

**Universidade Federal do Rio Grande do Sul
Instituto de Informática
Curso de Pós-Graduação em Ciência da Computação**

Disciplina de Sistemas de Banco de Dados

Indexação de Documentos Textuais

por

Leandro Krug Wives

Professora Lia Golendziner

Porto Alegre, Junho de 1997

Índice

Resumo	3
Introdução	4
Metodologia Básica de um Sistema de Recuperação de Informações.....	4
Metodologia de Indexação.....	6
Indexação Automática	6
Aperfeiçoando a indexação automática.....	8
Técnicas de localização	10
Expressões Booleanas	10
Term Weighting.....	12
Relevance Feedback	13
Thesaurus.....	14
Integrando Conceitos (BD Tradicional e BD Textual).....	14
Conclusão.....	16
Bibliografia	17

Resumo

Os Bancos de Dados tradicionais são ótimos para armazenar informações que possuam estruturas e relações fáceis de serem identificadas ou determinadas. Identificar uma estrutura pode ser, entre outras coisas, definir seu domínio, tamanho, conteúdo e contexto.

Nem sempre se encontram informações que atendam estas necessidades. Documentos, figuras e gráficos são exemplos de informações que possuem conteúdos e semânticas variáveis. Isso significa que estas informações não são facilmente transformadas em Tabelas, que possuem registros e campos. Este fato dificulta a localização e o relacionamento destes tipos de informação com outras estruturas.

Apesar de todas estas dificuldades, estes tipos de informação são largamente utilizados atualmente (em programas multimídia e nas páginas WWW da Internet), e devem ser objeto de estudo da comunidade científica.

Este trabalho demonstra métodos para a indexação e localização de documentos (informações textuais). Não trata de outros tipos de informações também ainda deficientes neste aspecto. Os métodos aqui apresentados são a base da maioria dos métodos atuais¹ de localização de documentos, e portanto são a base para aqueles que tem interesse nesta área. As pessoas que desejarem aprofundar-se na área, devem procurar métodos mais eficazes nas referências bibliográficas incluídas ao final deste trabalho.

¹ Os metodos atuais são muitas vezes apenas variações ou aperfeiçoamentos dos métodos aqui apresentados. Isso pode ser verificado consultando a bibliografia apresentada no decorrer deste trabalho.

Introdução

Muitas pessoas acreditam que a área de recuperação de informações textuais é uma área nova. Esta idéia talvez tenha surgido com a *WEB* (um dos serviços oferecidos pela *Internet*), onde milhares de informações, dispostas em forma de *páginas* (documentos textuais), estão disponíveis.

Com a implantação da rede global (*Internet*) em quase todos os locais, interligando pessoas de diversas partes do mundo, os usuários viram que as informações estão por todo o lado, de forma desordenada. Algum meio de catalogar estas informações deveria surgir, e de fato surgiram vários: *Altavista*², *WAIS*³, *Yahoo*⁴, entre outros. Mas não foi com a *Internet* que esta preocupação de catalogar e integrar informações teve início de fato.

Há algum tempo, cientistas estudam meios de catalogar informações textuais (o tipo de informação presente na *WEB*). Na verdade desde que a informática surgiu esta preocupação existe. O pesquisador *Gerard Salton* [ACM96a], por exemplo, vem trabalhando nesta área desde a década de 60 e publicou mais de 150 artigos.

Portanto, técnicas já bem definidas e comprovadas existem há alguns anos. Os trabalhos científicos atuais utilizam muitas destas técnicas. O *ACM/SIGIR*, (Special Interest Group on Information Retrieval, da ACM) promove uma conferência internacional de pesquisa e desenvolvimento em Recuperação de Informações que ocorre anualmente, e é um dos meios de divulgação dos estudos na área. Verificando-se seus artigos ([FOX95], [ACM96b]), constata-se que os métodos atuais buscam aperfeiçoar os métodos mais antigos, permanecendo a metodologia básica.

Metodologia Básica de um Sistema de Recuperação de Informações

A busca de informações textuais é diferente da tradicional. Segundo *Salton* [SAL83], os Bancos de Dados Tradicionais⁵ preocupam-se com o armazenamento, manutenção e a recuperação de informações disponíveis explicitamente no sistema. Ao contrário dos Bancos de Dados Textuais, onde a informação está implícita (muitas vezes escondida ou difícil de ser localizada), em forma de *Linguagem Natural*⁶. Neste último, não há *Campos*, capazes de identificar os *Atributos* específicos de determinados *Registros*, ou seja, as informações não estão armazenadas em *Tabelas* como em Bancos de Dados Relacionais.

Por exemplo, para se buscar informações sobre determinada pessoa em um Banco de Dados (BD) tradicional, basta percorrer no BD a Tabela que possui o atributo *Nome* e localizar o Registro (*Tupla*) que possui o nome da pessoa desejado (em [KOR94] podem ser obtidos outros exemplos e informações sobre Bancos de Dados Tradicionais).

Caso este Banco de Dados fosse textual, os dados não estariam distribuídos de uma forma *tabular*. Até mesmo porque o texto é uma seqüência de caracteres, não existindo atributos. não há como saber o que é um nome em um documento, a não ser que se faça uma análise de *Linguagem Natural*, e se descubra o que pode vir a ser um nome - o que não é fácil de ser feito (*Salton* [SAL83], dá maiores detalhes sobre as diferenças entre os vários tipos de Sistemas de Informação).

² Endereço Internet: <http://www.altavista.digital.com>.

³ Wide Area Information System - Um sistema acadêmico de busca de informações muito difundido na Internet.

⁴ Endereço Internet: <http://www.yahoo.com>.

⁵ Relacional, Hierárquico, Redes.

⁶ Linguagem Natural devido ao fato de ser a linguagem normalmente utilizada pelo homem para comunicar-se (exemplo: Português, Inglês, Alemão...).

Logo, para localizar as informações sobre determinada pessoa, em um Banco de Dados Textual, seria necessário analisar caracter-por-caracter do texto até que a seqüência de caracteres correspondente ao nome fosse localizada.

Este tipo de análise (caracter-a-caracter) não é conveniente, é necessário haver alguma forma mais eficiente de acesso aos documentos. Para isto, os documentos precisam de algo que os identifique entre os demais, permitindo a sua localização.

