um sistema *web*, mas, por enquanto, conta apenas com o campo de busca principal.

As consultas realizadas pelos usuários devem ser traduzidas para uma linguagem do motor de busca. A utilizada neste trabalho suporta pesquisas complexas e apresenta grande tolerância a erros de sintaxe. Sua configuração foi estabelecida para que as buscas sejam realizadas de maneira que os documentos contenham todos os termos da entrada.

Depois de construída a consulta, o índice é examinado e os documentos correspondentes são recuperados. Devido à API da plataforma, este complexo processo é realizado de maneira transparente para o desenvolvedor.

Finamente, os resultados são exibidos para que o usuário os utilize da maneira que lhe convier.

O objetivo deste trabalho é aumentar a qualidade dos resultados obtidos nas consultas, construindo melhores analisadores, tanto na indexação quanto na busca.

MATERIAL E MÉTODOS

O Ainfo-Consulta é construido com o Apache Solr, uma plataforma de busca que fornece diversas funcionalidades. O Solr utiliza em seu núcleo a biblioteca de código aberto Apache Lucene, a qual oferece recursos de indexação e busca de textos. O protótipo desenvolvido utiliza essas mesmas tecnologias, uma vez que a melhoria do Ainfo-Consulta é um dos objetivos deste trabalho.

Nessas tecnologias, a análise de texto é feita com um analisador (*analyzer*) que envolve um *tokenyzer* e zero ou mais filtros (*filters*)[4] (THE APACHE SOFTWARE FOUNDATION, 2012a). Enquanto o primeiro transforma um fluxo de texto em uma lista de *tokens*, os filtros a analisa e processa, adicionando, modificando ou removendo *tokens.* Assim, obtém-se uma nova lista de *tokens*.

Em SRIs, há analisadores envolvidos na indexação e na consulta. Os analisadores propostos neste trabalho, descritos na próxima seção, utilizam cinco filtros diferentes, com destaque ao que trata sinônimos. Para estabelecer as relações de equivalência entre termos agropecuáiros, foi utilizado o Thesagro, thesaurus brasileiro especializado em literatura agrícola (MINISTÉRIO DA AGRICULTURA E DO ABASTECIMENTO, 1999).

[4] *Analyzer*, *Tokenyzer* e *Filters* fazem parte da terminologia utilizada pela plataforma Lucene/Solr. Neste artigo, os termos 'Analisadores' e 'Filtros' correspondem, respectivamente, a 'Analyzers' e 'Filters', conforme são utilizados na plataforma.

OS ANALISADORES PROPOSTOS

A fim de obtermos melhores resultados nas buscas realizadas com o Ainfo-Consulta, construímos um protótipo em que novos analisadores foram utilizados. Em sua definição, utilizamos o *tokenizer StandardTokenizerFactory* e cinco filtros: *ISOLatin1AccentFilterFactory*, *LowerCaseFilterFactory*, *ShingleFilterFactory*, *SynonymFilterFactory* e *StopFilterFactory.* Esses componentes, que são disponibilizados pelo Apache Lucene com interface simples para o Solr, foram utilizados em uma ordem adequada para maior eficiência de indexação e busca.

As Figuras 1 e 2 exibem a ordem em que os componentes de análise foram organizados respectivamente nas fases de indexação e de consulta. Enquanto na primeira fase foram utilizados seis componentes, na segunda foram usados cinco, os quais podem ser identificados no início de cada passo. As figuras ilustram o caso em que se indexa o texto 'Mandioca: valor energético' e se consulta 'Teor de nutriente do Aipim'.

