KARINA SILVEIRA SANTOS

DEPENDÊNCIA ENTRE TERMOS NO MODELO VETORIAL

Dissertação apresentada ao programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia, como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Área de concentração: Banco de dados.

Orientador: Professor Dr. João Nunes de Souza
Co-orientador: Professor Dr. Ilmério Reis da Silva

UBERLÂNDIA – MG

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

2003

Universidade Federal de Uberlândia

Programa de Pós-Graduação em Ciência da Computação

Os abaixo assinados, por meio deste, certificam que leram e recomendam a aceitação da dissertação intitulada 'Dependência entre termos no Modelo Vetorial' por Karina Silveira Santos como parte dos requisitos exigidos para a obtenção do título de mestre em Ciência da Computação.

Uberlândia, 17 de Dezembro de 2003.

Banca Examinadora:

Prof. João Nunes de Souza – Orientador - UFU

Prof. Ilmério Reis da Silva – Co-orientador- UFU

Prof.ª Denise Guliato - UFU

Prof. Nivio Ziviani - UFMG

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Data: Dezembro 2003

Autora: **Karina Silveira Santos**

Título: **Dependência entre termos no modelo vetorial**

Faculdade: **Faculdade de Computação**

Grau: **Mestre** Convocação: **Dezembro** Ano: **2003**

Karina Silveira Santos

Autora

À minha família, pelo apoio, incentivo e carinho incondicional

AGRADECIMENTOS

A Deus, por sua infinita bondade e amor em me propiciar esta oportunidade e a força necessária para concluir este trabalho.

A minha família, pelo amor, carinho e encorajamento que sempre me deram em todos os momentos da minha vida. Agradeço, especialmente, aos meus pais, José Amado e Nedina, que sempre dedicaram suas vidas aos filhos, ensinando a importância de uma educação de qualidade. Ao meu irmão Fernando, por me apoiar nos momentos difíceis.

Aos meus orientadores, João Nunes e Ilmério, que foram essenciais para o desenvolvimento deste projeto. Agradeço a dedicação, a paciência, os conselhos e incentivos na orientação deste trabalho.

Ao meu namorado Vinícius, por tantos bons momentos compartilhados, cujo amor, contribuição, incentivo e compreensão foram preciosos para a concretização deste trabalho.

A todos os colegas e amigos do programa de pós-graduação pela boa vontade em ajudar, em especial, aos membros do grupo de recuperação de informação por todas as suas contribuições ao longo destes anos. Agradeço ainda os funcionários da Faculdade de Computação sempre solícitos a nos atender.

Aos meus amigos do coração que torceram comigo e acreditaram na conclusão deste trabalho.

RESUMO

O número de informações eletrônicas disponíveis para acesso nas bibliotecas digitais e na *Web* vem crescendo em ritmo acelerado. Em decorrência disto, a tarefa de encontrar informação útil torna-se difícil. Melhorar essa situação requer avanços no projeto e implementação de sistemas de recuperação de informação, dentre elas, algoritmos de *ranking*. O Modelo Vetorial é uma abordagem que vem sendo utilizada ao longo dos anos para prover tal ordenação. Neste modelo, cada termo do índice corresponde a um vetor, e esses vetores, em conjunto, geram a base do espaço vetorial de interesse. Nesta base, os vetores são ortogonais entre si, indicando que os respectivos termos são mutuamente independentes. Entretanto, esta é uma simplificação que não corresponde à realidade. Diante desse cenário, apresentamos, neste trabalho, uma extensão ao Modelo Vetorial para contemplar a correlação entre os termos. No modelo proposto, os vetores de termos, originalmente ortogonais, são rotados no espaço refletindo geometricamente a semântica de dependência entre os termos. Essa rotação pode ser feita com base em técnicas que resultem em informações sobre o relacionamento entre termos da coleção. Propomos as técnicas regras de associação e a geração de termos lexicograficamente semelhantes. A geração de regras de associação é uma conhecida técnica da mineração de dados. Ela é utilizada na recuperação de informação para encontrar conjuntos de termos que co-ocorrem na coleção de documentos. A técnica de obtenção de termos lexicograficamente semelhantes é uma estratégia semelhante à extração de radicais. A eficácia de recuperação do modelo proposto é avaliada para as duas técnicas, empregando as medidas de Precisão e Revocação. Os resultados mostram que há um aumento na efetividade de recuperação do modelo proposto em comparação ao Modelo Vetorial clássico para todas as coleções de referência avaliadas, obtendo um ganho de até 31% na média da precisão.

ABSTRACT

The number of available electronic information for access in digital libraries and Web is growing fast. An immediate consequence is that the task of finding useful information becomes difficult. Improving upon this situation requires progresses in the project and implementation of information retrieval systems, among them, ranking algorithms. The Vector Space Model is an approach, which has been used along the years to provide such ranking. In this model, each index term corresponds to a vector, and these vectors, together, generate the basis of the vector space of interest. In this basis, the vectors are pairwaise orthogonal, indicating that the corresponding terms are mutually independent. However, this simplification does not correspond to the reality. Then, we present, in this work, an extension to the Vector Model to take into account the correlation between terms. In the proposed model, term vectors, originally orthogonal, are rotated in space geometrically reflecting the dependence semantics among terms. This rotation is done with any technique that generates information on the relationship among terms of the collection. We propose two techniques, named, association rules and the generation of terms lexicographically similar. The generation of association rules is a known data mining technique. It is used in the information retrieval to find sets of terms that co-occur in documents collection. The technique of obtaining terms lexicographically similar creatures is a strategy similar to the extraction of radicals. The retrieval effectiveness of the proposed model is evaluated for the two techniques using the measures of precision and recall. The results shows that our model improves in average precision, relative to the standard Vector Model, for all collections evaluated, leading to a gain up to 31%.

Índice

Lista de Figuras

Lista de Tabelas

1

INTRODUÇÃO

O número de informações eletrônicas disponíveis vem crescendo em ritmo acelerado desde o surgimento da *Web* em meados dos anos 1980 [40] e do crescente avanço na moderna tecnologia computacional. Três fatores contribuíram decisivamente para esse processo. Primeiro, o baixo custo para ter acesso a várias fontes de informação. Segundo, os avanços em todo tipo de comunicação digital possibilitaram grande acesso às redes. Isto resulta que a fonte de informação está disponível mesmo se localizada em lugares distantes, e o acesso pode ser feito de forma rápida. Terceiro, a liberdade em compartilhar qualquer tipo de informação que alguém julgue interessante contribui fortemente para a popularidade da *Web*.

A dinâmica, a abundância e a heterogeneidade das informações que compõem a *Web* e as modernas bibliotecas digitais trazem consigo novos desafios relacionados à obtenção dessas informações. A tarefa de encontrar documentos relevantes, entre os disponíveis, quase sempre, é entediante e difícil.

Diante desse cenário, surgiram alternativas para auxiliar o usuário a encontrar informações relevantes em grandes repositórios de informações, dentre elas, podemos citar os sistemas de recuperação de informação para a *Web*, também chamados de engenhos de busca.

Este capítulo descreve o contexto no qual este trabalho está inserido, apresentando a motivação e o problema a serem tratados. Em seguida, serão discutidos o objetivo do trabalho, os trabalhos relacionados e a organização dos capítulos seguintes.

1.1 Descrição do Problema

Sistemas de recuperação de informação tradicionais, usualmente, adotam índices de termos para indexar e recuperar documentos [5]. Em um senso restrito, um termo é uma palavra-chave que contém algum significado próprio (geralmente um substantivo). De uma forma mais geral, um termo é simplesmente qualquer palavra que aparece no texto de um documento da coleção. Recuperações baseadas em palavras-chave são simples, porém geram questionamentos relativos à tarefa de recuperação de informação. Por exemplo, recuperações usando palavras-chave adotam como principal fundamento a idéia de que a semântica dos documentos e a necessidade de informação do usuário podem ser naturalmente expressas por conjuntos de palavras-chave. Claramente, isto é uma considerável simplificação do problema, uma vez que grande parte da semântica em um documento ou consulta do usuário é perdida quando seu texto é substituído por um conjunto de palavras. Além disso, o casamento entre cada documento e a consulta do usuário é feito nesse espaço impreciso de palavras-chave. Assim, não é surpresa que, em reposta a uma consulta do usuário expressa como um conjunto de palavras-chave, encontremos documentos recuperados irrelevantes para o usuário.

Um documento pode ou não ser relevante para uma consulta do usuário, dependendo de muitas variáveis referentes ao documento (o assunto, a organização, a clareza etc), assim como outras inúmeras características relacionadas ao usuário (o motivo da procura, conhecimento prévio, o usuário saber ou não o que quer etc). Visto que relevância depende, de uma maneira complexa, de vários fatores, é reconhecido que um sistema de recuperação de informação não consegue precisamente selecionar somente documentos relevantes. Então, é sugerido que um sistema de recuperação disponha os documentos em ordem de relevância potencial para a consulta do usuário [38].

Uma abordagem que vem sendo utilizada, ao longo dos anos, para prover tal ordenação, delineia documentos e consultas como vetores, sendo denominado Modelo Vetorial [27, 29, 30]. As palavras-chave utilizadas para descrever o conteúdo dos documentos ou consultas correspondem aos vários elementos dos vetores. Nesse

modelo, se o vocabulário indexado consiste de *n* palavras-chave distintas, cada documento é um vetor *n*-dimensional, no qual o *i*-ésimo elemento representa a importância da *i*-ésima palavra-chave para o documento em questão. Quando uma consulta é apresentada, o sistema formula o vetor da consulta e faz o casamento com os documentos, baseado em um método de determinação da similaridade entre vetores.

No Modelo Vetorial, cada termo do índice corresponde a um vetor, e estes vetores, em conjunto, geram o espaço vetorial de interesse. O efeito disso é que qualquer documento da coleção pode ser expresso como uma combinação linear desses vetores de termos. Similarmente, quando uma consulta é apresentada, um vetor também é construído com base em vetores de termos.

Os vetores de termos que geram o espaço vetorial de interesse são ortogonais entre si, porque não é conhecida a priori a correlação entre estes vetores. Isto significa que os termos da coleção são igualmente independentes uns dos outros. Nessa representação, as palavras "rede" e "computadores", por exemplo, não têm conexão. Da mesma forma, as palavras "algoritmo" e "algoritmos" não têm relação entre si. Os termos são independentes lingüística e estatisticamente. Independência estatística indica que a ocorrência de um termo não está relacionada com a ocorrência de outro. Independência lingüística significa que a interpretação de um termo não afeta a interpretação de outro.

Dessa forma, para o usuário obter documentos relevantes recuperados, em um sistema de recuperação de informação que utiliza o Modelo Vetorial clássico, em geral, é recomendado especificar uma consulta longa, contendo várias palavras que expressem a sua necessidade. Esse modelo retorna somente documentos que contenham pelo menos uma das palavras especificadas na consulta.

1.2 Trabalho Proposto

Tendo como motivação o cenário descrito anteriormente, o principal objetivo deste trabalho é adicionar informações de dependência entre os termos da coleção no Modelo Vetorial. Algumas iniciativas que acrescentam dependência entre termos já

foram apresentadas na literatura e são descritos na seção 1.3. As mudanças no modelo têm como principal objetivo não comprometer a simplicidade do modelo vetorial original.

No Modelo Vetorial clássico, os termos da coleção são representados por vetores de termos ortogonais entre si, assumindo que os termos não possuem qualquer relação entre si. O algoritmo proposto neste trabalho tem como principal fundamento a rotação dos vetores de termos no espaço, de forma que suas representações reflitam a dependência entre os termos.

Todos os vetores de termos que possuem algum tipo de dependência com um ou mais termos são rotados no espaço. Após todas as rotações, os vetores de termos não são necessariamente ortogonais entre si. No conjunto de vetores resultantes, a proximidade entre os vetores está relacionada com o grau de dependência entre os respectivos termos. Quanto mais próximos os vetores de termos, maior dependência observada entre si.

A principal conseqüência dessa rotação dos vetores de termos, no cálculo da similaridade entre a consulta e os documentos, é a expansão automática da consulta. A consulta é expandida com termos relacionados aos seus termos originais. Além disso, documentos que possuem os termos da consulta mais os termos relacionados a eles ocupam uma posição no *ranking* acima dos documentos que possuem apenas os termos da consulta.

Duas abordagens para a definição de dependência entre os termos são mostradas: as regras de associação e os termos lexicograficamente semelhantes. Estas duas abordagens são aplicadas ao algoritmo proposto e implementadas para suas validações.

1.3 Trabalhos Relacionados

Várias abordagens para a incorporação de dependência entre os termos já foram apresentadas na literatura, especialmente utilizando a co-ocorrência de termos

na coleção de documentos. Destacamos algumas delas e detalhamos aqui os trabalhos que mais se aproximam do proposto nesta dissertação.

A expansão da consulta no Modelo Vetorial é sugerida em várias propostas, dentre elas, [11,20,22,34]. Em [34], Voorhees examinou a utilidade da expansão de consulta léxica na coleção TREC. Todos os termos da consulta foram expandidos empregando um *thesaurus* léxico. Voorhees obteve ganho considerável de efetividade apenas no uso de consultas curtas. Mandala et al. [20] analisaram as características de diferentes tipos de *thesaurus* e propuseram um método para combiná-los e expandir consultas. Em [11], Nie e Jin, empregaram o operador lógico OR para conectar termos de expansão com os termos originais da consulta. Outra técnica para expansão de consultas é a utilização de *feedback* do usuário, como fizeram Buckley et al. [22].

Em [7], Becker e Kuropka expõem um modelo de recuperação de informação para a comparação de documentos que representa tópicos, termos e documentos como vetores. A base do espaço é formada por um conjunto de vetores de tópicos ortogonais entre si. Os termos são representados nesse espaço de tópicos de forma que a proximidade entre eles reflita a correlação entre os termos. Além disso, vetores de termos apontam para a mesma direção dos tópicos nos quais estão relacionados. Cada termo tem um peso associado, que representa sua ligação com um tópico geral. O ângulo entre os vetores de termos e o peso do termo é calculado utilizando informações sobre a coleção. Por exemplo, é citado, em [7], que dada uma lista de radicais dos termos da coleção, o ângulo entre dois termos que tenham o mesmo radical deve ser zero grau, e o peso do termo é 1. Por outro lado, se há uma lista de *stop-words*, o ângulo entre uma *stop-word* e qualquer outro termo deve ser de noventa graus e o peso 0. Este modelo é flexível no sentido da especificação da similaridade entre os termos, porém exige muitas informações sobre a coleção de termos na construção dos vetores. Em [7], não são apresentadas as formas de obtenção dos vetores tópicos e nem os resultados relacionados à eficácia de recuperação.

O modelo proposto neste trabalho difere do apresentado em [7] nos aspectos descritos a seguir. Em nossa abordagem, a base do espaço vetorial é formada por vetores de termos e não de tópicos. Utilizamos a definição de peso dos termos *tf-idf*. Além disso, em [7] não é apresentado um método automático de obtenção da

correlação entre os termos incorporando essas informações no modelo como é feito no trabalho aqui exposto.