Sabe-se que os documentos textuais possuem um contexto, isto é, um assunto. Este assunto pode ser identificado pelas palavras (termos) que este documento contém, portanto, o termo é o meio de acesso a um documento.

Decorrente disso, um sistema de Recuperação de Informações (ou banco de dados textual) tem como base a seguinte teoria, proposta por *Salton* [SAL83]: perguntas (*Queries*) são submetidas pelo usuário. Perguntas estas baseadas em termos (palavras) que identificam a idéia desejada por este usuário. Os documentos são identificados pelos termos que eles contém, portanto, a localização de um documento desejado pelo usuário dá-se a partir da identificação da similaridade entre o(s) termo(s) fornecido(s) pelo usuário e os termos que identificam os documentos contidos na base de dados. A figura a seguir representa esquematicamente esta teoria:



Figura - Função *Similaridade*

Esta função *Similaridade* busca identificar uma relação entre os termos da *Query* (consulta) e os termos dos documentos. Teoricamente pode ser feita uma comparação direta entre estes termos, mas na prática é difícil estabelecer esta relação de similaridade entre estes termos devido a alguns problemas.

Um destes problemas é analisado por *Chen* em vários de seus trabalhos. O que pode ocorrer é que as palavras utilizadas pelo sistema (palavras contidas nos documentos) sejam diferentes das palavras utilizadas pelo usuário, mesmo que estas palavras (sinônimos) representem a mesma idéia. Este problema é conhecido por *Problema do Vocabulário* [CHE94a], e ocorre geralmente quando os usuários desconhecem o sistema, ou possuem um conhecimento superficial dos assuntos que estão tentando localizar.

Há ainda o problema da *Busca Incerta* (*Search Uncertainty*), ou seja, pode ocorrer que os usuários não saibam quais são as melhores palavras que identificam o assunto que querem localizar. Por consequência, acabam não recuperando informações precisas. Este problema também é discutido por *Chen* [CHE94c], *Salton* [SAL83] e outros autores.

Estes problemas fazem com que sejam recuperados muitos documentos, ou documentos de assuntos variados (pois o termo é muito abrangente), ou ainda, podem recuperar informação alguma.

É buscando solucionar estes problemas (e alguns outros) que mecanismos de mapeamento entre os diferentes termos similares foram criados. *Salton* [SAL83], cita vários sistemas universitários e comerciais que se utilizam destes mecanismos: STAIRS (IBM), Dialog System (Lookhead Information Systems), BRS (State University of New York), MEDLARS (National Library of Medicine), SMART (Cornell University). Em [ACM96a] são citados mais alguns: WIN (West Publishing Company), DOWQUEST (Dow Jones Newswire), WAIS, e um muito conhecido, o INQUERY. Nem sempre estes sistemas conseguem satisfazer o usuário, mas foram a base para as técnicas atuais e das que estão por vir. A metodologia básica destes sistemas é discutida a seguir.

Metodologia de Indexação

Após estas definições e estudos iniciais, percebe-se portanto que o meio de acesso aos documentos são as palavras que ele contém. Para tornar possível o acesso a estas palavras, é preciso colocá-las em uma estrutura auxiliar - o índice, isso porque fica inviável pesquisar todos os textos toda a vez que for requisitada uma consulta.

Salton [SAL83] diz que a indexação é o processo de mapeamento citado anteriormente. Ela é o meio pelo qual a função Similaridade vai comparar os termos da *Query* com os termos presentes nos documentos, e após localizar os documentos relacionados com o assunto desejado pelo usuário.

Indexação Automática

O método de *Salton* é um método bastante aceito, inclusive vários outros trabalhos utilizam este método apesar de poderem variar em alguns aspectos (os trabalhos de *Chen*, por exemplo).

Este método é conhecido por *Indexação Automática*, e constitui-se de várias etapas. Ao final das etapas, os termos resultantes são adicionados a um arquivo de índice cuja estrutura geralmente é baseada em *Arquivos Invertidos* (ou *Listas Invertidas*). Segundo *Salton*, outros tipos de arquivos podem ser utilizados, mas a experiência mostra que este tipo de estrutura é uma das mais eficientes para a indexação de documentos. Na figura abaixo é apresentado um exemplo da estrutura de uma lista invertida.

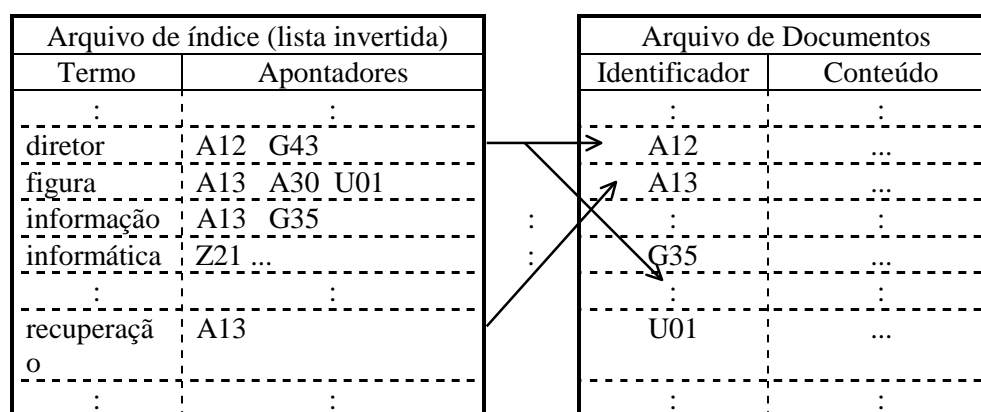


Figura - Estrutura de uma Lista Invertida

Basicamente, a estrutura permite que um único termo aponte para vários documentos. Maiores detalhes sobre esta estrutura podem ser obtidos em [SAL83].