'Mandioca: valor energético'				
PASSO 1	StandardTokenizerFactory			
	Posição 0	Posição 1	Posição 2	
	Mandioca	valor	*energético*	
PASSO 2	ISOLatin1AccentFilterFactory			
	Posição 0	Posição 1	Posição 2	
	Mandioca	valor	*energetico*	
PASSO 3	LowerCaseFilterFactory			
	Posição 0	Posição 1	Posição 2	
	mandioca	valor	energetico	
PASSO 4	ShingleFilterFactory - (3)			
	Posição 0	Posição 1	Posição 2	
	mandioca ***mandioca valor*** ***mandioca valor energetico***	valor ***valor energetico***	energetico	
PASSO 5	SynonymFilterFactory - (Synonym.txt)			
	Posição 0	Posição 1	Posição 2	Posição 3
	mandioca ***aipim*** ***macaxeira*** mandioca valor mandioca valor energetico	valor	valor energetico ***valor proteico*** ***teor de nutriente***	energetico
PASSO 6	StopFilterFactory - (StopWord.txt)			
	Posição 0	Posição 1	Posição 2	Posição 3
	mandioca aipim macaxeira mandioca valor mandioca valor energetico	valor	valor energetico valor proteico teor de nutriente	energetico

FIGURA 1: Análise da indexação

'Teor de nutriente do Aipim'					
PASSO 1	StandardTokenizerFactory				
	Posição 0	Posição 1	Posição 2	Posição 3	Posição 4
	Teor	de	nutriente	do	Aipim
PASSO 2	ISOLatin1AccentFilterFactory				
	Posição 0	Posição 1	Posição 2	Posição 3	Posição 4
	Teor	de	nutriente	do	Aipim
PASSO 3	LowerCaseFilterFactory				
	Posição 0	Posição 1	Posição 2	Posição 3	Posição 4
	teor	de	nutriente	do	*aipim*
PASSO 4	ShingleFilterFactory - (3)				
	Posição 0	Posição 1	Posição 2	Posição 3	Posição 4
	teor ***teor de*** ***teor de nutriente***	*de* ***de nutriente*** ***de nutriente do***	nutriente ***nutriente do*** ***nutriente do aipim***	*do* ***do aipim***	aipim
PASSO 5	StopFilterFactory - (StopWord.txt)				
	Posição 0	Posição 1	Posição 2	Posição 3	Posição 4
	teor teor de teor de nutriente	de nutriente de nutriente do	nutriente nutriente do nutriente do aipim	do aipim	aipim

FIGURA 2: Análise da busca

A cada passo, exibe-se a lista de termos resultante da aplicação do componente correspondente. São destacadas nas figuras as células onde houve alterações, com indicação do que foi modificado no termo[5].

Nas análises de indexação e de consulta, o primeiro passo é realizado por um *tokenizer*, que transforma uma cadeia de caracteres em uma lista de *tokens*. O *StandardTokenizerFactory* utiliza espaços em branco e caracteres de pontuação como separadores. No entanto, endereços de e-mail e de páginas *web* são reconhecidos adequadamente. Na Figura 1, este componente cria três posições, cada qual com uma palavra do texto de entrada. Tal divisão é realizada devido à identificação dos espaços em branco e do caractere ":", elementos analisados como delimitadores, que, portanto, não são diretamente inseridos no índice e nem incluídos nos termos indexados. Neste exemplo, três *tokens* foram identificados: 'Mandioca', 'valor' e 'energético'.

É importante ressaltar que o *tokenizer* utilizado na consulta deve ser o mesmo que o da indexação, para que um *token* identificado na consulta possa ser encontrado no índice. Para ilustrar essa importância, considere que, no lugar do *StandardTokenizerFactory,* fosse utilizado, na análise de indexação, o *WhiteSpaceTokenizerFactory,* que estabelece apenas espaços em branco como delimitadores. Em vez de 'Mandioca', seria extraído o *token* 'Mandioca:', com o

[5]Os termos que sofreram alterações estão representados em itálico nos passos corrente e anterior, com os caracteres alterados destacados em negrito. Termos acrescentados a uma posição são colocados em negrito e itálico, enquanto os removidos são sublinhados no passo anterior e ocultados no passo corrente.

caracter ':'. Mesmo com uma consulta a 'Mandioca: valor energético', não haveria total casamento entre as frases indexada e consultada porquê o *StandardTokenizerFactory* incluiria o termo 'Mandioca' na lista de *tokens*, enquanto o *WhiteSpaceTokenizerFactory* incluiria 'Mandioca:'.