Um trabalho semelhante ao proposto aqui foi realizado por Possas et al. em [23,24,25]. Uma extensão ao Modelo Vetorial clássico foi sugerida considerando a correlação entre os termos, obtida empregando regras de associação. Em [25], é apresentado um novo modelo, denominado *set-based model*, para calcular pesos dos termos baseado na teoria de conjuntos e ordenar documentos. Para o cálculo desses pesos, a teoria das regras de associação é utilizada. A partir do vocabulário da coleção, são gerados os *closed termsets* definidos em [42], que são conjuntos de termos que ocorrem simultaneamente na coleção de documentos. Estes *termsets* substituem os vetores de termos e formam a base do espaço vetorial, permitindo a representação de documentos e consultas como combinações lineares de seus vetores. A métrica para o cálculo da similaridade é o produto interno normalizado entre os vetores de documentos e consultas, a mesma utilizada no Modelo Vetorial clássico. Ao representar vetores de documentos e consultas usando o conjunto de *termsets*, o modelo privilegia documentos que contenham maior número de *termsets* da consulta, ou seja, documentos em que a co-ocorrência entre os termos da consulta seja alta. A proposta apresentada pelos autores em [24] é semelhante à descrita, sendo que a principal diferença consiste no uso de *termsets* freqüentes e não *closed termsets*. Já em [23], uma extensão ao *set-based model* é proposta valendo-se de informações sobre proximidade entre os termos da consulta nos documentos.

O trabalho descrito acima é semelhante ao apresentado nesta pesquisa, porque ambos utilizam as regras de associações para modificar o Modelo Vetorial clássico. Entretanto, os dois modelos diferem em alguns aspectos. Em [25], os *termsets* são gerados e formam a base do espaço, enquanto no modelo aqui proposto, os *termsets* são gerados para aproximar os vetores de termos que continuam a formar a base do espaço. Além disso, ao representar o vetor de consultas na nova base, é feita a expansão automática da consulta com termos relacionados aos termos da consulta, o que não ocorre em [25].

O Modelo Vetorial Generalizado (MVG) é uma outra abordagem de Modelagem Vetorial, que contempla a correlação entre termos [38,39]. Nesse modelo,

assim como no exposto aqui, os vetores de termos são modificados com o objetivo de refletir a dependência entre os termos da coleção. No MVG, os termos podem ser não-ortogonais e são representados por componentes denominados *mintermos*. Os *mintermos* são vetores, com pesos binários, que indicam todas as possibilidades de co-ocorrência de termos em documentos. A base para o MVG é formada por um conjunto de 2^t (t é o número de termos distintos da coleção) vetores de *mintermos*. Os vetores $\mathbf{k}_i$ são determinados utilizando os vetores de *mintermos* e a frequência do termo no documento. Assim, $\mathbf{k}_i \in \Re^{2^t}$. Os vetores de documentos e consultas e a similaridade entre ambos são obtidos empregando os mesmos cálculos do Modelo Vetorial clássico. O modelo é detalhado na seção 2.3.2. O MVG é estendido em [37] para lidar com situações em que a consulta é especificada utilizando expressões booleanas.

O modelo aqui proposto tem algumas semelhanças com o MVG. Ambos modificam o Modelo Vetorial clássico para contemplar a dependência entre termos e fazem expansão automática da consulta. Ao considerar, nesse modelo, a técnica de dependência entre termos regras de associação, os dois modelos fazem uso de informações de co-ocorrência dos termos nos documentos da coleção. Porém, no MVG, para calcular a co-ocorrência, é utilizada a freqüência enquanto no modelo apresentado aqui, além da freqüência é usada a confiança da regra de associação, que indica a certeza da regra descoberta. Enquanto o modelo proposto permite ajustar os parâmetros de aproximação entre os vetores de termos, no MVG, não há esta flexibilidade.

Neste trabalho, a incorporação das associações entre os termos é clara, porque é feita caso a caso, ou seja, termo a termo, alterando a base do espaço vetorial. Já no MVG, todas as associações entre termos são adicionados ao modelo, a incoporporação é feita de forma generalizada, sem ajustes. Além disso, neste último, a dimensão dos vetores de termos é exponencial, depende do número de termos da coleção, elevando o custo computacional.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte maneira:

O Capítulo 2 mostra os principais conceitos relacionados aos sistemas de recuperação de informação, a saber: técnicas de representação dos dados, os modelos clássicos e os principais métodos de avaliação dos sistemas.

O Capítulo 3 expõe as regras de associação como ferramenta para a mineração de termos que co-ocorrem na coleção de documentos, bem como o algoritmo para a geração dessas regras.

No Capítulo 4, apresentamos a metodologia proposta para contemplar a dependência entre os termos no Modelo Vetorial. Descrevemos também duas técnicas de geração de dependência entre os termos de uma coleção de documentos: as regras de associação e os termos semelhantes lexicograficamente.

No Capítulo 5, mostramos e discutimos os resultados da execução dos métodos aqui propostos, bem como do Modelo Vetorial clássico.

Finalmente, o Capítulo 6 conclui o trabalho e aponta direções futuras.

2

FUNDAMENTOS DE RECUPERAÇÃO DE INFORMAÇÃO

A primeira solução sistemática para o problema de recuperação de informação em uma grande coleção de dados foi desenvolvida há cerca de 4000 anos, por bibliotecários, que passaram a organizar os livros catalogando-os por autor e título [40]. A solução seguinte surgiu no século XVI, com a construção dos índices. Os índices correspondiam a listas de palavras-chave, que apontavam para documentos. Um problema inerente a esta estrutura consistia em selecionar as palavras "apropriadas" para serem associadas a um documento.

Com o advento dos computadores, a partir da década de 1950, surgiram os índices, que armazenavam todos os termos pertencentes a um documento. Esses índices solucionaram o problema descrito acima, porém, outro problema emergiu: para a maioria das consultas, muitos documentos eram retornados, e poucos deles eram relevantes.

Os sistemas de recuperação de informação são projetados com o objetivo de proverem o usuário, em resposta a uma consulta, referências a documentos que contenham a informação desejada pelo usuário. Uma aplicação computacional típica de um sistema de recuperação de informação é em um ambiente de biblioteca, em que o banco de dados consiste de livros, jornais, revistas etc.

Claramente, um problema central relativo a sistemas de recuperação de informação é a questão de determinar quais documentos são relevantes para a consulta do usuário e quais não o são. Tal decisão é, geralmente, dependente de um algoritmo de ordenação, que tenta estabelecer uma ordem de precedência dos documentos

recuperados. Diferentes estratégias de algoritmos geram diferentes modelos de recuperação de informação. O modelo de recuperação de informação adotado determina as previsões do que é relevante e o que não é.

Neste capítulo, são abordados os conceitos básicos, técnicas e modelos relacionados à recuperação de informação (RI).

2.1 Processo de Recuperação de Informação

A arquitetura de um sistema de recuperação de informação textual é representada na Figura 2.1.

Um dos elementos dessa arquitetura é a base de dados textual. Esta base engloba: (i) os documentos a serem utilizados pelo sistema de RI; (ii) as operações a serem realizadas sobre o conteúdo textual desses documentos; (iii) e a visão lógica, que consiste na estrutura que armazenará o texto e nos elementos que poderão ser recuperados por meio dessa estrutura [5].

As operações sobre o texto transformam os documentos iniciais em suas correspondentes visões lógicas (passos 1 e 2, Figura 2.1). Com base na visão lógica é construído um índice para o texto (passo 3, Figura 2.1). O índice corresponde a uma estrutura de dados importante para um sistema de RI, visto que permite uma busca rápida sobre um grande volume de dados.

Existem várias estruturas de índice, porém a mais empregada é a estrutura de arquivos invertidos, que será detalhada na Seção 2.2.1.

A construção e a manutenção dos arquivos invertidos corresponde a um processo bastante custoso no que diz respeito ao recurso tempo. Porém esses custos são amortizados, à medida que as consultas são submetidas ao sistema.

Após a construção de um índice para a base textual, temos o início do processo de recuperação de informação. Este se inicia quando o usuário submete uma consulta ao sistema (passo 4, Figura 2.1). São realizadas operações sobre a consulta, que irão transformá-la numa representação inteligível ao sistema (passo 5, Figura 2.1). Essa

consulta é processada (passos 6 e 7, Figura 2.1) e um conjunto de documentos é retornado em resposta a esse processamento (passo 8, Figura 2.1).

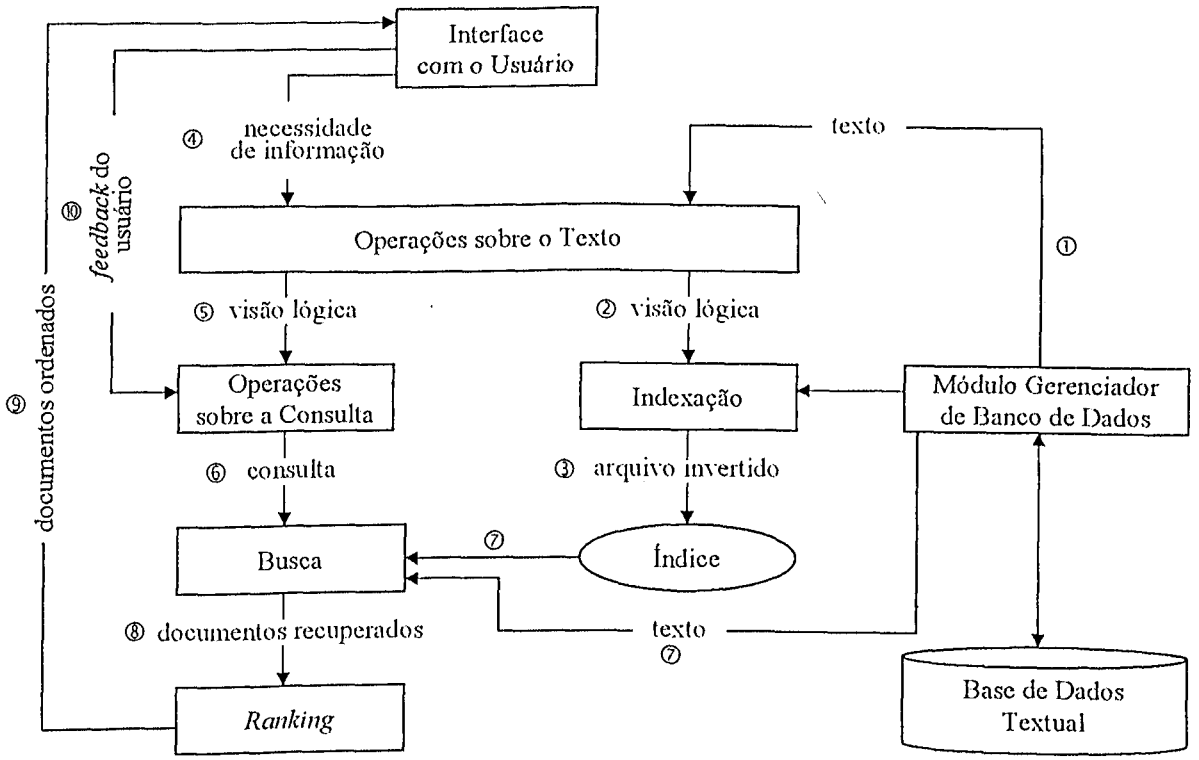

Figura 2.1 - Processo de Recuperação de Informação – adaptado de [5].

Antes de serem retornados ao usuário, esses documentos são ordenados de acordo com um método de relevância (passo 9, Figura 2.1). Existem várias medidas que objetivam obter um valor de relevância para um documento. Na Seção 2.4, serão abordadas algumas delas.

Após os documentos terem sido retornados ao usuário, este examina o conjunto de documentos à procura de informações que atendam a sua necessidade de informação. Nesse momento, o usuário pode dar início a um ciclo de *feedback* (passo 10, Figura 2.1), em que, o sistema utiliza os documentos selecionados pelo usuário como relevantes para alterar a consulta formulada inicialmente e submetê-la novamente ao sistema.

2.2 Representação dos Dados

Um dos principais desafios inerentes ao processo de recuperação de informação consiste em acessar de forma eficiente o conteúdo dos documentos, bem como a relação existente entre eles. Estruturas de dados capazes de representar os documentos e localizá-los de forma eficiente são essenciais para o sucesso de um sistema de recuperação de informação.

A seguir, detalhamos a estrutura de acesso denominada arquivo invertido - a mais comumente utilizada tanto por sistemas de recuperação de informação, quanto por sistemas de banco de dados. Outras estruturas de acesso existem, como a *Suffix Trees* [5] que é uma estrutura construída sobre todos os sufixos do texto, porém descrevemos apenas os arquivos invertidos.

2.2.1 Arquivos Invertidos

Os arquivos invertidos são estruturas de dados que permitem encontrar de forma rápida quais documentos de uma coleção possuem um dado termo. Essa estrutura é composta por dois elementos: (i) o vocabulário - que consiste no conjunto de todas as palavras diferentes que ocorrem na coleção; (ii) e as listas invertidas - que são listas contendo todos os documentos da coleção nos quais surge um dado termo. A primeira tabela da Figura 2.2 ilustra uma coleção de documentos e os termos que aparecem nos respectivos documentos. A segunda tabela da Figura 2.2 mostra o arquivo invertido correspondente à coleção, sendo que, para cada termo, há uma lista de documentos no qual o termo aparece e sua respectiva freqüência.

As listas invertidas podem conter, além dos documentos em que ocorre o termo, outras informações que sejam úteis ao engenho de busca, como por exemplo, a freqüência do termo no documento. Esse valor é chamado na literatura [5, 18] de *Term-Frequency* (*tf*).

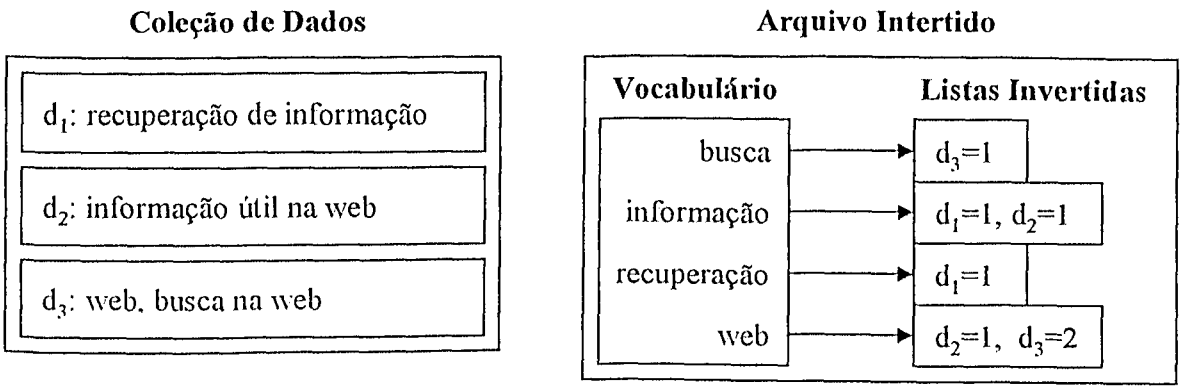

Figura 2.2 - Uma coleção de dados e um arquivo invertido construído com fundamento nela.

Um arquivo invertido é construído por meio do processamento de cada documento da coleção. Este processamento é chamado indexação e é responsável por identificar todos os termos existentes na coleção, excluindo as palavras que aparecem com muita freqüência nos documentos da coleção (ex.: artigos, preposições e conjunções), inserindo-os no vocabulário juntamente com o ponteiro para a lista de inversão. As palavras ignoradas na construção do arquivo invertido são denominadas *stop-words,* e um conjunto de *stop-words* corresponde a uma *stop-list.*

Quando uma consulta contendo mais de um termo é submetida a um sistema de recuperação de informação, o processo de busca é responsável por localizar a lista de inversão para cada termo da consulta. Dependendo dos operadores booleanos utilizados na formulação da consulta, a resposta será gerada mediante a interseção ou união dos conjuntos de documentos existentes em cada lista.