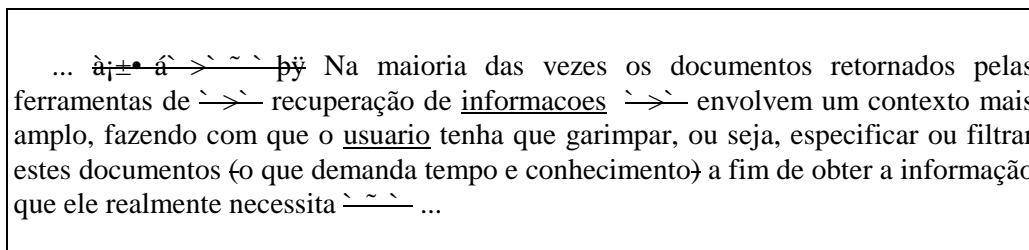
As etapas geralmente utilizadas na Indexação Automática são:

1. Identificação de palavras - Identifica as palavras nos documentos a serem indexados. Nada mais é do que a identificação de palavras, analisando-se as seqüências de caracteres no texto. *Salton* aconselha fazer um *Dictionary lookup*, ou seja, comparar as seqüências de caracteres retiradas do texto com um dicionário a fim de validar se estas palavras realmente existem. Este processo de validação torna-se bastante útil, especialmente quando o documento apresenta muitos caracteres inválidos ou palavras com erros gramaticais. As seqüências de caracteres inválidas devem ser eliminadas e as palavras com erros corrigidas. Pode-se aplicar ainda um processo de filtragem naqueles arquivos que possuem formatos de texto específicos, a fim de eliminar as seqüências de controle e/ou formatação de texto.

O dicionário pode também auxiliar a identificação de termos específicos, quando se deseja utilizar palavras pré-definidas no índice, evitando que palavras desconhecidas sejam identificadas (ou seja, evita a utilização de um vocabulário descontrolado).

Um simples *Analizador Léxico* que identifique seqüências de caracteres e monte palavras pode ser utilizado.

A figura abaixo apresenta o trecho de um documento com diversas seqüências de caracteres. As seqüências marcadas são seqüências inválidas, que não devem passar pela fase de identificação de palavras. As demais seqüências podem ser identificadas como termos válidos. Os termos sublinhados são termos identificados como incorretos pelo dicionário, e devem ser corrigidos. Os caracteres de pontuação são desprezados.



... à;±• á`->`~`~`þÿ Na maioria das vezes os documentos retornados pelas ferramentas de `->` recuperação de informacoes `->` envolvem um contexto mais amplo, fazendo com que o usuario tenha que garimpar, ou seja, especificar ou filtrar estes documentos (o que demanda tempo e conhecimento) a fim de obter a informação que ele realmente necessita `~`~` ...

Figura - Identificação de termos válidos

2. Remoção de Stop-words - Nem todas as palavras dos documentos podem ser adicionadas na estrutura de índice. As palavras que aparecem em todos os documentos ou na maioria deles, são um exemplo. Isso porque a utilização de uma palavra com estas características não é capaz de selecionar documentos relativos a um assunto específico.

As preposições são um exemplo deste tipo de palavra, pois são termos que servem para fazer o encadeamento de idéias e palavras, portanto, são termos inerentes a linguagem, e não ao conteúdo dos documentos.

Logo, as palavras que aparecem em muitos documentos não devem ser indexadas, pois sua utilização compromete a precisão e a eficiência do sistema.

Nos sistemas já implementados, foi construída uma estrutura (uma lista) contendo todas as palavras que não devem ser indexadas. A esta estrutura foi atribuído o nome de "*stop-list*", e as palavras presentes nesta lista são conhecidas como *Stop-words*.

O processo de obtenção das *stopwords* pode ser manual, onde o projetista do sistema avalia quais palavras devem ou não ser indexadas (o que varia de língua para língua, ou até mesmo entre sistemas). Há ainda a possibilidade de se montar esta lista automaticamente, verificando-se quais são as palavras com maior freqüência (que aparecem em mais documentos), e selecionando-as como *stop-words*.

Então, após uma palavra ser reconhecida no processo de indexação, sua presença na *Stop-list* é verificada. Caso exista na lista de palavras negativas, ela não é adicionada ao índice.

A figura abaixo apresenta o documento resultante da etapa anterior, após ser validado por uma *stop-list*. Neste caso a lista de *Stop-words* contém artigos, preposições, conjunções e algumas seqüências de caracteres que não devem ser adicionadas ao índice por possuírem freqüência elevada.

... Na maioria das vezes os documentos retornados pelas ferramentas de recuperação de informações envolvem um contexto mais amplo fazendo com que o usuário tenha que garimpar ou seja especificar ou filtrar estes documentos e que demanda tempo e conhecimento a fim de obter a informação que ele realmente necessita ...

Figura - Identificação de *Stop-Words*

Com estas etapas já é possível criar-se índices que localizem documentos a partir da comparação direta entre os termos da consulta do usuário e os termos presentes nos documentos. Este é um método ainda ineficiente, e algumas técnicas adicionais podem ser utilizadas a fim de melhorá-lo.

Aperfeiçoando a indexação automática

As técnicas a seguir permitem ao sistema melhorar sua eficiência, mas possuem alguns inconvenientes. São técnicas recomendadas por *Salton*, mas podem ser feitas a parte. Alguns sistemas não as utilizam (por exemplo: [WIV96a], [WIV96b]), e conseguem recuperar documentos de forma razoável. Há ainda quem diga que a utilização destas técnicas não compensa, como é o caso de *Riloff* [RIL95]. *Riloff* realizou alguns testes, e chegou a conclusão de que em alguns casos a eficiência do sistema pode ser pior se o *Stemming* (ver adiante) for utilizado. Comenta ainda que até a fase de *Stop-word remotion* pode comprometer o sistema. Segundo *Riloff*, estas palavras tem papel importante em alguns domínios.

Os estudos realizados até o momento sobre a eficiência destes métodos de remoção de palavras não indicam ainda uma conclusão. Utilizá-los ou não depende muito do sistema e dos próprios documentos. *Church* [CHU95] apresenta algumas comparações de eficiência entre a utilização ou não destes métodos. De qualquer modo eles são apresentados a seguir.

3. Word Stemming - Identificação de radicais (agrupamento de palavras similares), a fim de melhorar a eficiência e solucionar o problema do vocabulário. É uma técnica que procura reduzir a variância morfológica de um termo, e portanto depende muito da linguagem utilizada nos documentos (técnicas elaboradas para uma língua não podem ser utilizadas em outra). Vários experimentos para a língua Inglesa foram realizados, e funcionam de maneira eficiente. O mais recente destes experimentos é [KRA96].