Depois de obtida a lista de *tokens,* ela é tratada por filtros do Solr que são aplicados em sequência. O primeiro, *ISOLatin1AccentFilterFactory*, remove acentos para tornar a busca mais simples e tolerante a erros de acentuação. Na Figura 1, por exemplo, a palavra 'energético' é substituída por 'energetico', sem acento agudo. A exclusão dos acentos deve ser feita tanto na indexação quanto na busca. Desta maneira, o que consta nos índices e nas buscas são variações dos termos sem acentuações.

De forma análoga, o filtro *LowerCaseFilterFactory* é aplicado tanto na indexação quanto na busca. Ele substitui letras maiúsculas por minúsculas, como pode ser observado no passo 3 da Figura 1, com o termo 'Mandioca', e da Figura 2, com os termos 'Teor' e 'Aipim'. Letras maiúsculas, portanto, não são encontradas nos termos dos índices e das buscas.

Os outros três filtros utilizados estão intimamente relacionados, e a ordem em que se apresentam é fundamental para uma adequada utilização de sinonímias. O filtro *SynonymFilterFactory* foi concebido justamente para isso. Seu principal parâmetro, representado na Figura 1 por '*Synonym.txt*', é o nome do arquivo texto que contém listas de palavras sinônimas em uma sintaxe específica e apropriada. No caso do Ainfo-Consulta, esta lista deve ser gerada a partir das relações de equivalência presentes no Thesagro, uma vez que trata de produção bibliográfica relacionada à Agropecuária. Outros tesauros, no entanto, podem enriquecer esta lista de sinônimos. O *SynonymFilterFactory* adiciona termos equivalentes em uma mesma posição, conforme pode ser observado no quinto passo da Figura 1. São inseridos os termos 'aipim' e 'macaxeira', sinônimos de 'mandioca', e ainda 'valor proteico' e 'teor de nutriente', equivalentes a 'valor energetico'.

Como pode ser notado neste último exemplo, equivalências não são estabelecidas apenas para palavras isoladas, mas também para termos que envolvem mais de uma palavra. O *ShingleFilterFactory* é fundamental nessa questão, pois, conforme ilustra o passo quatro das análises, permite a formação de termos compostos. Seu principal parâmetro, três no exemplo utilizado, determina o máximo de termos que podem ser concatenados.

Por fim, o filtro *StopFilterFactory* elimina palavras irrelevantes *(stopwords)* para as análises. Um parâmetro, que no exemplo tem valor '*StopWords.txt'*, determina o

arquivo que contém a lista destas palavras. Na Figura 2, 'de' e 'do' foram eliminadas no último passo por constarem em tal lista, que normalmente inclui artigos, preposições, conjunções e outras palavras que não acrescentam valor semântico. A não remoção dessas palavras causaria impacto de várias maneiras (THE APACHE SOFTWARE FOUNDATION, 2012b). Em primeiro lugar, o tamanho do índice ficaria bem maior. Em relação ao desempenho, consultas a frases que envolvem *stopwords* tenderiam a ficar bem mais lentas. Finalmente, por se tratar de palavras que ocorrem com muita frequência, há uma influência desproporcional à sua importância no algoritmo de ranqueamento dos resultados. O uso do filtro *ShingleFilterFactory* em passo anterior faz com que as *stopwords* não sejam totalmente ignoradas, mas consideradas em seu contexto e associadas a outras palavras, o que contribui para a qualidade dos resultados.

Para que a exploração das relações de equivalência do Thesagro sejam adequadamente exploradas, o *StopFilterFactory* deve ser o último filtro da sequência de análise porque as palavras irrelevantes são mantidas até se obter o resultado do *SynonymFilterFactory*, que pode usar termos compostos formados por estas palavras. O uso dos filtros *ISOLatin1AccentFilterFactory* e *LowerCaseFilterFactory* no início do analisador também facilita sua implementação, uma vez que as relações de equivalência podem ser todas consideradas para variações dos termos que não apresentam acentuação e nem letras maiúsculas.