2.3 Modelos de Recuperação de Informação

Tendo representado a coleção de dados e a necessidade de informação, o próximo passo na elaboração de um sistema de recuperação de informação consiste em determinar a relação de relevância existente entre os dois. As premissas que formam a

base para o algoritmo de ordenação determinam o modelo de recuperação de informação.

Uma caracterização formal de um modelo de recuperação de informação [5] é uma quádrupla [**D**, **Q** , $\Im$, $R(q_i, d_j)$] em que

(1) **D** é um conjunto composto de visões lógicas (ou representações) para os documentos da coleção.

(2) **Q** é um conjunto composto de visões lógicas (ou representações) para as necessidades de informação do usuário. Tais representações são denominadas consultas.

(3) $\Im$ é uma estrutura para modelar representações de documentos, consultas e seus relacionamentos.

(4) $R(q_i, d_j)$ é uma função de ordenação, que associa um número real com a consulta $q_i \in \mathbf{Q}$ e a representação do documento $d_j \in \mathbf{D}$. Tal ordenação define uma ordem de relevância entre os documentos recuperados em relação à consulta q_i.

A seguir, descrevemos o Modelo Vetorial clássico de recuperação de informação e o Modelo Vetorial Generalizado, uma extensão ao clássico.

2.3.1 Modelo Vetorial

O Modelo Vetorial foi, inicialmente, proposto por Gerard Salton [27,29]. Neste modelo, todos os objetos relevantes para um sistema de recuperação de informação são representados como vetores: termos, documentos e consultas. Uma medida de distância entre vetores é utilizada para ordenar os documentos recuperados para uma consulta.

Cada termo k_i é representado como um vetor t-dimensional, em que t é o número de termos distintos da coleção. No Modelo Vetorial, o vetor $\mathbf{k_i}$ representa o termo k_i. Se a_r é o r-ésimo elemento do vetor $\mathbf{k_i}$, então

$\mathbf{k_i} = (a_1, a_2, ..., a_t)$ em que

$$\begin{cases} a_r = 0 \Leftrightarrow r \neq i \\ a_r = 1 \Leftrightarrow r = i \end{cases}$$

ou seja,

$$\mathbf{k_1} = (1, 0, 0, ..., 0)$$
$$\mathbf{k_2} = (0, 1, 0, ..., 0)$$
$$\ddots$$
$$\mathbf{k_t} = (0, 0, 0, .., 1)$$

O conjunto de todos os vetores de termos $\mathbf{K} = \{\mathbf{k_1}, \mathbf{k_2}, ..., \mathbf{k_t}\}$ é linearmente independente e forma a base canônica para o espaço $\Re^t$ do modelo vetorial. Os vetores de termos são todos ortogonais entre si e, como conseqüência, os termos correspondentes são considerados independentes. A presença de um termo não indica a presença ou ausência de outro termo. A presença de um termo também não aponta sua relação com os outros termos.

Vetores de documentos e consultas são representados utilizando o conjunto de vetores de termos **K**. Estes vetores são construídos como uma combinação linear dos vetores de termos. O vetor $\mathbf{d_j}$ associado ao documento d_j é definido por:

$$\mathbf{d_j} = \sum_{i=1}^{t} w_{i,j}\, \mathbf{k_i} \qquad \text{ou} \qquad \mathbf{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$$

Analogamente, o vetor para a consulta q é definido por:

$$\mathbf{q} = \sum_{i=1}^{t} w_{i,q}\, \mathbf{k_i} \qquad \text{ou} \qquad \mathbf{q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$$

Nas igualdades acima, $w_{i,j}$ e $w_{i,q}$ são pesos do termo i no documento j e na consulta q, respectivamente. A representação gráfica dos vetores de documentos e consultas é exemplificada na Figura 2.3.

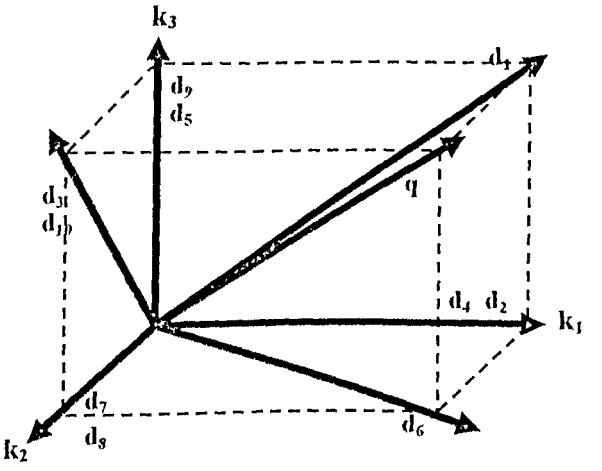

Figura 2.3 - Exemplo de documentos e consultas no espaço vetorial.

A definição de peso dos termos no modelo vetorial é baseada no fator de freqüência do termo e na freqüência inversa no documento.

Um esquema comum para o primeiro fator é empregar a freqüência do termo k_i no documento d_j. O fator de freqüência do termo é, usualmente, referenciado por tf e fornece uma medida de quão bom aquele termo descreve o conteúdo do documento [28].

A freqüência inversa do documento está relacionada à importância de um termo na coleção de documentos. Em geral, esse fator é indicado por idf e assume que a importância de um termo é inversamente proporcional ao número de documentos no qual ele aparece [28].

A definição dos pesos de termos mais eficiente para a recuperação de informação balanceia esses fatores [5].

Definição 2.1 (Peso do termo): Seja N o número total de documentos no sistema e n_i o número de documentos no qual o termo k_i aparece. Seja $freq_{i,j}$ a freqüência do termo k_i no documento d_j, $max(s)$ uma função que retorna o termo s com maior freqüência na coleção de documentos. A freqüência normalizada $f_{i,j}$ do termo k_i no documento d_j é dada por

$$f_{i,j} = \frac{freq_{i,j}}{max(s)\, freq_{s,j}}$$

em que o máximo é computado sob todos os termos mencionados no texto do documento d_j. Se o termo k_i não aparece no documento d_j, então $f_{i,j} = 0$. A freqüência inversa idf_i para k_i é dada por

$$idf_i = \log \frac{N}{n_i}$$

Se n_i é um número pequeno, então, idf_i é grande. Isto denota que o termo k_i aparece em um número pequeno de documentos, sendo, portanto, discriminante desses documentos. Esta é a razão porque idf_i é grande quando n_i é pequeno.

Geralmente, o peso do termo i no documento j é dado por

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

ou por variações desta fórmula. Essa estratégia de peso dos termos é denominada *tf-idf* [5].

O modelo vetorial avalia o grau de similaridade do documento d_j em relação à consulta q como a correlação entre os vetores $\mathbf{d_j}$ e $\mathbf{q}$. A relevância de um documento para uma consulta é proporcional à distância entre os respectivos vetores. Usualmente, essa correlação é quantificada pelo co-seno do ângulo entre esses dois vetores. Isto é,

$$sim(d_j, q) = \frac{\mathbf{d_j} \bullet \mathbf{q}}{|\mathbf{d_j}| \times |\mathbf{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \quad \text{(Eq. 2.1)}$$

A similaridade do documento em relação à consulta utiliza um fator de normalização no tamanho do documento. Documentos grandes são mais propensos a serem recuperados que documentos pequenos uma vez que os maiores, comumente, têm um conjunto de termos maior que os menores [28].

Como $w_{i,j} \geq 0$ e $w_{i,q} \geq 0$, então, $0 \leq sim(d_j, q) \leq 1$. Os documentos mais similares (mais próximos no espaço) à consulta são considerados relevantes para o usuário e retornados como resposta para a consulta. Após o cálculo dos graus de similaridade, é possível montar uma lista ordenada (*ranking*) de todos os documentos e seus respectivos graus de relevância à consulta, em ordem decrescente de relevância.

O Modelo Vetorial apresenta algumas vantagens: (i) a atribuição de pesos melhora o desempenho; (ii) sua estratégia de encontro parcial possibilita a recuperação de documentos que aproximam as condições da consulta; (iii) os documentos são ordenados de acordo com seu grau de similaridade com a consulta. A maior desvantagem do Modelo Vetorial é assumir que os termos são mutuamente independentes. Outra desvantagem é que um documento relevante pode não conter termos da consulta.

2.3.2 Modelo Vetorial Generalizado

O Modelo Vetorial Generalizado é uma extensão do Modelo Vetorial proposta em [38,39] que captura a correlação entre os termos que co-ocorrem na coleção de documentos e inclui esta informação de co-ocorrência no modelo original. Nesse modelo, os vetores de termos são linearmente independentes, porém não são necessariamente ortogonais.

No Modelo Vetorial Generalizado, os vetores de termos podem ser não ortogonais. Isto mostra que esses vetores não são os vetores ortogonais que formam a base do espaço. Os vetores de termos são representados por componentes menores denominados mintermos.

Definição 2.2: Dado o conjunto de termos da coleção $\{k_1, k_2, ..., k_t\}$, seja $w_{i,j}$ o peso do termo k_i no documento d_j. Se os pesos $w_{i,j}$ são todos binários, então, todas as possibilidades de co-ocorrência de termos nos documentos podem ser representadas por um conjunto de 2^t mintermos, em que t é o número de termos da coleção, como segue:

$$m_1 = (0, 0, 0, .., 0)$$
$$m_2 = (1, 0, 0, .., 0)$$
$$m_3 = (0, 1, 0, .., 0)$$
$$m_4 = (1, 1, 0, .., 0)$$
$$\ddots$$
$$m_{2^t} = \underbrace{(1, 1, 1, .., 1)}_{t \text{ termos}}$$

Seja $g_i(m_j)$ a função que retorna o peso do termo k_i no mintermo m_j, em que

$$g_i(m_j) = \begin{cases} 1 \Leftrightarrow \text{o } i\text{-ésimo termo de } m_j \neq 0 \\ 0 \Leftrightarrow \text{o } i\text{-ésimo termo de } m_j = 0 \end{cases}$$

Assim, o mintermo m_1 (no qual $g_i(m_1) = 0$ para todo i) representa os documentos não contendo nenhum dos termos. O mintermo m_2 (no qual $g_i(m_1) = 1$ para $i = 1$ e $g_i(m_1) = 0$ para $i > 1$) representa documentos abrangendo somente o termo k_1. Analogamente, o mintermo m_{2^t} representa documentos compreendendo todos os termos da coleção. Portanto, cada mintermo representa um conjunto de documentos.

A idéia central do Modelo Vetorial Generalizado é introduzir um conjunto de vetores ortogonais $\mathbf{m}_i$, associado ao conjunto de mintermos e adotar esse conjunto como a base do subespaço de interesse. O conjunto de vetores $\mathbf{m}_i$ é definido por

$$\mathbf{m}_1 = (1, 0, 0, 0, .., 0)$$
$$\mathbf{m}_2 = (0, 1, 0, 0, .., 0)$$
$$\mathbf{m}_3 = (0, 0, 1, 0, .., 0)$$
$$\mathbf{m}_4 = (0, 0, 0, 1, .., 0)$$
$$\ddots$$
$$\mathbf{m}_{2^t} = \underbrace{(0, 0, 0, 0, .., 1)}_{2^t \text{ termos}}$$

em que cada vetor $\mathbf{m}_i$ está associado com o respectivo mintermo m_i.

Note-se que $\mathbf{m_i} \bullet \mathbf{m_j} = 0$ para todo $i \neq j$ e, assim, por definição, o conjunto de vetores $\mathbf{m_i}$ é ortogonal. Além disso, o conjunto de vetores $\mathbf{m_i}$ forma a base ortogonal do Modelo Vetorial Generalizado em $\Re^{2^t}$.

Os termos da coleção estão correlacionados pelos vetores $\mathbf{m_i}$, que representam os mintermos m_i. Por exemplo, o vetor $\mathbf{m_4}$ está associado com o mintermo $m_4 = (1, 1, 0, .., 0)$ que indica os documentos abrangendo apenas os termos k_1 e k_2. Se existe um documento na coleção que contenha apenas estes dois termos, o mintermo m_4 está ativo e a dependência entre os termos k_1 e k_2 é induzida.

Para determinar o vetor de termo $\mathbf{k_i}$ associado com o termo k_i, inicialmente definimos o fator de correlação $c_{i,r}$.

$$c_{i,r} = \sum_{d_j \mid g_l(d_j) = g_l(m_r) \text{ para todo } l} w_{i,j}$$

em que i é o índice do termo k_i, r é o índice do mintermo m_r e j é o índice do documento d_j. Esse fator soma o peso do termo k_i em todos os documentos que tenham o mesmo padrão de co-ocorrência de m_r.

O vetor de termo é uma combinação linear de alguns vetores da base. Os vetores da base considerados correspondem aos mintermos que têm o valor do fator de correlação diferente de zero para o termo considerado. O vetor resultante é normalizado pela raiz quadrada da soma dos valores de correlação. Assim, o vetor $\mathbf{k_i}$ é definido por

$$\mathbf{k_i} = \frac{\sum_{\forall r, g_l(m_r) = 1} c_{i,r} * \mathbf{m_r}}{\sqrt{\sum_{\forall r, g_l(m_r) = 1} c_{i,r}^2}}$$

No Modelo Vetorial clássico $\mathbf{k_i}$ é o vetor que corresponde ao termo k_i. Por outro lado, no Modelo Vetorial Generalizado essa correspondência não existe. O vetor $\mathbf{k_i}$ não necessariamente corresponde ao termo k_i. Isso ocorre porque os vetores $\mathbf{m_r}$ da base estão relacionados com os mintermos m_r, que definem a co-ocorrência dos termos na coleção de documentos. Além disso, a ordenação dos mintermos não é especificada no modelo.

Uma vez estabelecidos os vetores de termos $\mathbf{k}_i$, é possível determinar os vetores de documentos e consultas. Assim como no Modelo Vetorial clássico, documentos e consultas são definidos como combinações lineares dos vetores de termos. O vetor $\mathbf{d}_j$ associado ao documento d_j é definido por:

$$\mathbf{d}_j = \sum_{i=1}^{t} w_{i,j}\,\mathbf{k}_i$$

Analogamente, o vetor para a consulta q é definido por:

$$\mathbf{q} = \sum_{i=1}^{t} w_{i,q}\,\mathbf{k}_i$$

Os vetores resultantes $\mathbf{d}_j$ e $\mathbf{q}$ são então utilizados para calcular a similaridade mediante a função padrão cosseno.

2.4 Avaliação de Sistemas de RI

Os sistemas de RI, em geral, são avaliados sob os seguintes aspectos [5]:

- ✓ **Funcionalidade:** capacidade do sistema de desempenhar as funcionalidades especificadas no mesmo;
- ✓ **Tempo de resposta:** tempo gasto pelo sistema para responder a uma solicitação do usuário;
- ✓ **Espaço utilizado:** espaço em memória necessário para armazenar as estruturas de dados do sistema;
- ✓ **Eficácia de recuperação:** este aspecto está relacionado à capacidade do sistema retornar todos os documentos relevantes (existentes na coleção), e apenas estes, para uma dada consulta. Consiste em o quão preciso é o conjunto resposta retornado por um sistema de RI a uma dada consulta.

Há um *tradeoff* inerente às medidas de tempo gasto e espaço utilizado, o qual faz com que o sistema de RI tenha sempre que dar preferência a um em detrimento do outro.

A avaliação do sistema de RI, proposto neste trabalho, está centrada na eficácia da recuperação, sendo detalhada no capítulo 5.