A técnica consiste em identificar os radicais das palavras, e adicioná-las no arquivo de índice desta forma. Uma maneira de identificar os radicais das palavras é remover seus sufixos e prefixos. Outro exemplo é a eliminação dos plurais das palavras.

Assim, todas as palavras que possuem o mesmo radical, e portanto com significados similares (mas categorias diferentes de linguagem: adjetivo, verbo, advérbio...) são reconhecidas pelo mesmo identificador (as palavras são armazenadas de uma só forma - o radical), facilitando a consulta. A desvantagem deste método é que ele pode acabar utilizando palavras muito abrangentes, não recuperando documentos específicos (de termos específicos).

4. Word Phrase Formation - Formação de Frases-termo. Junta as palavras adjacentes para formar novos termos, buscando solucionar o problema dos termos abrangentes, pois as idéias estão agrupadas em contextos, e palavras compostas geralmente categorizam melhor os assuntos (os termos passam a ser mais específicos).

A utilização de palavras mais específicas consegue fazer com que o sistema recupere documentos de forma mais precisa, justamente pelo fato destas palavras aparecerem em um

número menor de documentos (geralmente os documentos de contextos específicos, utilizam termos específicos).

Para exemplificar, pode-se imaginar uma pessoa buscando informações sobre *programas de computador*. Esta pessoa poderia formular uma consulta utilizando a palavra *Programa*, o que poderia ocasionar a recuperação de muitos documentos, que contém a palavra *programa*, mas que não pertencem ao contexto *computador*.

Uma solução para este problema, seria utilizar o termo composto “programa de computador”, ou simplesmente “programa computador” (pela eliminação da preposição). Esta *frase*, contextualiza melhor a palavra *programa*, tornando-a menos abrangente e mais específica. Agora os documentos retornados por esta *frase-termo*, fariam parte somente do contexto *programa de computador*.

Deve-se tomar o cuidado para não confundir o conceito de *frase-termo* com a utilização das duas palavras de forma independente. Ou seja, caso o usuário não tenha de alguma forma especificado que as duas palavras devem aparecer juntas, ou o sistema não possua alguma técnica que unifique as duas palavras, a consulta pode se tornar ainda mais abrangente. Isso significa que seriam retornados tanto documentos que tratam do assunto *computador* quanto documentos que tratam do assunto *programa*.

Em geral não é necessário armazenar as palavras de forma composta, pois este processo de unificação das palavras exige tempo. *Salton*, em seus estudos, e *Croft* [CRO82] recomendam que ela não seja utilizada, pois não aumenta de forma considerável a eficiência do sistema. O que pode ser feito é o armazenamento da informação sobre as distâncias entre as palavras de um mesmo documento, e deixar com que a técnica de consulta avalie se as palavras são ou não adjacentes (no livro de *Salton* [SAL83], é descrita uma técnica de consulta que utiliza a distância entre as palavras).

O diagrama abaixo resume o processo total de Indexação. Pode-se ver que os documentos são fornecidos à ferramenta de indexação, e ao final é produzido um arquivo de índices que consegue localizar os documentos apresentados.

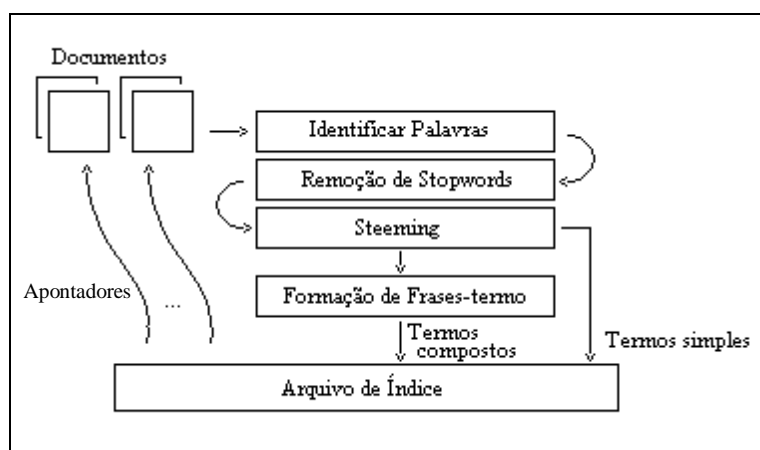


Figura - Etapas do Processo de Indexação Automática

É importante salientar que este tipo de indexação automática é o mais simples (pode ser chamada de *FullText*, pois analisa todo o documento). Esta técnica não considera a semântica do documento e nem a posição sintática das palavras nas orações. Baseando-se nestas duas últimas considerações surgiram outras formas de indexação mais complexas: a indexação *Sintática*⁷ e a indexação *Semântica*⁸.

⁷ É possível utilizar-se de uma análise sintática para descobrir quais são as palavras mais importantes dentro de uma oração. A linguagem do documento permite que este tipo de análise seja feito, já que as orações (a exemplo do português) possuem posições sintáticas pré-definidas para os

Técnicas de localização

De posse da estrutura de índice, pode-se construir um sistema que seja capaz de identificar os documentos da base de dados que são relacionados com determinado termo (identificador). Existem vários métodos que realizam esta tarefa (a tarefa que corresponde à função similaridade no esquema da Figura 1). Alguns destes métodos serão apresentados a seguir, outros, poderão ser conseguidos nos anais de congressos e revistas da área, mas é bom lembrar que a maioria destes métodos corresponde a variações ou aperfeiçoamentos dos métodos a seguir.

Expressões Booleanas

Normalmente o usuário fornece um termo ao sistema, e este termo é utilizado para localizar os documentos correspondentes na estrutura de índice. Algumas vezes o usuário pode querer restringir o número de documentos retornados pelo sistema, ou diminuir a abrangência de sua consulta, contextualizando melhor sua consulta. Isto é feito fornecendo mais de um termo ao sistema de consultas.