A sequência de filtros nas análises de indexação e de consulta difere apenas no *SynonymFilterFactory,* que é usado apenas na primeira. Diferentemente dos filtros comentados anteriormente, não é recomendável defini-lo nas duas etapas, por caracterizar redundância (SMILEY e PUGH, 2009). Além disso, o Apache Lucene utiliza um algorítimo de ranqueamento baseado na quantidade de vezes que um termo aparece no índice. Desta forma, se o filtro que trata sinônimos fosse utilizado durante a busca, este ranqueamento atribuiria um valor maior a um documento contendo um sinônimo de menor frequência (SMILEY e PUGH, 2009), o que não é desejado neste caso. Portanto, o filtro *SynonymFilterFactory* foi colocado na fase de indexação, fazendo com que termos sinônimos tenham a mesma frequência no conjunto de documentos.

RESULTADOS E DISCUSSÃO

A qualidade da busca geralmente é descrita utilizando-se métricas de precisão e cobertura. Enquanto a primeira mede a capacidade do sistema em manter os documentos irrelevantes fora do resultado de uma consulta, a segunda mede sua

capacidade em recuperar os documentos que são relevantes para o usuário. Outro critério de grande importância é o grau de relevância de um documento em uma busca, fundamental para o algorítimo de ranqueamento dos resultados.

Para a proposta dos analisadores, consideramos todos esses critérios, mas com ênfase na cobertura. Assumindo que também são relevantes documentos com palavras equivalentes àquelas utilizadas na consulta, os resultados obtidos com os analisadores propostos apresentam maior cobertura. Além disso, favorecemos a precisão e o ranqueamento, devido à maneira como manipulamos *stopwords* e termos compostos por múltiplas palavras e também à opção de tratar sinônimos na indexação.

CONCLUSÃO

Os analisadores propostos favorecem a obtenção de melhores resultados em SRIs, especialmente no Ainfo-Consulta. As tecnologias aqui consideradas já são utilizadas neste sistema, o que não demanda tantos ajustes para sua alteração. Uma vez que constatamos a viabilidade e a qualidade destas mudanças, faz-se necessária sua validação por parte dos usuários do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

EMBRAPA INFORMÁTICA AGROPECUÁRIA. **Ainfo**. Disponível em: <http://www.ainfo.cnptia.embrapa.br/wiki/index.php/Página_principal>. Acesso em 14 jun. 2012a.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. **Ainfo-Consulta**. Disponível em: <http://ainfo.cnptia.embrapa.br/consulta/>. Acesso em 14 jun. 2012b.

HATCHER, E.; GOSPODNETIC, O.; McCANDLESS, M. **Lucene in Action**. 2. ed. Connecticut: Manning Publications, 2009. 475p.

KOWALSKI, G. **Information Retrieval Systems: Theory and Implementation**. Boston: Kluwer Academic Publishers, 1997. 282p.

MINISTÉRIO DA AGRICULTURA E DO ABASTECIMENTO. Secretaria de desenvolvimento rural. CENAGRI. **Thesagro**: Thesaurus Agrícola Nacional. Brasília, DF, 1999.

SMILEY D.; PUGH E. **Solr 1.4 Enterprise Search Server**. Birmingham, UK: Packt Publishing, 2009. 317p.

THE APACHE SOFTWARE FOUNDATION. **Solr Wiki**: Analyzers, Tokenizers and Token Filters. Disponível em:

<http://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters>. Acesso em 25 jun. 2012a.

THE APACHE SOFTWARE FOUNDATION. **Solr Wiki**: Language Analysis. Disponível em: <http://wiki.apache.org/solr/LanguageAnalysis>. Acesso em 25 jun. 2012b.