Duas questões importantes ao se tratar da avaliação de sistemas de recuperação são "O que avaliar?" e "Como avaliar?". Para responder a essas perguntas detalhamos a seguir, algumas métricas e métodos utilizados na avaliação da eficácia de sistemas de RI. Inicialmente, faremos uma breve discussão sobre o conceito de relevância.

2.4.1 Relevância

Um documento é considerado relevante a uma dada consulta se ele atende à necessidade de informação do usuário, que é expressa por meio de uma consulta, normalmente, constituída de um conjunto de palavras-chave.

Relevância é um conceito cognitivo, e por esta razão, diferentes usuários podem possuir diferentes conceitos de relevância. Todavia, a diferença entre os conceitos de relevância não é representativa a ponto de invalidar experimentos. Tais experimentos são realizados com base no julgamento de relevância de documentos pertencentes a uma dada coleção, com o objetivo de calcular medidas de eficácia de recuperação. Voohees [35] verificou que o julgamento de relevância de uma dada coleção por um grupo de usuários gera resultados equivalentes aos obtidos por essa tarefa realizada por um único usuário. A maioria dos experimentos que necessitam do julgamento de relevância de usuários comuns, com necessidades de informações específicas, ficam incumbidos do julgamento de relevância de um conjunto de documentos.

O julgamento de relevância pode ser feito pela classificação do documento como interessante ou não, ou pode consistir em um julgamento mais refinado, no qual é associado um nível de importância ao documento – excelente, bom, regular,

irrelevante. Exemplos da utilização desses tipos de julgamentos são encontrados em [17].

2.4.2 Métricas de Avaliação da Eficácia em Sistemas de RI

As métricas de avaliação da eficácia em sistemas de RI correspondem a medidas que refletem a habilidade do sistema em satisfazer o usuário. Essas medidas respondem, portanto, à questão enunciada no início deste capítulo: "O que deve ser avaliado em um sistema de RI?". As principais métricas de avaliação da eficácia são: a precisão e a revocação. A seguir, apresentamos as características principais destas métricas.

Precisão

Esta métrica consiste na fração dos documentos retornados realmente relevantes. Ou seja, é o número de documentos relevantes do conjunto resposta retornado ao usuário (Ra), sobre o número de documentos do conjunto resposta (A) .

$$Precisão = \frac{|Ra|}{|A|}$$

A Figura 2.4 mostra a representação gráfica da divisão dos documentos entre relevantes e retornados durante o processo de recuperação de informação.

Revocação

O valor de revocação corresponde à fração do conjunto dos documentos relevantes existentes na base. Ou seja, é o número de documentos relevantes do conjunto resposta retornado ao usuário (Ra) sobre o número de documentos relevantes existentes na coleção àquela consulta (R) .

$$Revocação = \frac{|Ra|}{|R|}$$

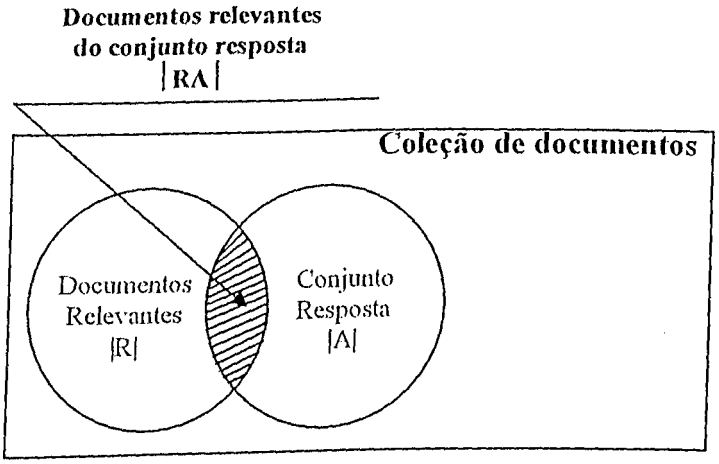

Figura 2.4 - Representação gráfica dos documentos no processo de recuperação de informação.

As medidas de precisão e revocação mostradas acima são aplicadas a uma única consulta. Entretanto, geralmente, algoritmos de recuperação são avaliados utilizando várias consultas distintas. Nesse caso, para cada consulta, uma curva precisão versus revocação distinta é gerada. Dada uma consulta q_i, o gráfico revocação x precisão, a curva é definida pelos pares $(r, p_i(r))$, tal que os níveis de revocação são r = 0%, 10%, 20%,..., 100%, e $p_i(r)$ é a precisão no nível de revocação r para a consulta q_i. Para avaliar a eficácia de recuperação de um algoritmo com todas as consultas teste, a média da precisão é calculada para cada nível de revocação, como segue:

$$\overline{P}(r) = \sum_{i=1}^{Nq} \frac{P_i(r)}{N_q}$$

em que $\overline{P}(r)$ é a precisão média no nível de revocação r, Nq é o número de consultas utilizadas, e $P_i(r)$ é a precisão no nível de revocação r para a i-ésima consulta.

Uma vez que os níveis de revocação para cada consulta podem ser distintos dos 11 níveis de revocação padrão r = 0%, 10%, 20%,..., 100%, a utilização de um

procedimento de interpolação é, usualmente, necessária. Assim, a precisão nos níveis de revocação padrão é interpolada como segue.

Seja $r_j, j \in \{0, 1, 2, ..., 10\}$, a referência ao j-ésimo nível de revocação padrão (ou seja, r_5 é uma referência ao nível de revocação 50%). Então,

$$P(r_j) = max_{r_j \leq r \leq r_{j+1}} P(r)$$

A equação acima denota que a precisão interpolada, no j-ésimo nível de revocação, é a precisão máxima conhecida em qualquer nível entre o j-ésimo nível de revocação e o $(j+1)$-ésimo nível de revocação.

Além das métricas de precisão e revocação descritas para a avaliação da eficácia de sistemas de RI, existem também métricas de valores únicos, que avaliam o sistema com uma medida única e métricas compostas, criadas para representar em uma única medida o comportamento de duas ou mais medidas simples.

2.5 Considerações Finais

Neste capítulo, apresentamos as principais técnicas envolvidas no processo de recuperação de informação. Foram descritos: algumas técnicas de representação dos dados, dois modelos em recuperação de informação e medidas de avaliação dos sistemas nesta área.

O Modelo Vetorial foi detalhado porquanto este trabalho propõe uma modificação nesse modelo para contemplar a dependência entre os termos. O trabalho aqui proposto foi avaliado com as medidas de Revocação e Precisão, uma vez que são medidas de avaliação padrão e são vastamente utilizadas na literatura de RI.

No capítulo seguinte, discorremos alguns conceitos básicos na área de mineração de dados necessários para o entendimento do trabalho. Propomos, neste trabalho, modificações no Modelo Vetorial, as quais empregam conceitos da Mineração de Dados.

3

REGRAS DE ASSOCIAÇÃO EM RECUPERAÇÃO DE INFORMAÇÃO

A mineração de dados [1,3, 8,15] é o conjunto de técnicas que permite explorar grandes bases de dados, em busca de informações valiosas que possam detectar padrões de comportamento, exceções, conceitos, regras, descobrindo diferentes tipos de conhecimentos em um amplo conjunto de dados.

Na área de mineração de dados, as regras de associação servem, tipicamente, para representar padrões freqüentes descobertos nos dados. A principal função das regras é caracterizar os dados, representando as regularidades encontradas. Um exemplo clássico de sua aplicação é na área de vendas, para analisar a tendência que existe na compra conjunta de *n* produtos. Esse tipo de análise pode servir como base para a criação de campanhas promocionais que melhorem os resultados de vendas [2,3].

No ambiente comercial, um exemplo de regra de associação é "Se um cliente compra vinho e pão, freqüentemente ele compra queijo, também". Esta regra expressa uma associação entre itens que podem ser produtos de um supermercado ou opções de equipamento especiais de um carro, serviços opcionais oferecidos por companhias de telecomunicação, etc. Uma regra de associação aponta que se um cliente foi escolhido ao acaso e descobre-se que ele selecionou certos itens (comprou certos produtos, escolheu certos serviços etc.), pode-se inferir, quantificado por uma porcentagem, que ele também selecionou outros itens (comprou outros produtos, escolheu outras opções etc.).

Um dos propósitos deste trabalho é empregar a mineração de dados na recuperação de informação. Em geral, a literatura referente à mineração de dados trabalha com itens e transações. Porém os algoritmos utilizados para a descoberta de regras de associação podem também ser adaptados para trabalhar com termos e documentos, identificando a co-ocorrência entre termos, como já utilizado na literatura em [23,24,25]. Neste trabalho, os conceitos existentes relacionados à mineração de dados são utilizados no contexto da recuperação de informação.

No contexto da recuperação de informação, X e Y são termos ou conjunto de termos. Considere o exemplo a seguir, que define uma regra de associação em recuperação de informação. A informação de que documentos cujo tema é turismo também discutem sobre hotéis é representada na regra de associação (1) abaixo:

turismo $\Rightarrow$ hotel
[suporte = 2% , confiança = 80%] (1)

O suporte e a confiança de uma regra são duas medidas que refletem, respectivamente, a utilidade e a certeza das regras descobertas. O suporte é um percentual em relação a toda a coleção de documentos analisados. No exemplo acima, em 2% da coleção, as palavras "turismo" e "hotel" aparecem juntas no mesmo documento. A confiança é um percentual em relação a um atributo. Uma confiança de 80% revela que 80% dos documentos que discutem turismo também discorrem sobre hotéis. Tipicamente, regras de associação são consideradas úteis se elas satisfazem um limiar de suporte e confiança mínimos [15].

Na próxima seção, tratamos dos conceitos básicos para a geração das regras de associação. Estes conceitos são apresentados no contexto da recuperação de informação, adaptado de [15].

3.1 Conceitos Básicos

Seja $\mathbf{T} = \{k_1, k_2, \ldots k_m\}$ um conjunto de termos de uma coleção de documentos D. Cada documento d_j da base de dados é representado por um conjunto de termos tal que

$d_j \subseteq \mathbf{T}$. Uma regra de associação é uma implicação da forma $A \Rightarrow B$, em que $A \subset \mathbf{T}$, $B \subset \mathbf{T}$, e $A \cap B = \varnothing$. A regra $A \Rightarrow B$ é válida no conjunto de documentos D com suporte s, se s é o percentual de documentos em D que contém $A \cup B$ (ou seja, ambos A e B). Este percentual pode ser interpretado como a probabilidade, $P(A \cup B)$. A regra $A \Rightarrow B$ tem confiança c no conjunto de documentos D se c é o percentual de documentos em D contendo A que também contém B. Este percentual pode ser interpretado como a probabilidade condicional, $P(B|A)$. Isto é,

$$\text{suporte}(A \Rightarrow B) = P(A \cup B) \qquad (2)$$

$$\text{confiança}(A \Rightarrow B) = P(B|A) \qquad (3)$$

Regras que satisfazem um suporte mínimo (*min_sup*) e uma confiança mínima (*min_conf*) são denominadas fortes.

Um conjunto de termos é referenciado como *termset*. Um *termset* que contém k termos é um *k-termset*. O conjunto {turismo, hotel} é um 2-*termset*. A freqüência de ocorrência de um *termset* é o número de documentos que contêm o *termset*, também conhecido simplesmente como freqüência de um *termset*. Um *termset* satisfaz o suporte mínimo, se sua freqüência é maior ou igual ao produto de min_sup e o número total de documentos em D. O número de documentos requeridos para o *termset* satisfazer o suporte mínimo é, portanto, referenciado como freqüência mínima. Se um *termset* satisfaz o suporte mínimo, logo, ele é um *termset* freqüente. O conjunto de *k_termsets* freqüentes é denotado por L_k.

Regras de associação são descobertas em grandes bases de dados em duas etapas:

1 – **Encontrar todos os *termsets* freqüentes:** Por definição, cada um destes *termsets* ocorrerá pelo menos tão freqüentemente quanto o suporte mínimo pré-determinado.

2 – **Gerar regras de associação fortes dos *termsets* freqüentes:** Por definição, essas regras devem satisfazer o suporte mínimo e a confiança mínima.

3.2 Algoritmo Apriori

Apriori é um algoritmo para mineração de *termsets* freqüentes para regras de associação [2,3,15]. Apriori emprega uma abordagem iterativa conhecida como busca em níveis, em que *k-termsets* são usados para explorar (*k*+1)-*termsets*. Primeiro, o conjunto de 1-*termsets* freqüente é encontrado. Este conjunto é denotado L_1. L_1 é usado para encontrar L_2, o conjunto de 2-*termsets* freqüentes, o qual é usado para encontrar L_3, e assim por diante, até que mais nenhum *k-termset* freqüente possa ser encontrado. A busca de cada L_k requer uma varredura completa no banco de dados. Para melhorar a eficiência da geração dos *termsets* freqüentes, uma importante propriedade chamada Apriori é utilizada para reduzir o espaço de busca. A propriedade é apresentada a seguir e um exemplo é ilustrado.

A fim de usar a propriedade Apriori, todos subconjuntos não vazios de um *termset* freqüente devem ser freqüentes também. Esta propriedade é baseada na observação a seguir. Por definição, se um *termset* I não satisfaz um suporte mínimo, *min_sup*, então, I não é freqüente, isto é P(I) < *min_sup*. Se um termo k_i é adicionado ao *termset* I, logo, o *termset* resultante (i.e., $I \cup \{k_i\}$) não pode ocorrer com mais freqüência que I. Portanto, $I \cup \{k_i\}$ não é freqüente também, isto é, $P(I \cup \{k_i\}) <$ *min_sup*.

Para entender como a propriedade Apriori é utilizada no algoritmo, vejamos como L_{k-1} é usado para encontrar L_k. Um procedimento de dois passos é seguido, consistindo das ações junção e poda.

O passo junção: Para encontrar L_k, um conjunto de *k-termsets* candidatos é gerado com a junção de L_{k-1} com ele próprio. Este conjunto de candidatos é denotado C_k. Sejam l_1 e l_2 *termsets* em L_{k-1}. A notação $l_i[j]$ refere-se ao j-ésimo termo em l_i (exemplo, l_1[k-2] refere-se ao segundo termos partindo do último termo em l_1). Por convenção, Apriori assume que termos dentro de um documento ou *termset* são ordenados em ordem lexicográfica. A junção L_{k-1} x L_{k-1} é executada, membros de L_{k-1} são associáveis se seus primeiros (k-2) termos são comuns. Isto é, membros l_1 e l_2 de L_{k-1} são unidos se $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] <$

l_2[k-1]). A condição l_1[k-1] < l_2[k-1] simplesmente assegura que nenhuma duplicação é gerada. O *termset* resultante formado da junção de l_1 e l_2 é {l_1 [1] l_1 [2]... l_1 [k-1] l_2 [k-1]}.

O passo poda: C_k é um conjunto que contém L_k, ou seja, seus membros podem ou não ser freqüentes, porém todos os *k-termsets* freqüentes estão inclusos em C_k. Uma varredura no banco de dados para determinar a freqüência de cada candidato em C_k, resultaria na determinação de L_k (i.e., todos os candidatos com uma freqüência maior ou igual ao suporte mínimo são freqüentes por definição e, portanto, pertencem a L_k). C_k, entretanto, pode ser imenso, consumindo muito tempo e recursos computacionais. Para reduzir o tamanho de C_k, a propriedade Apriori é usada como segue. Qualquer (*k*-1)-*termset* que não é freqüente não pode ser um subconjunto de um *k-termset* freqüente. Logo, se qualquer subconjunto (*k*-1) de um *k-termset* candidato não está em L_{k-1}, então o candidato também não pode ser freqüente e, assim, pode ser removido de C_k. Esse teste do subconjunto pode ser feito rapidamente, mantendo uma *hash tree* de todos os *termsets* freqüentes.