Voltando ao exemplo dos termos *programa* e *computador*, onde o usuário deseja localizar documentos sobre *programas de computador*, é necessário que este usuário indique de alguma forma ao sistema que as duas palavras precisam aparecer no mesmo documento. Isto é feito através de *Expressões Booleanas*: *AND* (e), *OR* (ou) e *NOT* (não/negação).

Estas *expressões booleanas* oferecem ao usuário três possibilidades:

1. Interseção - O operador *AND* permite que o usuário indique que os documentos retornados devem conter ambas as palavras. A maneira utilizada pelo sistema para localizar documentos que atendam estas necessidades é localizar o conjunto de documentos que contém a palavra *X*, localizar o conjunto de documentos que contém a palavra *Y*, e retornar os documentos correspondentes a Interseção destes dois conjuntos.

termos (sujeito, predicado, local do verbo...), e alguns destes termos são mais importantes do que os outros (seus auxiliares). Somente os termos *importantes* são adicionados a estrutura de índice. Esta técnica exige uma *Base de Conhecimento* que contenha todas as combinações sintáticas possíveis, além de exigir mais poder computacional e tempo. Portanto, geralmente não é utilizada. Um estudo sobre o assunto pode ser encontrado em [SAL88] e [SAL83].

⁸ A indexação semântica, baseia-se no princípio de que o documento já possui estruturas de formatação que indicam a semântica dos termos. Por exemplo, em HTML existem marcações (*Tags*) que indicam onde encontram-se os títulos, as palavras-chave e algumas outras estruturas importantes ao documento. O processo de indexação deve identificar estas marcações e indexar os termos presentes entre estas marcações com maior importância. Podem surgir alguns problemas, como o da *indexação incerta*, onde a pessoa encarregada de demarcar o documento não utiliza palavras que identificam corretamente o documento. Há alguns trabalhos que se destacam neste assunto: [YAT96] e [MOU92].

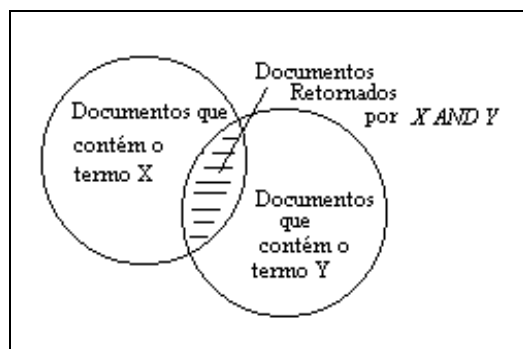


Figura - Resultado da consulta $X \text{ And } Y$

2. União - A união corresponde à localização dos documentos que contém ambas as palavras, independente do fato delas ocorrerem no mesmo documento. O operador de união é o *OR*, e é utilizado quando o usuário deseja documentos que pertençam a mais de um contexto, ou quando o usuário deseja abranger um número de documentos maior. Neste caso todos os documentos que possuem o termo *X* ou o termo *Y* são retornados.

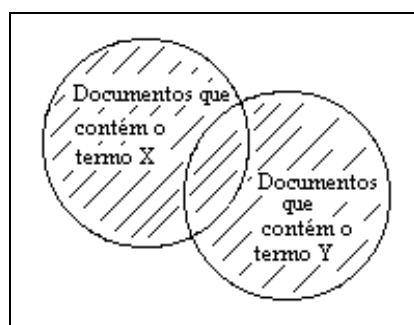


Figura - Resultado da consulta $X \text{ Or } Y$

3. Subtração - Corresponde ao operador *NOT*. Uma consulta do tipo $X \text{ Not } Y$, corresponde à localização de todos os documentos que contém *X*, com exceção àqueles que também contém *Y*.

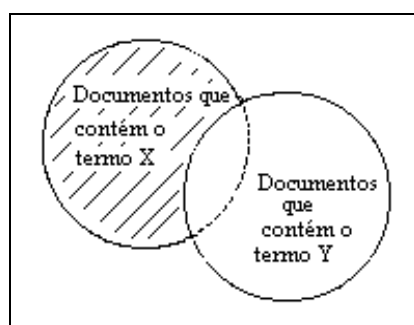


Figura - Resultado da Consulta $X \text{ Not } Y$

Os operadores podem ser utilizados em conjunto, a fim de restringir ainda mais os documentos que se deseja. São válidas portanto consultas do tipo *Programa AND (Computador OR Micro)*. Nestes casos é necessário haver uma maneira de se estabelecer a precedência dos operadores, pois há como decidir quais dos dois (ou mais) operadores devem ser executados primeiro, e a ordem de execução destes operadores influencia no resultado. Uma destas maneiras é a utilização dos *parênteses*, que são os indicadores de precedência universais (todos os programas aceitam). Caso não sejam utilizados estes indicadores, a

precedência é definida pelo sistema (e sistemas diferentes podem utilizar precedências diferentes).

Aparentemente a *pesquisa Booleana* apresenta resultados satisfatórios, pois permite que consultas complexas sejam realizadas. Apesar disso os usuários (mesmo os mais experientes) tem algumas dificuldades em utilizar este tipo de consulta (especialmente as mais complexas). Por consequência acabam não utilizando toda as facilidades que o método oferece.

Term Weighting

Apesar do índice utilizado até o momento ser capaz de indicar quais documentos possuem determinado termo, e a consulta *Booleana* conseguir recuperar estes documentos, o resultado destas consulta não indica o grau de relação dos documentos recuperados com a consulta formulada, ou seja, não há uma distinção entre os documentos. Havendo esta distinção, alguns documentos podem ser considerados mais importantes do que outros, mesmo que possuam os mesmos termos de consulta. De posse desta informação, é possível listar o resultado de uma consulta em ordem de *relevância*, ou seja, os documentos que aparecem nos primeiros lugares desta lista possuem um grau de relação maior com o assunto desejado pelo usuário do que os que aparecem nos últimos lugares.

Isso facilita o trabalho do usuário, pois ele não precisa analisar todos os documentos retornados para saber se servem ou não à sua necessidade. O usuário pode simplesmente analisar os primeiros, porque *estatisticamente* são mais importantes (relevantes) para ele.