Vejamos um exemplo simples do algoritmo Apriori. Considere-se um vocabulário de cinco termos **T** = {*a*, *b*, *c*, *d*, *e*} e uma coleção de nove documentos d_j, $1 \leq j \leq 9$, que contém estes termos, D = {(*a*, *b*, *e*), (*b*, *d*), (*b*, *c*), (*a*, *b*, *d*), (*a*, *c*), (*b*, *c*), (*a*, *c*), (*a*, *b*, *c*, *e*), (*a*, *b*, *c*)}, como mostrado na Tabela 3.1.

Tabela 3.1- Coleção de documentos exemplo

Documentos	Lista de termos
d_1	*a*, *b*, *e*
d_2	*b*, *d*
d_3	*b*, *c*
d_4	*a*, *b*, *d*
d_5	*a*, *c*
d_6	*b*, *c*
d_7	*a*, *c*
d_8	*a*, *b*, *c*, *e*
d_9	*a*, *b*, *c*

1) Na primeira iteração do algoritmo, cada termo é membro do conjunto de 1-*termsets* candidatos, C_1. O algoritmo simplesmente percorre todos os documentos com o objetivo de contar o número de ocorrências de cada termo.

2) Suponhamos que a freqüência mínima de documentos seja 2 (min_sup = 2/9 = 22%). O conjunto de 1-*termsets* freqüentes, L_1, pode, assim, ser determinado. Consiste dos 1-*termsets* candidatos satisfazendo a freqüência mínima, ilustrada na figura 3.1(a).

3) Para descobrir o conjunto de 2-*termsets* freqüentes, L_2, o algoritmo usa a junção L_1 x L_1 para gerar o conjunto de 2-*termsets* candidatos, C_2.

4) Em seguida, os documentos em D são percorridos, e a freqüência de cada *termset* candidato é acumulada, como mostra a Figura 3.1 (b). O conjunto de 2-*termsets* freqüentes, L_2, é ,portanto, determinado, consistindo dos 2-*termsets* candidatos em C_2 cuja freqüência é maior ou igual à freqüência mínima.

5) A geração do conjunto de 3-*termsets* candidatos, C_3, é detalhada a seguir. Primeiro, $C_3 = L_2$ x $L_2 = \{\{a,b,c\},\{a,b,e\},\{a,c,e\},\{b,c,d\},\{b,c,e\},\{b,d,e\}\}$. Baseado na propriedade Apriori de que todos os subconjuntos de um *termset* freqüente devem também ser freqüente, podemos determinar que os últimos quatro candidatos não podem ser freqüentes. Nós, então, os removemos de C_3, economizando assim o esforço desnecessário de contar a freqüências desses *termsets* na varredura dos documentos.

6) Os documentos em D são percorridos a fim de determinar L_3 (figura 3.1(c)), consistindo dos 3-*termsets* candidatos em C_3 tendo a freqüência mínima.

7) O algoritmo usa L_3 x L_3 para gerar o conjunto de 4-*termsets* candidatos, C_4. Embora a junção resulte em $\{\{a,b,c,e\}\}$, este *termset* é podado, já que seu subconjunto $\{\{b,c,e\}\}$ não é freqüente. Logo, $C_4 = \varnothing$, e o algoritmo termina, encontrando os *termsets* freqüentes.

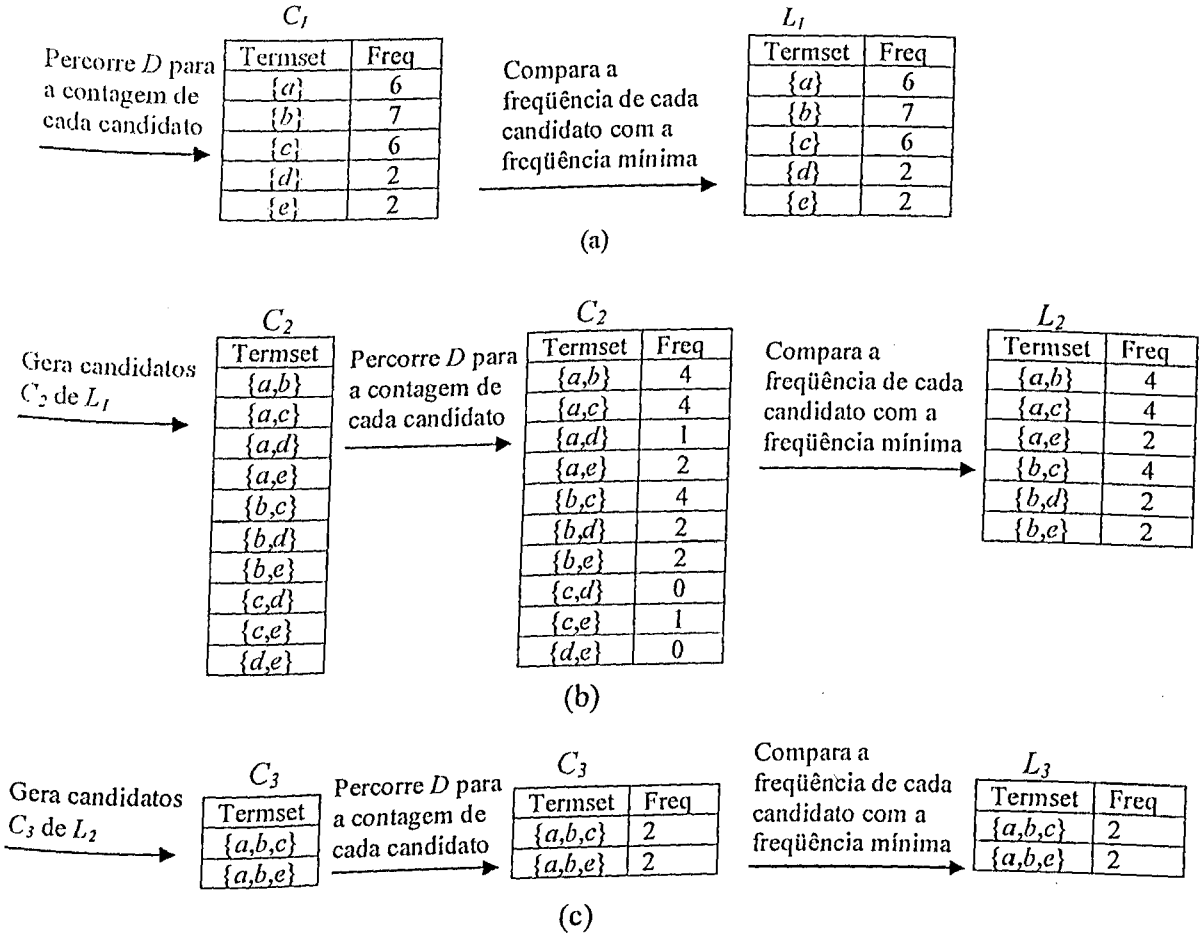

Figura 3.1- Gerando os termsets candidatos e os termsets freqüentes, onde a freqüência mínima é 2.

No desenvolvimento anterior, C_3 é determinado pelos passos a seguir:

1. Junção: $C_3 = L_2 \times L_2 = \{\{a,b\},\{a,c\},\{a,e\},\{b,c\},\{b,d\},\{b,e\}\} \times \{\{a,b\},\{a,c\},\{a,e\},\{b,c\},\{b,d\},\{b,e\}\} = \{\{a,b,c\},\{a,b,e\},\{a,c,e\},\{b,c,d\},\{b,c,e\},\{b,d,e\}\}$.

2. Poda utilizando a propriedade Apriori: Todos os subconjuntos não vazios de um *termset* freqüente também devem ser freqüentes. Vejamos cada um dos candidatos:
   - Os subconjuntos de tamanho dois de $\{a,b,c\}$ são $\{a,b\}$, $\{a,c\}$ e $\{b,c\}$. Todos estes subconjuntos são itens de L_2. Então, mantenha-se $\{a,b,c\}$ em C_3.
   - Os subconjuntos de tamanho dois de $\{a,b,e\}$ são $\{a,b\}$, $\{a,e\}$ e $\{b,e\}$. Todos estes subconjuntos são itens de L_2. Dessa forma, $\{a,b,e\}$ permanece em C_3.

- Os subconjuntos de tamanho dois de $\{a,c,e\}$ são $\{a,c\}$, $\{a,e\}$ e $\{c,e\}$. $\{c,e\}$ não é um membro de L_2, sendo assim, não é freqüente. Então, remova-se $\{a,b,c\}$ de C_3.
- Os subconjuntos de tamanho dois de $\{b,c,d\}$ são $\{b,c\}$, $\{b,d\}$ e $\{c,d\}$. $\{c,d\}$ não é um membro de L_2, sendo assim, não é freqüente. Então, remova-se $\{b,c,d\}$ de C_3.
- Os subconjuntos de tamanho dois de $\{b,d,e\}$ são $\{b,d\}$, $\{b,e\}$ e $\{d,e\}$. $\{d,e\}$ não é um membro de L_2, sendo assim, não é freqüente. Então, remova-se $\{b,d,e\}$ de C_3.

3. Logo, $C_3 = \{\{a,b,c\}, \{a,b,e\}\}$ após a poda.

A

Figura 3.2 mostra um pseudocódigo para o Algoritmo Apriori e seus procedimentos relacionados adaptados de [15].

Algoritmo Apriori: Encontra termsets freqüentes usando uma abordagem iterativa baseada em geração de candidatos.

Entrada: Lista invertida, *D*, de documentos; Suporte mínimo (min_sup).

Saída: *L*, *Termsets* freqüentes em *D*

Método:

```
L1 = find_frequent_1_termsets(D);
Para (k=2;Lk-1 ≠ ∅; K++) {
 Ck = apriori_gen(Lk-1,min_sup)
 para cada documento d ∈ D {
   Ct = subset(Ck, t);
   para cada candidato c ∈ Ct
     c.count++;
 }
 Lk = {c ∈ CK| c.count ≥ min_sup}
}
retorna L = ∪kLk;
```

procedure apriori_gen(L_{k-1}: (*k*-1)-*termsets* frequentes; *min_sup*: limiar suporte mínimo)

```
    para cada termset l1 ∈ LK-1
     para cada termset l2 ∈ LK-1
       se (l1[1] = l2[1] ) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k-2] = l2[k-2]) ∧ (l1[k-1] < l2[k-1]) então {
          c = l1 x l2;
          se has_infrequent_subset(c,Lk-1) então
             deleta c;
          senão adiciona c a Ck;
     }
    retorna Ck;

procedure has_infrequent_subset(c: k-termsets candidatos; Lk-1: (k-1)-termsets frequentes);
    para cada (k-1)-subconjunto s de c
      se s ∉ Lk-1 então
        retorna Verdadeiro;
    retorna Falso;
```

Figura 3.2 - O Algoritmo Apriori para descoberta de termsets freqüentes para a mineração de regras de associação.

3.3 Gerando Regras de Associação de *Termsets* Freqüentes

Uma vez que os *termsets* freqüentes dos documentos em D foram encontrados, as regras de associações fortes (regras que satisfazem a um suporte e confiança mínimos) podem ser geradas. Isto pode ser feito empregando a seguinte equação para a confiança, em que a probabilidade condicional é expressa em termos da freqüência do *termset*:

$$\text{Confiança}(A \Rightarrow B) = P(B|A) = \frac{freq(A \cup B)}{freq(A)}$$

em que freq($A \cup B$) é o número de documentos contendo os *termsets* $A \cup B$, e freq(A) é o número de documentos contendo A. Baseadas nesta equação, regras de associação podem ser geradas como segue:

- Para cada *termset* freqüente l, gerar todos os subconjuntos não vazios de l.
- Para cada subconjunto não vazio s de l, gerar a regra "$s \Rightarrow (l\text{-}s)$" se $\frac{freq(l)}{freq(s)} \geq min\_conf$, onde min_conf é o limiar da confiança mínima.

Visto que as regras são geradas utilizando os *termsets* freqüentes, cada uma automaticamente, satisfaz o suporte mínimo.

Exemplo: Vejamos um exemplo da geração de regras de associação de um *termset* freqüente mostrado na Figura 3.1. Suponhamos $l = \{a,b,e\}$, os subconjuntos não vazios de l são $\{a,b\}$, $\{a,e\}$, $\{b,e\}$, $\{a\}$, $\{b\}$ e $\{e\}$. As regras de associação resultantes e respectivas confianças são listadas a seguir:

$a, b \Rightarrow e$ confiança = 2/2 = 100%
$a, e \Rightarrow b$ confiança = 2/3 = 66%
$b, e \Rightarrow a$ confiança = 2/3 = 66%
$a \Rightarrow b, e$ confiança = 2/3 = 66%
$b \Rightarrow a, e$ confiança = 2/5 = 40%
$e \Rightarrow a, b$ confiança = 2/4 = 50%

Se o limiar de confiança mínima fosse 60%, por exemplo, as quatro primeiras regras seriam regras de associação fortes.

3.4 Considerações Finais

Apresentamos, neste capítulo, a técnica da mineração de dados regras de associação. No contexto da recuperação de informação, esta técnica identifica a co-ocorrência entre termos na coleção de documentos.

Na geração das regras de associação, inicialmente, é necessário encontrar conjuntos de termos que satisfazem uma freqüência mínima especificada, os *termsets* frequentes. Para executar esta tarefa, detalhamos o algoritmo Apriori. Neste trabalho, empregamos uma variação desse algoritmo para a mineração de pares de termos que co-ocorrem na coleção.

Os conceitos apresentados neste capítulo são utilizados para gerar informações sobre dependência entre os termos. Estas informações são incorporadas ao modelo aqui proposto, descrito no próximo capítulo.

4

DEPENDÊNCIA ENTRE TERMOS NO MODELO VETORIAL

Este capítulo apresenta as modificações propostas ao Modelo Vetorial para contemplar dependências entre os termos. Neste trabalho, descrevemos dois métodos que permitem gerar dependência entre os termos, as regras de associação e a geração de termos lexicograficamente semelhantes.

No próximo capítulo, mostramos os experimentos realizados com estes dois métodos.

4.1 Descrição do Problema

Como foi citado no Capítulo 2, o Modelo Vetorial clássico é um dos modelos mais populares na área de Recuperação de Informação. Sua definição de peso do termo no documento e o casamento parcial da consulta com os documentos resultam em uma boa estratégia de *ranking*. Além disso, computacionalmente, é simples e rápido [5,9,27]

Embora seja um dos modelos de recuperação de informação mais utilizados, o Modelo Vetorial clássico expõe desvantagens [5,9,36]. Os documentos são

representados por palavras-chave extraídas dos documentos, e não é considerada a relação entre os termos. Isto significa, por exemplo, que o contexto em que os termos estão inseridos não é representado. Esta é uma simplificação do modelo que não corresponde à realidade. Muitas palavras têm múltiplos significados, e os termos de uma consulta poderão casar, literalmente, com termos de um documento irrelevante. Para exemplificar, podemos citar uma busca pelas palavras "rede" e "computadores". Documentos que possuem apenas a palavra "rede" podem ser recuperados e estes documentos podem estar citando rede de pescaria ou rede de descanso em vez de rede de computadores. Além disso, existem muitas maneiras de expressar um conceito, e termos de documentos relevantes que não estão indexados por algum termo da consulta não serão recuperados.

A seguir, indicamos a solução, proposta neste trabalho, para representar a dependência entre os termos no Modelo Vetorial clássico.

4.2 Solução Proposta

A solução proposta neste trabalho, para o problema descrito acima, é alterar a representação dos vetores de termos no Modelo Vetorial clássico. Neste último, os termos são representados por vetores ortogonais, e portanto, mutuamente independentes. No modelo exposto aqui, os termos são representados por vetores não necessariamente ortogonais. Os vetores, nesse modelo, são rotados no espaço, refletindo de forma geométrica a semântica de dependência entre os termos. Essa rotação pode ser feita com base em técnicas que resultem em informações sobre o relacionamento entre termos da coleção. Propomos as técnicas regras de associação e a geração de termos lexicograficamente semelhantes.

No modelo Vetorial Clássico, os termos são representados como vetores t-dimensionais, ortogonais entre si, o que, semanticamente, significa a independência entre os termos. Vejamos a representação de três termos k_1, k_2 e k_3 no Modelo Vetorial clássico na .

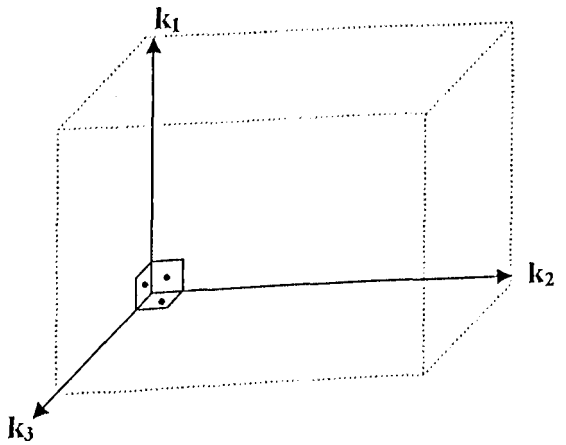

Figura 4.1 - Representação de três vetores de termos no Modelo Vetorial Clássico.

No modelo proposto, os vetores de termos não são necessariamente ortogonais. A proximidade entre dois vetores de termos é proporcional à dependência entre os respectivos termos. O ângulo entre dois vetores de termos varia entre 0 e 90 graus, sendo que um ângulo de 0 grau entre dois vetores de termos aponta que os termos têm similaridade máxima, ou seja, são equivalentes, enquanto que um ângulo de 90° entre dois vetores de termos mostra que os respectivos termos são independentes entre si. Assim, a posição do vetor de termo no espaço dependerá de sua dependência em relação aos demais termos. Supondo que o termo k_1 tenha algum tipo de relacionamento com o termo k_2, a representação dos vetores de termos no Modelo proposto poderia ser como exibido na

Figura 4.2. O valor do ângulo θ entre os vetores de termos k_1 e k_2 é proporcional ao grau de dependência do termo k_1 em relação ao termos k_2.

Na

Figura 4.2, apenas o vetor $\mathbf{k}_1$ aproximou-se do vetor $\mathbf{k}_2$ e é denominado $\mathbf{k}_1$'. Este tipo de aproximação é feita utilizando as regras de associação (seção 4.3). Note-se que $\mathbf{k}_1$' $\bullet$ $\mathbf{k}_3 = 0$, pois $\mathbf{k}_1$ foi rotado no plano definido por $\mathbf{k}_1$ e $\mathbf{k}_2$. Assim, os vetores $\mathbf{k}_1$' e $\mathbf{k}_3$ ainda são ortogonais e, conseqüentemente, os respectivos termos são independentes. Caso contrário, se os termos k_1 e k_3 tivessem algum relacionamento entre si, então teríamos $\mathbf{k}_1$' $\bullet$ $\mathbf{k}_3 \neq 0$.

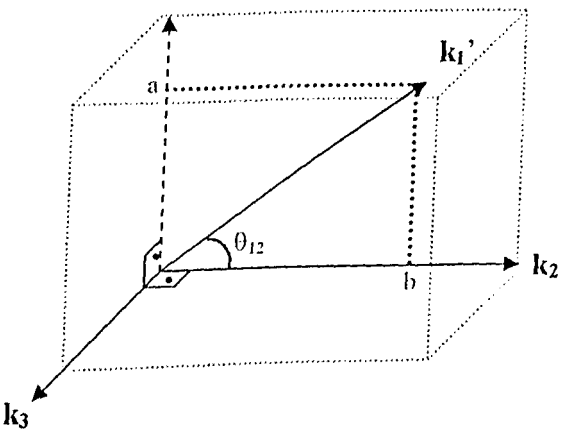

Figura 4.2 - Representação de três vetores de termos no Modelo Vetorial Modificado, onde o termo k1 está relacionado ao termo k2.

Dessa forma, os vetores de termos são modificados de acordo com as dependências encontradas para os termos da coleção. No exemplo da , o vetor $\mathbf{k}_1$, no Modelo Vetorial original, é representado por (1,0,0, ..., 0). Após a rotação do vetor $\mathbf{k}_1$ (

Figura 4.2), ele é denominado $\mathbf{k}_1$' e é representado da seguinte forma:

$\mathbf{k}_1' = (a, b, 0,0, ..., 0)$ onde $a, b \neq 0$

Semanticamente, o vetor $\mathbf{k}_1$' tem valor diferente de zero não somente no seu próprio índice, mas também no índice 2, porquanto o termo k_1 está relacionado com o termo k_2. Vale ressaltar que quanto menor o ângulo θ entre os vetores de termos $\mathbf{k}_1$ e $\mathbf{k}_2$ da

Figura 4.2, maior será o valor de b.

Dois termos podem estar também relacionados um ao outro, neste caso, os dois vetores correspondentes aos termos são rotados no espaço. Um se aproxima do outro, como na Figura 4.3.

No caso da Figura 4.3, os vetores $\mathbf{k}_1$ e $\mathbf{k}_2$ são modificados. Os novos vetores $\mathbf{k}_1$' e $\mathbf{k}_2$' são representados respectivamente por $(a_1, b_1, 0, 0,..., 0)$ e $(a_2, b_2, 0, 0,..., 0)$, em que $a_1, b_1, a_2, b_2 \neq 0$.

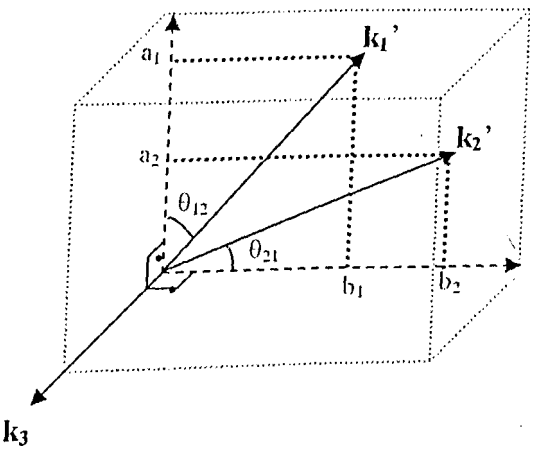

Figura 4.3 - Representação de três vetores de termos no Modelo Vetorial Modificado, onde os termos k1 e k2 estão relacionados.

Como definido na seção 2.3.1 (Modelo Vetorial), a base do espaço vetorial **K** é formada pelo conjunto de vetores de termos $\{\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_t\}$. Após a rotação dos vetores de termos, a nova base para o espaço vetorial, denominada **K'**, é obtida de **K**, substituindo os vetores $\mathbf{k}_i$ por $\mathbf{k}_i$', ou seja, $\mathbf{K'} = \{\mathbf{k}_1', \mathbf{k}_2', ..., \mathbf{k}_t'\}$. O conjunto **K'** deve continuar formando a base do espaço vetorial $\mathfrak{R}^t$. A prova desse fato será dada adiante, na proposição 4.1.

Ao alterar o espaço vetorial de termos, os vetores de documentos e consultas conseqüentemente, também são alterados. No modelo Vetorial Clássico, o vetor dj é definido como:

$$\mathbf{d_j} = \sum_{i=1}^{t} w_{ij}\, \mathbf{k_i}$$

No modelo vetorial modificado por dependência entre os termos, permanece esta definição, porém o vetor $\mathbf{k}_i$ é modificado, como foi visto acima. Logo, com a modificação dos vetores dos termos k_i, o vetor de documentos não é o mesmo definido no Modelo Vetorial clássico. Os vetores de documento $\mathbf{d_j}$ e de consulta **q** são representados na nova base **K'** para o espaço vetorial e denominados $\mathbf{d_j'}$ e **q'**. Nesta base, define-se que:

$$\mathbf{d_j}' = \sum_{i=1}^{t} w_{ij}\,\mathbf{k_i}' \qquad \text{(Eq. 4.1)}$$

$$\mathbf{q}' = \sum_{i=1}^{t} w_{iq}\,\mathbf{k_i}' \qquad \text{(Eq. 4.2)}$$

Vale observar que os vetores de documentos e consultas, $\mathbf{d_j}'$ e $\mathbf{q}'$, refletem, agora, a semântica de dependência entre os termos, implícita na base do espaço vetorial $\mathbf{K}'$. No Modelo Vetorial original, os vetores de documentos e consultas só possuem valores diferentes de zero nas posições dos termos que apareçam nos documentos e nas consultas respectivamente. Isso acontece porque $w_{i,j} = 0$ se o termo k_i não aparece no documento d_j. O mesmo fato ocorre com a consulta q. No Modelo Vetorial modificado por dependência entre os termos, os vetores de documentos e consultas podem ter valores diferentes de zero nas posições de termos que não ocorrem nos documentos e consultas respectivamente. Isto decorre dos vetores $\mathbf{k_i}$ terem sido modificados.

Geralmente, no modelo vetorial Clássico, a correlação entre o documento d_j e a consulta q é quantificada por meio do produto interno normalizado dos vetores $\mathbf{d_j}$ e $\mathbf{q}$. Essa mesma função pode ser utilizada para calcular a similaridade entre $\mathbf{d_j}'$ e $\mathbf{q}'$ no Modelo Vetorial modificado por dependência entre os termos. Assim temos,

$$sim(d_j, q) = \frac{\mathbf{d_j}' \bullet \mathbf{q}'}{|\mathbf{d_j}| \times |\mathbf{q}|} = \frac{\sum_{i=1}^{t} w_{i,j}\,\mathbf{k_i}' \cdot \sum_{s=1}^{t} w_{s,q}\,\mathbf{k_s}'}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2}\ \sqrt{\sum_{s=1}^{t} w_{s,q}^2}} = \frac{\sum_{i,s=1}^{t} w_{i,j}\,\mathbf{k_i}' \cdot w_{s,q}\,\mathbf{k_s}'}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2}\ \sqrt{\sum_{s=1}^{t} w_{s,q}^2}} \qquad \text{(Eq. 4.3)}$$

A similaridade entre a consulta e os documentos é modificada devido às alterações nos respectivos vetores. A normalização da similaridade, ou seja, os fatores no denominador da fórmula, é feita utilizando a norma original dos documentos, ou seja, as informações sobre dependência entre termos não são utilizadas para

normalizar os vetores. Essa estratégia foi adotada porque caso contrário, se a normalização ocorresse utilizando os vetores de documentos $\mathbf{d_j}$', a norma de todos os documentos deveria ser recalculada elevando muito o custo computacional do cálculo da similaridade. Além disso, essa simplificação não compromete os resultados, que apresentamos no capítulo 5.

No cálculo da similaridade, a conseqüência direta da modificação no vetor da consultas é a sua expansão. As consultas nas quais um ou mais termos possuem outros associados na coleção são automaticamente expandidas. Além disso, ao modificar os vetores de documentos, os que possuem termos da consulta mais os termos relacionados aos termos da consulta ocupam uma posição no *ranking* superior à de documentos que possuem apenas os termos da consulta e não os termos relacionados.

O ranking resultante do Modelo proposto combina o peso padrão $w_{i,j}$ do termo no documento com a informação de dependência entre os termos, implícita na base **K'**.

4.2.1 Algoritmo

O algoritmo utilizado na implementação do Modelo Vetorial modificado por dependência entre os termos é similar ao do Modelo original. Ele considera a estrutura de arquivos invertidos para a representação dos dados descrita na seção 2.2.1.

O algoritmo de busca para o Modelo Vetorial modificado por dependência entre os termos é descrito na Figura 4.4. Considera A uma lista de acumuladores, sendo que cada item A_j de A armazena a similaridade parcial do documento d_j' em relação à consulta q. A função *valor*(k_i, i) retorna o valor armazenado na posição i no vetor do termo k_i. As modificações necessárias ao algoritmo original para contemplar a dependência entre os termos, estão no passo (2) e no laço do passo (6).

(1) Crie e inicialize uma estrutura de acumuladores (A)

(2) Para cada termo k_i da consulta, adicione à consulta todos os termos associados a ele.

(3) Para cada termo k_i da consulta modificada faça:

(4) Para cada par $[d_j', f_{ij}]$ na lista invertida do termo k_i faça:

(5) aux = $w_{ij} * w_{iq} * (valor(k_i, i))^2$

(6) Para cada termo k_j associado ao termo k_i faça:

(7) aux = aux + $(w_{ij} * w_{iq} * valor(k_i, i) * valor(k_i, j))$

(8) Fim Para

(9) Se $A_j \notin A$ então

(10) A_j = aux

(11) senão

(12) $A_j = A_j$ + aux

(13) $A = A + \{A_j\}$

(14) Fim Para

(15) Fim Para

(16) Divida cada acumulador A_j pela norma do documento d_j'.

(17) Ordene a lista de acumuladores A_j e retorne os documentos d_j' recuperados.

Figura 4.4 -Algoritmo de busca para o Modelo Vetorial modificado por dependência entre os termos.

Inicialmente, em (1), é criada a lista de acumuladores para cada documento d_j, que contem os valores das similaridades parciais de cada documento com a consulta. No passo (2), há uma diferença em relação ao algoritmo original. Determinados os identificadores dos termos da consulta em uma lista, os termos associados a cada termo da consulta são adicionados á lista da consulta. Isto ocorre porque, de acordo com a Eq. 4.2, se a consulta possui um termo que, por sua vez, tem um outro termo associado a ele, o vetor **q** também terá valor diferente de zero na posição do termo associado. Assim, os termos associados são adicionados à lista da consulta. Este passo do algoritmo define a expansão automática da consulta com os termos relacionados aos termos da consulta.

Na seqüência, passo (3), para cada termo da consulta modificada é percorrida a lista de documentos e freqüências, a lista invertida. Para cada par $[d_j', f_{ij}]$ da lista invertida, é calculada a similaridade parcial do documento d_j' em relação à consulta. Os passos de (5) a (8) equivalem ao somatório $\sum_{i,s=1}^{t} w_{i,j}\mathbf{k}_i\, w_{s,q}\mathbf{k}_s$ da Eq. 4.3, ou seja, o produto

interno entre os vetores $\mathbf{d_j}$' e $\mathbf{q}$. O passo (5) corresponde ao somatório para $i = s$. E o laço do passo (6) corresponde aos outros casos, quando $i \neq s$.

No algoritmo original do Modelo Vetorial não existe o laço do passo (6). Este passo é desnecessário porque o vetor do termo k_i só tem valor diferente de zero na posição i do vetor. Isso sucede porque os vetores de termos, neste modelo, são ortogonais entre si. Por outro lado, no Modelo Vetorial modificado por dependência entre os termos, os vetores de termos não são necessariamente ortogonais, o que exige a execução do laço no passo (6). No modelo proposto, o vetor k_i terá valores diferentes de zero no seu próprio índice e nas posições correspondentes aos termos associados ao termo da consulta k_i. Esta é a razão para o laço do passo (6) ser executado apenas para essas posições.