Para que exista esta distinção entre os documentos, é necessário refinar o processo de indexação e de consulta de maneira que os termos possam indicar seu grau de importância nos documentos. É isto que algumas técnicas buscam fazer, atribuindo *pesos* ou *graus* de relação entre uma palavra e os documentos em que ela aparece. Estas técnicas partem do princípio de que havendo distinção entre os documentos, é possível obter uma performance melhor, já que os itens relevantes podem ser recuperados isoladamente, sem que os seus *vizinhos* de menor importância sejam recuperados.

Existem várias técnicas que buscam identificar o grau de relação entre um termo e um documento. Um estudo realizado por *Viles* [VIL95], indica que a maioria dos modelos de recuperação de informações utiliza estas técnicas. *Salton* cita algumas delas em seu livro [SAL83]. Neste trabalho somente a técnica mais simples, porém eficiente, é analisada. A técnica baseia-se na teoria de que as palavras que aparecem com maior frequência em um documento têm uma forte relação com seu conteúdo. A experiência indica também, que esta relação tende a diminuir quando este termo aparece em muitos documentos.

Portanto, a técnica consiste em identificar a frequência de determinada palavra em um documento (*Term Frequency*) e o número de documentos em que esta palavra aparece (*Inverse Document Frequency*). Com estas informações é possível atribuir um valor de relação entre esta palavra e o documento, e este valor é dado pela fórmula abaixo:

$$Peso_{td} = \frac{Freq_{td}}{DocFreq_t}$$

Onde $Peso_{td}$ é o grau de relação entre o termo t e o documento d ; a $Freq_{td}$ é o número de vezes que o termo t aparece no documento d ; e a $DocFreq_t$ representa o número de documentos que o termo t aparece.

Para cada termo do documento, é necessário calcular a sua relação utilizando-se a fórmula acima. Este peso é armazenado na lista invertida. Quando a consulta for requisitada pelo usuário, estes valores são utilizados no processo de identificação dos documentos relevantes a esta consulta.

Este processo de identificação da similaridade entre os termos e os documentos que utiliza algum método de relacionar as palavras com os documentos é conhecido como espaço de vetores (*Vector Space*). Agora cada documento possui um vetor com pares de elementos na forma {(palavra1, peso1), (palavra2, peso2), ... , (palavra n, peso n)}. Nestes pares, as *palavras* representam os termos utilizados na consulta, e o peso, seus respectivos valores de frequência no documento. Caso uma palavra não exista em um documento, seu valor de frequência é zero (0). Ao final, os pesos são somados, e os documentos listados por ordem decrescente de pesos.

Exemplo:

Termos da consulta:

Recuperação, Informações, Documentos

Índice:

<u>Palavra</u>	<u>Documentos em que aparece</u>		
<i>Documentos</i>	U(8)		
<i>Informações</i>	A(5)	H(2)	X(7)
<i>Recuperação</i>	A(3)	U(4)	X(6)

Espaço de vetores dos documentos:

- A: {(Recuperação, 3), (Informações, 5), (Documentos, 0)} Soma: 8
- H: {(Recuperação, 0), (Informações, 2), (Documentos, 0)} Soma: 2
- U: {(Recuperação, 4), (Informações, 0), (Documentos, 8)} Soma: 12
- X: {(Recuperação, 6), (Informações, 7), (Documentos, 0)} Soma: 13

Resultado da Consulta:

<u>Documento</u>	<u>Valor</u>
X	(13)
U	(12)
A	(8)
H	(2)

Logo, somando-se as frequências dos documentos retornados pelos termos da consulta, obtém-se um *rank* dos documentos mais importantes. *Salton e Buckley* [SAL87a] descrevem um método bastante conhecido (e melhor do que o citado acima) para o cálculo de relevância do documento à *consulta*, tornando melhores os resultados. É aconselhável estudar o trabalho de *Singhal* [SIG96], pois apresenta uma técnica mais recente.

Relevance Feedback

Após vários anos de profundo estudo na área de recuperação de informações, descobriu-se que o usuário não consegue recuperar os documentos de seu interesse na primeira tentativa de consulta ao sistema. Geralmente o que ocorre é que este usuário faz uma consulta tentativa, e vai refinando-a, alterando-a, de acordo com os resultados que obtém.

As consultas subsequentes passam a retornar cada vez mais documentos relevantes ao usuário, pois ele vai *contextualizando* melhor o assunto que deseja, utilizando novas palavras, e retirando as palavras que *desvirtuam* sua consulta recuperando documentos fora de seu interesse. Assim, são produzidas novas consultas teoricamente mais precisas e mais úteis (mais documentos relevantes são retornados).

Buckley [BUK95] define *Relevance Feedback* como sendo o processo automático de refinamento (alteração) de uma *query*, utilizando informações fornecidas pelo usuário sobre a relevância dos documentos previamente retornados (em uma consulta anterior).

Portanto, o *Relevance Feedback* (realimentação de relevâncias) é um processo de refinamento, que busca tornar a consulta do usuário mais precisa, recuperando somente os itens relevantes e descartando os irrelevantes.

Teoricamente existem vários processos de se fazer este refinamento. Vários artigos sobre a eficiência e o aperfeiçoamento da técnica - [MOR82], [ALL95], [BUK95] - e mostram que estes processos resumem-se em dois segmentos: a seleção dos termos (eliminando e adicionando termos), e a modificação dos pesos dos termos utilizados nas consultas (o que exige fórmulas matemáticas complexas).

A Fundamentação teórica desta técnica pode ser encontrada em [SAL83], [SAL87b].

Os métodos usuais, pedem ao usuário que selecione os documentos aparentemente relevantes, e então utilizam algumas palavras advindas destes documentos (dos títulos, por exemplo). Após, uma nova consulta que utiliza estes novos termos é submetida ao sistema. O processo pode repetir-se sucessivamente até que o usuário consiga atingir seu objetivo.

As últimas novidades da técnica podem ser observadas em [XU96].

Thesaurus

O *Thesaurus* vem a ser uma técnica de auxílio ao usuário. Pode ser entendido como um dicionário, pois ele ajuda o usuário a encontrar os melhores termos para a localização da informação que deseja. Geralmente é utilizado para que o usuário descubra quais são os termos utilizados pelo sistema, que identificam o assunto que ele deseja (fornece sinônimos).