Os passos de (9) a (13) correspondem ao procedimento original da implementação de acumular as similaridades de cada documento em relação à consulta. Após percorrer a lista invertida de todos os termos da consulta modificada, cada acumulador A_j é dividido pela norma original do respectivo documento d_j.

Finalmente, os acumuladores são dispostos em ordem decrescente, retornando os documentos recuperados na ordem definida. Ao analisar o algoritmo, notamos, claramente, que o modelo proposto é uma extensão do Modelo Vetorial original. Isso é justificado porque, se não existir associação entre os termos da coleção, o algoritmo descrito é equivalente ao algoritmo original.

4.3 Modelo Vetorial modificado por regras de associação

Nesta seção, expomos as regras de associação como ferramenta para a geração de dependência entre os termos e sua aplicação ao Modelo Vetorial modificado por dependência entre os termos.

Como foi citado no capítulo 3, as regras de associação representam padrões freqüentes descobertos na base de dados. Os algoritmos da mineração de dados para a descoberta de regras de associação, adaptados para a recuperação de informação, produzem pares de termos que co-ocorrem nos documentos de uma coleção. Assim, a

dependência entre os termos gerada pelas regras de associação está associada à co-ocorrência de termos na coleção de documentos.

As regras de associação são utilizadas para rotar os vetores de termos como definido no modelo proposto. Os vetores de termos são rotados no espaço, refletindo, de forma geométrica, a semântica definida pelas regras de associação. Este método é baseado na suposição de que um par de palavras que ocorre freqüentemente junto nos mesmos documentos está relacionado ao mesmo assunto.

O algoritmo Apriori utiliza a lista invertida de uma coleção de documentos para determinar os *termsets* freqüentes, ou seja, os pares de termos que co-ocorrem freqüentemente. O algoritmo encontra *termsets* freqüentes de todos os tamanhos, ou seja, vários conjuntos de termos que co-ocorrem na coleção de documentos. Uma simplificação do algoritmo Apriori foi empregada, uma vez que é suficiente, neste trabalho, obter apenas os 2-*termsets* freqüentes. O motivo para essa simplificação é que, se for encontrado um *termset* de tamanho 3, como, por exemplo, o conjunto de termos $\{a, b, c\}$, com certeza, os subconjuntos não vazios desse *termset* já foram encontrados $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a\}$, $\{b\}$ e $\{c\}$. E como todos os pares de termos são utilizados não há perda de informação.

Após a determinação do conjunto de *termsets* freqüentes, as regras de associação são geradas. Se $k_i \rightarrow k_j$ é uma regra de associação, c_{ij} é o índice de confiança da regra, que indica o grau de dependência do termo k_i em relação ao termo k_j. Esse índice é utilizado, neste trabalho, para calcular o novo ângulo entre os vetores de termos k_i e k_j. A confiança foi escolhida como parâmetro para determinar a aproximação dos vetores de termos, porque ela reflete a certeza da regra de associação. Os vetores de termos são aproximados uns dos outros, de acordo com as regras de associação criadas para os respectivos termos como segue.

Definição 4.1 (Rotação de vetores da base): Sejam $\mathbf{k}_i$ e $\mathbf{k}_j$ vetores de termos, c_{ij} o índice de confiança da regra de associação $k_i \rightarrow k_j$. O novo ângulo θ_{ij} entre $\mathbf{k}_i$ e $\mathbf{k}_j$ é dado por

$$\theta_{ij} = 90\,(1 - c_{ij})$$

em que 90 é o ângulo entre os vetores $\mathbf{k_i}$ e $\mathbf{k_j}$ (no modelo vetorial os vetores de termos são ortogonais). Neste caso, a rotação é feita apenas no vetor $\mathbf{k_i}$, o vetor $\mathbf{k_j}$ não é modificado. Isso acontece em razão da semântica da regra de associação e da confiança. O índice c_{ij} da regra de associação $k_i \rightarrow k_j$ determina que, em c% das vezes em que surge o termo k_i, ocorre também o termo k_j. Por isso, a rotação é feita no vetor correspondente ao termo do antecedente da regra de associação.

θ_{ij} é o novo vetor entre os vetores de termos $\mathbf{k_i}$ e $\mathbf{k_j}$ sempre que $\theta_{ij} < 90°$. O vetor $\mathbf{k_i}$ aproxima-se do vetor $\mathbf{k_j}$, e o novo vetor é denominado $\mathbf{k_i}$', em que o r-ésimo elemento do vetor $\mathbf{k_i}$', chamado a_r, é definido por:

$$\begin{cases} a_r = \text{sen}(\theta_{ij}) \Leftrightarrow r = i \\ a_r = \cos(\theta_{ij}) \Leftrightarrow r = j \\ a_r = 0 \qquad \Leftrightarrow r \neq i \; e \; r \neq j \end{cases}$$

Assim, o vetor $\mathbf{k_i}$ é transformado no vetor $\mathbf{k_i}' = (a_1, a_2, ..., a_t)$, alterando as posições i e j do vetor original. Na posição i, temos $\text{sen}(\theta_{ij})$ e, na posição j, temos $\cos(\theta_{ij})$.

No caso em que um termo k_p apresenta dois ou mais termos associados, é feita uma normalização no novo vetor k_p' como segue. Sejam as regras de associação e $k_p \rightarrow k_n$ e $k_p \rightarrow k_v$, com antecedentes iguais e respectivas confianças c_{pn}, c_{pv}, o novo vetor $\mathbf{k_p}$' é definido por

$$\mathbf{k_p}' = \frac{\mathbf{k_{pn}} + \mathbf{k_{pv}}}{|\mathbf{k_{pn}} + \mathbf{k_{pv}}|}$$

em que $\mathbf{k_{pn}}$ é o vetor $\mathbf{k_p}$ modificado utilizando $k_p \rightarrow k_n$ e c_{pn} (definição 4.1), $\mathbf{k_{pv}}$ é o vetor modificado, empregando $k_p \rightarrow k_v$ e c_{pv}.

As regras de associação determinam mudanças nos vetores de termos $\mathbf{k_i}$', modificando a base do espaço vetorial. Como a construção dos vetores de documentos e consultas é condicionada à base do espaço, ambos também são alterados, refletindo, geometricamente, a semântica de co-ocorrência definida pelas regras de associação.

Logo, a similaridade entre documentos e consulta combina o peso padrão $w_{i,j}$ e a co-ocorrência entre os termos, representada nos vetores de documentos e consulta.

As demonstrações das proposições abaixo mostram as condições em que, após a rotação dos vetores de termos pelas regras de associação, o conjunto de vetores de termos $\mathbf{K'} = \{k_1', k_2', \ldots, k_t'\}$ continua formando a base do espaço vetorial $\Re^t$.

Proposição 4.1: Seja $\mathbf{K} = \{k_1, \ldots, k_i, \ldots, k_t\}$ uma base de $\Re^t$ e $k_i \rightarrow k_j$ uma regra de associação. Se $\mathbf{K_i'} = \{k_1, \ldots, k_i', \ldots, k_t\}$ é obtida de K, substituindo k_i por k_i', utilizando a regra $k_i \rightarrow k_j$ (definição 4.1), então Ki' também é uma base de $\Re^t$ se e somente se $\theta ij < 90°$.

Demonstração: Dada uma regra de associação $k_i \rightarrow k_j$, a base $\mathbf{K} = \{k_1, \ldots, k_i, \ldots, k_t\}$ é modificada para outra base $\mathbf{K_i'} = \{k_1, \ldots, k_i', \ldots, k_t\}$, obtida de K, substituindo k_i por k_i'. Neste caso, k_i' é calculado conforme a definição 4.1. k_i' está no plano estabelecido por k_i e k_j, se $\theta ij < 90°$ então $k_i' \neq k_j$, e portanto, $\mathbf{K_i'}$ é linearmente independente. Logo $\mathbf{K_i'}$ é base de $\Re^t$. Suponhamos, agora, por absurdo que $\theta ij = 90°$. Neste caso, $k_i' = k_j$ e $\mathbf{K_i'}$ não é base. Logo Ki' é base de $\Re^t$ se e somente se $\theta ij < 90°$.

Proposição 4.2: Seja $\mathbf{K_i'} = \{k_1, \ldots, k_j, \ldots, k_i', \ldots, k_t\}$ uma base de $\Re^t$, em que k_i' e k_j não são ortogonais e $k_i \rightarrow k_u$ uma regra de associação. Se $\mathbf{K_i''} = \{k_1, \ldots, k_j, \ldots, k_i'', \ldots, k_t\}$ é obtida de $\mathbf{K_i'}$, substituindo k_i' por k_i'', utilizando $k_i \rightarrow k_u$, então Ki'' também é base de $\Re^t$ se e somente se $\theta ij < 90°$.
A demonstração dessa proposição é análoga à demonstração da proposição 4.1.

Proposição 4.3: Seja $\mathbf{K_i'} = \{k_1, \ldots, k_j, \ldots, k_i', \ldots, k_u, \ldots, k_t\}$ uma base de $\Re^t$, em que k_i' e k_j não são ortogonais e $k_u \rightarrow k_i$ uma regra de associação. Se $\mathbf{K_{iu}'} = \{k_1, k_2, \ldots, k_j, \ldots, k_i', \ldots, k_u', \ldots, k_t\}$ é obtida de $\mathbf{K_i'}$, substituindo k_u por k_u', utilizando $k_u \rightarrow k_i$, então Kiu' é base de $\Re^t$ se e somente se $\theta ij < 90°$.
A demonstração dessa proposição é análoga à demonstração da proposição 4.1.

Proposição 4.4: Seja $\mathbf{K} = \{\mathbf{k}_1, ..., \mathbf{k}_t\}$ uma base de $\mathfrak{R}^t$ e $k_i \rightarrow k_j$ e $k_j \rightarrow k_i$ duas regras de associação. Se $\mathbf{K}_{ij}' = \{\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_i', ..., \mathbf{k}_j', ..., \mathbf{k}_t\}$ é obtida de **K**, substituindo $\mathbf{k}_i$ por $\mathbf{k}_i'$, utilizando $k_i \rightarrow k_j$ e $\mathbf{k}_j$ por $\mathbf{k}_j'$, utilizando $k_j \rightarrow k_i$, então $\mathbf{K}_{ij}'$ é base de $\mathfrak{R}^t$ se e somente se $\theta ij + \theta ji < 90°$.

A demonstração dessa proposição é análoga à demonstração da proposição 4.1.

Observação: No caso em que $\theta ij + \theta ji = 90°$, podemos desconsiderar $\mathbf{k}_i$'e obter uma base menor. Neste caso, $\mathbf{k}_i$' é sinônimo de $\mathbf{k}_j$'.

4.3.1 Exemplo

Nesta seção, o mecanismo de busca do Modelo Vetorial modificado por regras de associação é exemplificado. Os resultados obtidos com o exemplo são comparados com os resultados do Modelo Vetorial clássico.

Seja D uma coleção com 7 documentos e k_1, k_2, k_3 e k_4 os termos que ocorrem na coleção D. Se os pesos dos termos nos documentos são binários, então temos a seguinte matriz termo-documento:

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
k_1	1	0	1	0	1	1	0
k_2	0	1	1	1	0	0	0
k_3	1	0	1	1	0	1	1
k_4	0	0	0	0	1	1	0

Consideremos o suporte mínimo para a geração das regras de associação 50%. Isso quer dizer que pares de termos que não ocorrem em pelo menos 4 documentos são ignorados. Aplicando o algoritmo Apriori para essa coleção de documentos, o único par de termos que satisfaz o suporte mínimo é $\{k_1, k_3\}$. Após encontrar os *termsets* freqüentes, a confiança das regras $k_1 \rightarrow k_3$ e $k_3 \rightarrow k_1$ são calculadas. Se a confiança mínima é 70% (min_conf), então, apenas a regra $k_1 \rightarrow k_3$ é considerada, visto que c_{13} =0,75 e c_{31} =0,6.

Para encontrar a aproximação do vetor k_1 na direção do vetor k_3, utilizamos a definição 4.1, e o novo ângulo entre os vetores é definido por

$$\theta_{13} = 90\ (1 - c_{13}) = 90\ (1\text{-}0{,}75) = 22{,}5$$

Logo, $\mathbf{k}_1 = (\text{sen}(\theta_{13}), 0, \cos(\theta_{13}), 0) = (0.38, 0, 0.92, 0)$.

Nesta coleção, apenas o vetor $\mathbf{k}_1$ é modificado, e, como os vetores de documentos são combinações lineares dos vetores de termos, os documentos que possuem o termo k_1 também são modificados:

$$\mathbf{d}_1 = \sum_{i=1}^{t} w_{i,1}\, k_i = w_{1,1}\mathbf{k}_1' + w_{3,1}\mathbf{k}_3' = 1(0.38, 0, 0.92, 0) + 1\ (0, 0, 1, 0) = (0.38, 0, 1.92, 0)$$

Analogamente, calculamos os outros vetores de documentos e obtemos:

$\mathbf{d}_2 = \mathbf{k}_2 = (0, 1, 0, 0)$

$\mathbf{d}_3 = \mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3 = (0.38, 1, 1.92, 0)$

$\mathbf{d}_4 = \mathbf{k}_2 + \mathbf{k}_3 = (0, 1, 1, 0)$

$\mathbf{d}_5 = \mathbf{k}_1 + \mathbf{k}_4 = (0.38, 0, 0.92, 1)$

$\mathbf{d}_6 = \mathbf{k}_1 + \mathbf{k}_3 + \mathbf{k}_4 = (0.38, 0, 1.92, 1)$

$\mathbf{d}_7 = \mathbf{k}_3 = (0, 0, 1, 0)$

Seja $q = \{k_1\}$ a consulta. A construção do vetor $\mathbf{q}$ associado à consulta q é análoga à construção dos vetores de documentos. O vetor da consulta é definido por:

$\mathbf{q} = \mathbf{k}_1 = (0.38, 0, 0.92, 0)$

Nota-se, na representação da consulta, que, embora possua apenas o termo k_1, o vetor $\mathbf{q}$ possui valor diferente de zero na posição correspondente ao termo k_3. Isso ocorre, porque o termo k_1 está associado ao termo k_3, e o vetor $\mathbf{k}_1$ foi modificado. Esta modificação no vetor da consulta garante a expansão automática com termos associados aos termos da consulta.

Após a construção dos vetores de documentos e consulta, o próximo passo é calcular a similaridade entre cada documento com a consulta. Para isto, utilizamos o produto interno normalizado entre os vetores da equação 4.3. Porém, diferente dessa equação, ignoramos a norma da consulta, porque ela não afeta a ordenação, já que é a mesma para todos os documentos. Como já foi dito, a norma utilizada no cálculo da similaridade é a norma original do Modelo Vetorial clássico, sem modificar os vetores de termos. Assim, a similaridade entre cada documento com consulta q é definida por:

$$sim(\mathbf{d_1},\mathbf{q}) = \frac{\mathbf{d_1} \bullet \mathbf{q}}{|\mathbf{d_1}|} = \frac{0.38 \times 0.38 + 0.92 \times 0.38}{1.41} = 1.35$$

Analogamente, obtemos as similaridades dos outros documentos com a consulta q:

$sim(\mathbf{d_2},\mathbf{q}) = 0$
$sim(\mathbf{d_3},\mathbf{q}) = 1.10$
$sim(\mathbf{d_4},\mathbf{q}) = 0.65$
$sim(\mathbf{d_5},\mathbf{q}) = 0.70$
$sim(\mathbf{d_6},\mathbf{q}) = 1.10$
$sim(\mathbf{d_7},\mathbf{q}) = 0.65$

Ao analisar os resultados, inicialmente, observamos que os documentos d_4 e d_7 não contêm termos da consulta, porém têm similaridade maior que zero. Isto é efeito da expansão automática, porque os documentos possuem o termo k_3 associado ao termo da consulta k_1. Para efeitos de comparação, calculamos também as similaridades entre cada documento e a consulta nessa coleção exemplo, para o Modelo Vetorial clássico, como mostra a Tabela 4.1.