Salton [SAL83] Define o *Thesaurus* como sendo uma estrutura que fornece grupos de palavras específicas a determinadas categorias. Ou seja, ao selecionar-se uma categoria, são informados os termos a ela relacionados. A estas categorias dá-se o nome de classes *Thesaurus*.

Chen [CHE96] realizou estudos com pesquisadores de ciências diferentes que necessitam trocar informações (na maioria das vezes os termos para um determinado objeto podem variar com a área). Notou que nestes casos particulares, onde pessoas necessitam realizar consultas em um Banco de Dados desconhecido (este é o mesmo caso do problema da *Busca Incerta* citado anteriormente), o *Thesaurus* mostra-se bastante útil, pois informa as diferentes palavras que identificam o mesmo objeto (assunto).

O estilo de apresentação de *Thesaurus* recomendado é o grafo. Cada nodo deste grafo representa um termo, que está ligado a outros termos (outros nodos do grafo). Estas ligações podem representar associações diferentes (vários tipos de relações), além de valores (pesos) diferentes. De posse do grafo, o usuário pode navegar por entre as palavras (como em um *Browser*). Após escolher a palavra de seu interesse, o *thesaurus* informa quais são as palavras que o usuário pode utilizar na consulta a fim de obter melhores resultados.

Os artigos mais interessantes sobre *Thesaurus* são os que tratam de sua construção (automática ou não) e as diversas utilizações que podem ter. Dentre estes artigos destacam-se os trabalhos de *Chen* [CHE94b], e o livro de *Salton* [SAL83].

Integrando Conceitos (BD Tradicional e BD Textual)

A integração entre um Banco de Dados Tradicional (BDT) (mais especificamente Relacional) e um Banco de Dados Textual é um desafio interessante. Isso porque os Banco de Dados Textuais não possuem a maturidade de um banco de dados tradicional em relação a alguns aspectos, como por exemplo o controle de concorrência, decorrentes das propriedades ACID⁹. Estes aspectos presentes em um BDT, devem ser estendidos para que as propriedades possam ser aplicadas em estruturas como a de documentos. Segundo *DeFazio* [DEF95], os estudos na área de integração indicam que um SGBD (Sistema de Gerenciamento de Banco de Dados) deve:

⁹ ACID: Propriedades de Atomicidade, Consistência, Isolamento e Durabilidade.

- Suportar armazenamento, indexação, recuperação e modificação de documentos;
- Semânticas de transação que possuam as propriedades ACID;
- Extensões de linguagem de consulta que permitam a seleção de documentos relevantes (em forma de *ranking*).

Enquanto um BDT ainda não consegue oferecer estas características, o processo de integração é *caro*, mas não impossível. Vários trabalhos buscam fazer esta integração. O trabalho de *DeFazio* [DEF95] é um dos mais atuais, e utiliza uma estrutura conceitual similar à apresentada no diagrama abaixo:

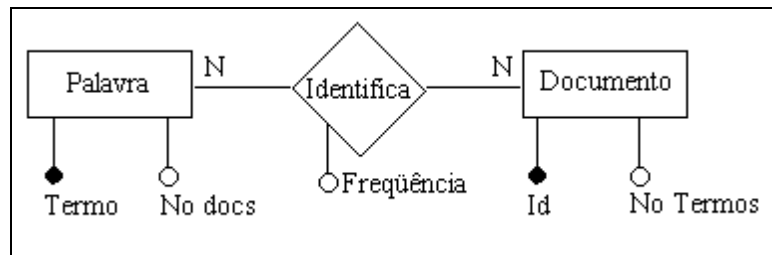


Figura - ER de um BD Textual

São criadas duas tabelas: uma para os documentos e outra para os termos. Esta última vai servir de índice para os documentos (como uma lista invertida). *DeFazio* chama esta técnica de Indexação cooperativa, já que já que o Banco de Dados já oferece uma estrutura de índice (baseada na chave-primária) e a tabela *Palavra* também representa um índice do tipo *Lista Invertida*.

Para que o sistema funcione, são necessários alguns módulos (funções) extras, que possibilitem a cooperação entre os índices. Estes módulos adicionais são como extensões ao banco de dados, e para que isto fosse possível, *DeFazio* utilizou uma versão especial do *Oracle*, o *RDB*.

As tarefas dos módulos adicionais resumem-se na criação dos índices específicos para os documentos e a recuperação por ordem de relevância. Ou seja, a inclusão de documentos deve também realizar a Indexação automática de palavras, e as *Queries* devem levar em conta as frequências das palavras, a fim de montar o *Ranking* com os documentos relevantes. Esta última tarefa é particularmente interessante, pois o algoritmo elaborado para a consulta *booleana*, primeiramente recupera as informações sobre os documentos utilizando *queries* comuns em *SQL*, e após aplica os cálculos de frequência e seleção sobre o conjunto retornado.

Esse tipo de extensão construído não é o mais adequado, pois o ideal é modificar a estrutura interna do BD (para que aceite índices especiais em caso de dados especiais) e sua linguagem de consulta para que estas características sejam incorporadas.

Os testes de *DeFazio* mostram que a utilização de índices em forma de tabela funciona perfeitamente, mas a performance do sistema cai. Não é possível realizar a compactação de índice e de documentos geralmente utilizada em sistemas de recuperação de Informações. Além disso, os BDT estão preparados para otimizar operações em cima de tuplas (geralmente informações pequenas), o que não é o caso de grandes bancos de informações textuais.

Conclusão

Apesar de existir a muitos anos, a área de Recuperação de Informações ainda é precária em alguns aspectos, ainda mais agora, com a expansão da Internet que possui um acúmulo grandioso de informações. As maiores preocupações giram em torno da preocupação de retornar documentos realmente úteis ao usuário, fazendo com que ele não perca tempo com informações irrelevantes.

Muitos cientistas estão preocupados ultimamente em elaborar baterias de testes que realmente comprovem que determinada técnica vai atender às necessidades do usuário. Testes que levam em conta o tipo de consulta realizada por usuários experientes e usuários primários de sistemas de IR. Buscam avaliar o retorno do sistema em relação estas consultas, e então decidir se este sistema é ou não eficiente.

Outro fator decisivo em um sistema é a facilidade de interação com o usuário, pois a maior dificuldade de um sistema está em descobrir como ele funciona e portanto em descobrir como ele pode dispor a informação que realmente interessa.

Resumindo, um sistema de recuperação de informações deve oferecer as seguintes características:

- Ferramentas de auxílio, que facilitem ao usuário identificar o melhor vocabulário que represente a informação que deseja (Thesaurus, dicionários de sinônimos...);
- Diversos caminhos para uma mesma informação, permitindo que o usuário possa encontrar o que deseja por meios diferentes;
- Inteligência a fim de identificar a intenção do usuário mais rapidamente e de forma eficaz;
- Interface amigável, com características gráficas e também com suporte a Linguagem Natural;
- Propriedades ACID, permitindo a concorrência, processamento paralelo e distribuição da informação;
- Otimização de recursos, com ferramentas de compactação da base de dados e do índice, a fim de que mais informações possam ser armazenadas.
- Facilidade de manutenção dos dados, permitindo modificações e inclusões on-line de informações, independente da forma com que esta se apresenta.
- Desempenho razoável, recuperando informações úteis o mais breve possível;

Bibliografia

- [ACM96a] ACM - Association for Computing Machinery, Inc. In Memoriam: Gerard Salton. **ACM Transactions on Information Systems**. v.14. n.1. Janeiro, 1996.
- [ACM96b] ACM - Association for Computing Machinery, Inc. **Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. 18-22 Agosto. 1996. Zurich, Switzerland.
- [ALL96] ALLAN, James. **Relevance Feedback with Too Much Data**. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.
- [BUK95] BUCKLEY, Chris; SALTON, Gerard. **Optimization of Relevance Feedback Weights**. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.
- [CHE96] CHEN, H. at alli. **A concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: An Experiment on the Worm Community System**. MIS Department. University of Arizona. 2, Julho. 1996. (<http://ai.bpa.arizona.edu/papers/>).
- [CHE94a] CHEN, H. **The Vocabulary Problem in Collaboration**. MIS Department. University of Arizona. 5, Julho. 1994. (<http://ai.bpa.arizona.edu/papers/>).
- [CHE94b] CHEN, H. at alli. **Generating a Domain-specific Thesaurus Automatically: Na Experiment on flyBase**. MIS Department. University of Arizona. 1994. (<http://ai.bpa.arizona.edu/papers/>).
- [CHE94c] CHEN, H. at alli. **A Textual Database/Knowledge-Base Coupling Approach to Creating Computer-Supported Organizational Memory**. MIS Department. University of Arizona. 5, Julho. 1994. (<http://ai.bpa.arizona.edu/papers/>).
- [CHU95] CHURCH, Kenneth W. **One Term or Two?**. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.
- [CRO82] CROFT, W. B; RUGGLES, L. **The Implementation of a Document Retrieval System**. Proceedings of Conference on Research and Development in Information Retrieval. Maio.1982. Berlin.

- [DEF95] DEFAZIO et alli. **Integrating IR and RDBMS using cooperative indexing.** Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.
- [FOX95] FOX, Edward A; INGWERSEN; FIDEL, Raya. **Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.** Seattle, Washington. USA. 9-13, Julho.1995.
- [KRA96] KRAAIJ, Wessel. **Viewing Stemming as Recall Enhancement.** Proceedings on 19th ACM International SIGIR Conference on Research and Development in Information Retrieval. 18-22 Agosto. 1996. Zurich, Switzeland.
- [KOR94] KORTH, H. F.; SILBERSCHATZ, A. **Sistemas de Bancos de Dados.** 2^a edição. Makron Books, MacGraw-Hill. 1994.
- [MOR82] MORRISSEY, Joan. **Na Intelligent Terminal for Implementing Relevance Feedback on Large Operational Retrieval Systems.** Proceedings of Conference on Research and Development in Information Retrieval. Maio. 1982. Berlim..
- [MOU92] MOULIN, Bernard; ROUSSEAU, Daniel. **Automated knowledge acquisition from regulatory texts.** IEEE Expert. Outubro, 1992.
- [RIL95] RILOFF, Ellen. **Little Words Can Make a Big Difference for Text Classifications.** Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.
- [SAL88] SALTON, Gerard; SMITH, Maria. **On the Application of Sintatic Methodologies in Automatic Text Analysis.** Technical Report. Department of Computer Science. Cornell University. Ithaca, New York. 1988.
- [SAL87a] SALTON, Gerard; BUCKLEY, Chris. **Term Weighting Approaches in automatic Text Retrieval.** Technical Report. Department of Computer Science. Cornell University. Ithaca, New York. 1987.
- [SAL87b] SALTON, Gerard; BUCKLEY, Chris. **Improving Retrieval Performance by Relevance Feedback.** Technical Report. Department of Computer Science. Cornell University. Ithaca, New York. 1987.
- [SAL83] SALTON, Gerard. **Introduction to Moder Information Retrieval.** McGraw-Hill. 1983.
- [VIL95] VILES, Charles L; FRENCH, James C. **Dissemination of Collection Wide Information in a Distributed Information Retrieval System.** Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington. USA. 9-13, Julho.1995.

- [WIV96a] WIVES, Leandro K.; SARDI; Filipe L. M.; LOH, Stanley. **Definição de uma ferramenta de busca em bases de dados textuais usando um Hiperdicionário.** Anais. III Jornadas de Informatica e Investigacion Operativa, VI Encuentro del Laboratorio de Ciencia de la Computacion. Montevideo, Uruguai, 12-14 de Dezembro de 1996.
- [WIV96b] WIVES, Leandro K. **Um Modelo de Hiperdicionário: Estudo de Caso em Prontuários Médicos.** Curso de Graduação em Ciência da Computação - UCPEL. Dezembro de 1996. Trabalho de Conclusão.
- [XU96] XU, Jinxi; CROFT, W. Bruce. **Query Expansion Using Local and Global Document Analysis.** Proceedings on 19th ACM International SIGIR Conference on Research and Development in Information Retrieval. 18-22 Agosto. 1996. Zurich, Switzeland.
- [YAT96] YATES, R. B. *An extended model for full text databases.* **Journal of the Brazilian Computer Society**, v.2, n.3, Abr 1996.