Ao comparar os resultados entre os dois modelos, visualizamos algumas diferenças. No Modelo Vetorial clássico, os documentos d_1 e d_5 têm a maior similaridade, enquanto, no modelo proposto d_1 tem prioridade sobre d_5. Isto se deve ao fato de que o documento d_1, além de possuir k_1, possui também o termo associado a este termo. Por esse mesmo motivo, os documentos d_3 e d_6 têm similaridade maior que o documento d_5 no modelo modificado, o que não acontece no modelo clássico.

Tabela 4.1- Similaridades entre os documentos e a consulta nos modelos vetorial clássico e modificado por regras de associação.

Documentos	Vetorial Modificado	Vetorial Clássico
D_1	1.35	0.71
D_2	0.00	0.00
D_3	1.10	0.58
D_4	0.65	0.00
D_5	0.70	0.71
D_6	1.10	0.58
D_7	0.65	0.00

4.4 Modelo Vetorial modificado por semelhança de termos

Esta seção apresenta um outro método de geração de dependência entre termos que também pode ser utilizado para modificar o Modelo Vetorial clássico. A idéia é mostrar que a rotação dos vetores pode ser realizada com estratégias alternativas às regras de associação.

No contexto deste trabalho, termos são considerados semelhantes se um termo está contido dentro de outro termo, ou seja, a semelhança entre os termos é lexicográfica. Por exemplo, os termos 'computador', 'computadores' e 'computadorizado' são lexicograficamente semelhantes. Esta é uma estratégia semelhante à estratégia de extração de radicais (*stemming*). No Modelo Vetorial original, duas palavras tão similares semanticamente, como por exemplo, 'computador' e 'computadores', são representadas por vetores ortogonais, o que significa independência entres os respectivos termos.

Para a obtenção de todos os termos semelhantes, o vocabulário da coleção é percorrido e os termos comparados entre si. Ao comparar dois termos k_i e k_j, se k_i estiver contido dentro de k_j ou k_j estiver contido dentro de k_i, um é semelhante ao outro. Termos de tamanho menor que três caracteres foram ignorados na comparação, porque, geralmente, são palavras-chave com significados irrelevantes.

Os vetores dos termos de uma coleção de documentos que possuem termos semelhantes são aproximados uns dos outros, refletindo a dependência entre eles. Assim como ocorre com as regras de associação, a base do espaço no Modelo Vetorial é modificada para contemplar a dependência entre os termos da coleção.

Os termos semelhantes lexicograficamente têm forte dependência entre si, porquanto, usualmente, possuem o mesmo radical ou, então, um termo é plural ou sinônimo do outro. A relação entre dois termos semelhantes k_i e k_j é recíproca, ou seja, um está associado ao outro. Por esse motivo, se dois termos k_i e k_j são lexicograficamente semelhantes, os dois vetores de termos $\mathbf{k_i}$' e $\mathbf{k_j}$' são rotados no espaço. Os vetores $\mathbf{k_i}$ e $\mathbf{k_j}$ são rotados no plano definido por $\mathbf{k_i}$ e $\mathbf{k_j}$, cada vetor é rotado na direção do outro. Definimos uma aproximação fixa entre os vetores de termos semelhantes, cada vetor tem uma rotação de 30° na direção do termo associado, sendo que o ângulo entre os dois vetores também é de 30°, dividindo uniformemente o ângulo original de 90°. Consideremos, por exemplo, que k_3 e k_5 são termos lexicograficamente semelhantes, assim, os respectivos vetores $\mathbf{k_3}$' e $\mathbf{k_5}$' são:

$$\mathbf{k_3}' = (0, 0, \text{sen}(30°), 0, \cos(30°), 0, ..., 0) = (0, 0, 0.5, 0, 0.87, ..., 0)$$

$$\mathbf{k_5}' = (0, 0, \cos(30°), 0, \text{sen}(30°), 0, ..., 0) = (0, 0, 0.87, 0, 0.5, ..., 0)$$

A representação dos vetores acima no espaço é análoga à representação dos vetores $\mathbf{k_1}$' e $\mathbf{k_2}$'na Figura 4.3.

Dessa forma, novamente temos uma nova base do espaço vetorial $\mathbf{K}' = \{\mathbf{k_1}', \mathbf{k_2}', ..., \mathbf{k_t}'\}$, modificada por semelhança de termos. A prova que $\mathbf{K}'$ é uma base de $\mathfrak{R}^t$ é análoga à demonstração da proposição 4.1. Ao alterar a base do espaço, vetores de documentos e consultas, conseqüentemente, também são alterados:

$$\mathbf{d_j}' = \sum_{i=1}^{t} w_{ij}\, \mathbf{k_i}'$$

$$\mathbf{q}' = \sum_{i=1}^{t} w_{iq}\, \mathbf{k_i}'$$

A similaridade entre a consulta e os documentos é calculada como no Modelo Vetorial original, assim, o ranking resultante do Modelo Vetorial modificado por

semelhança entre os termos combina o peso padrão wi,j com a dependência lexicográfica entre os termos da coleção implícita na base K'.

4.5 Considerações Finais

Neste capítulo, vimos que, embora o Modelo Vetorial seja bastante popular, ele apresenta a desvantagem de não considerar a dependência entre os termos da coleção no cálculo da similaridade.

Explorando essa desvantagem do Modelo Vetorial, propomos uma modificação nesse modelo para contemplar a dependência entre os termos. A transformação ocorre na base do espaço vetorial, em que os vetores de termos são rotados no espaço de forma que a proximidade entre eles seja proporcional ao grau de dependência entre os termos. As alterações no algoritmo de busca do Modelo Vetorial são mínimas, não comprometendo a sua simplicidade.

Para a utilização do modelo proposto, apresentamos duas técnicas de geração de informações sobre dependência entre os termos. Uma delas são as regras de associação que contêm informações sobre termos que co-ocorrem na coleção. A outra técnica está relacionada com a geração de termos semelhantes lexicograficamente.

No capítulo seguinte, referimos os experimentos realizados com o modelo proposto e discutimos os resultados obtidos.

5

EXPERIMENTOS

Neste capítulo, apresentamos os experimentos realizados com o Modelo Vetorial modificado por dependência entre os termos. Implementamos as duas técnicas de geração de dependência entre termos e aplicamos ao modelo proposto. Em primeiro lugar, mostramos as coleções de referência utilizadas para a realização dos experimentos. Em seguida, descrevemos os procedimentos usados para a implementação das associações entre os termos e do modelo proposto. Apresentamos e discutimos os resultados na seção 5.3. A eficácia da recuperação do modelo proposto foi avaliada utilizando as medidas de Revocação e Precisão.

5.1 Coleções de Referência

Para avaliar a eficiência do Modelo Vetorial modificado por dependência entre os termos, os experimentos foram feitos com quatro coleções de referência denominadas CACM, *Cystic Fribosis* (CFC), CISI e *Third Text Retrieval Conference* (TREC-3). Uma descrição das coleções é feita a seguir, e as suas características são mostradas na Tabela 5.1.

A CACM [13] contém registros relativos a 3.204 artigos publicados no periódico *Communications of the CACM* de 1958 a 1979. Os registros incluem um resumo que foi utilizado para indexar a coleção. Os documentos que não têm resumo não foram considerados. Como resultado, os experimentos foram realizados com uma sub-coleção de 1602 documentos. A coleção pode ser obtida em [12].

A coleção CFC [32] é composta de 1239 documentos indexados com o termo '*cystic fribosis*' no banco de dados da *National Libray of Medicine*. Nesta coleção, o número total de tópicos é 100, sendo que para cada um, o banco de dados CFC inclui o conjunto de documentos relevantes especificados por peritos. Como a avaliação da relevância foi influenciada por tabelas e figuras nos documentos, em muitos casos, essas avaliações refletiram informações presentes em uma tabela ou figura, porém não descrita no texto do documento. Conseqüentemente, para vários tópicos, não foi possível formular uma consulta que recuperasse uma porção mínima dos documentos considerados relevantes por especialistas [33].

Na coleção CISI, o número total de tópicos é 76. Para cada tópico, o conjunto de documentos relevantes foi especificado. Em 50 dos 76 tópicos, a necessidade de informação é descrita em linguagem natural. Os tópicos restantes são descritos por conceitos derivados dos documentos da coleção e, por isso, não são usados em nossos experimentos. Como resultado, o número de tópicos da CISI nos experimentos é de 50.

A TREC–3 [16] é uma coleção composta por 741855 documentos de fontes diversas: *Wall Street Journal*, no período de 1987 a 1992, *Associated Press newswire*, no período de 1989 a 1990; *Computer Selects articles* (Ziff-Davis), *Federal Register*, no período de 1988 a 1989 e resumos das publicações de U.S. DOE. A coleção inclui um conjunto de 50 consultas numeradas de 151 a 200. Associada a cada consulta, existe um conjunto de documentos relevantes.

Tabela 5.1 –Características das coleções CFC, CACM, CISI e TREC-3.

Coleções de referência	Nro de palavras-chave	Nro de documentos	Nro médio de palavras por documento	Nro de consultas exemplo	Nro médio de palavras por consulta	Nro médio de documentos relevantes por consulta
CFC	2105	1239	12,2	64	4,0	39
CACM	8716	1602	46,6	50	12,7	13
CISI	9728	1460	53,6	50	9,4	50
TREC-3	1749555	741855	301,1	50	18,58	106,38

Após apresentarmos as coleções de referência utilizadas para realizar os experimentos, descrevemos os procedimentos usados para a implementação das associações entre os termos e do modelo proposto.

5.2 Implementação

A implementação do modelo apresentado é dividida em duas fases. A primeira é a geração das informações sobre dependência entre os termos, o que significa a montagem dos vetores $\mathbf{k_i}$'. Esta tarefa é toda realizada em fase de pré-processamento. A segunda fase é o desenvolvimento do modelo proposto.

Na primeira fase, a lista invertida foi empregada para gerar as regras de associação e a lista de termos da coleção foi utilizada para a criação dos termos lexicograficamente semelhantes como descrito nas seções 4.3 e 4.4 respectivamente.

Após percorrer as respectivas listas, é possível montar os vetores de termos $\mathbf{k_i}$'. Estes vetores são indexados e gravados em disco, com o intuito de agilizar o processo de recuperação de informação do modelo proposto. É gerado um arquivo que armazena um vetor de termos associados para cada termo da coleção. Esse arquivo é utilizado no processo de busca do modelo modificado por dependência entre os termos.

Alguns parâmetros podem ser ajustados durante o processo de geração das regras de associação, em que, *min_sup* e *min_conf* são, respectivamente, limiares de suporte e confiança mínimos. Realizamos experimentos e observamos que *min_sup* deve conter um valor baixo (até 5%) porque, em geral, a freqüência dos termos na coleção de documentos é baixa. Além disso, caso *min_sup* seja alto, regras de associação, envolvendo termos cuja freqüência é pequena na coleção de documentos, são descartadas. Por outro lado, *min_conf* deve conter um valor mais alto (acima de 40%), porque esse parâmetro determina a aproximação entre os vetores. Caso *min_conf* contenha um valor baixo, vetores de termos que têm co-ocorrência muito baixa são aproximados. Isto prejudica a eficácia da recuperação, porque o sistema irá fazer a expansão da consulta com termos pouco relacionados aos termos da consulta.

Na geração da lista de termos associados, muitos termos possuem vários outros associados. Conseqüentemente, documentos contendo tais termos são privilegiados no cálculo da similaridade, uma vez que os vetores de documentos são combinações lineares dos vetores de termos. Além disso, a probabilidade desses termos aparecerem em documentos longos é maior, o que privilegiaria também esses documentos. Com o intuito de evitar os problemas citados, normalizamos o vetor de termo de acordo com o número de termos associados a ele. Por exemplo, se o termo k_i tem os termos k_j, k_u e k_v associados a ele, após a rotação do vetor $\mathbf{k_i}$ (empregando a definição 4.1), o vetor é dividido pelo número de termos associados que possui, neste caso 3.

Após a indexação da lista de termos associados para cada termo da coleção, é possível implementar e executar o algoritmo descrito na seção 4.2.1, que é o algoritmo de busca do modelo proposto.

Descrevemos aqui algumas estratégias, para a geração da dependência entre os termos, que são utilizadas no algoritmo de busca. Na próxima seção, apresentamos os resultados obtidos.

5.3 Resultados

A avaliação do sistema de RI aqui proposto está relacionada com a eficácia da recuperação, ou seja, quão preciso é o conjunto resposta retornado pelo sistema de RI a uma dada consulta. As medidas Revocação e Precisão, descritas na seção 2.4.2, são utilizadas para comparar a eficiência do Modelo Vetorial modificado por dependência entre termos com a do Modelo Vetorial clássico. Apresentamos os resultados para as duas técnicas de dependência entre termos referidas neste trabalho.

A média da precisão versus revocação foi a estratégia adotada para a avaliação do sistema proposto por ser largamente utilizada na literatura de recuperação de informação. É útil porque nos permite avaliar quantitativamente a qualidade de todo o conjunto resposta e a profundidade do algoritmo. Além disso, podem ser combinadas em uma única curva [5].

O Modelo Vetorial modificado foi processado para as duas técnicas de dependência entre os termos. As tabelas 5.2 e 5.3 apresentam os resultados obtidos

para o Modelo Vetorial modificado por regras de associação para as coleções CACM, CISI, CFC e TREC-3. Além dos resultados para a abordagem proposta, apresentamos também resultados para o Modelo Vetorial clássico e os ganhos obtidos do modelo modificado relativo ao original. As Figura 5.1 e Figura 5.2 são as curvas de revocação e precisão correspondentes às TABELAS 5.2 e 5.3 respectivamente. Os resultados mostram que há um aumento na efetividade de recuperação para todas as coleções de referência avaliadas.

Tabela 5.2 – Médias de precisão para as coleções CACM e CISI no Modelo Vetorial Modificado por regras de associação e Modelo Vetorial Clássico e ganhos obtidos.

Recall	CACM			CISI		
	Vetorial Clássico	Vetorial Modificado	Ganho	Vetorial Clássico	Vetorial Modificado	Ganho
0%	74,11%	79,20%	+ 6,87%	54,81%	66,67%	+ 21,64%
10%	55,32%	56,67%	+ 2,44%	32,50%	37,13%	+ 14,25%
20%	48,51%	49,24%	+ 1,50%	22,99%	28,11%	+ 22,27%
30%	40,71%	42,99%	+ 5,60%	19,06%	21,05%	+ 10,44%
40%	33,37%	34,62%	+ 3,75%	16,74%	17,89%	+ 6,87%
50%	25,99%	29,15%	+ 12,16%	14,49%	15,55%	+ 7,32%
60%	18,97%	22,34%	+ 17,76%	12,13%	12,55%	+ 3,46%
70%	13,74%	15,37%	+ 11,86%	10,13%	9,51%	- 6,12%
80%	9,29%	10,43%	+ 12,27%	7,02%	7,48%	+ 6,55%
90%	5,53%	6,98%	+ 26,22%	3,57%	4,60%	+ 28,85%
100%	4,82%	5,87%	+ 21,78%	0,61%	0,49%	- 19,67%
Média	30,03%	32,08%	+ 6,83%	17,64%	20,09%	+ 13,89%

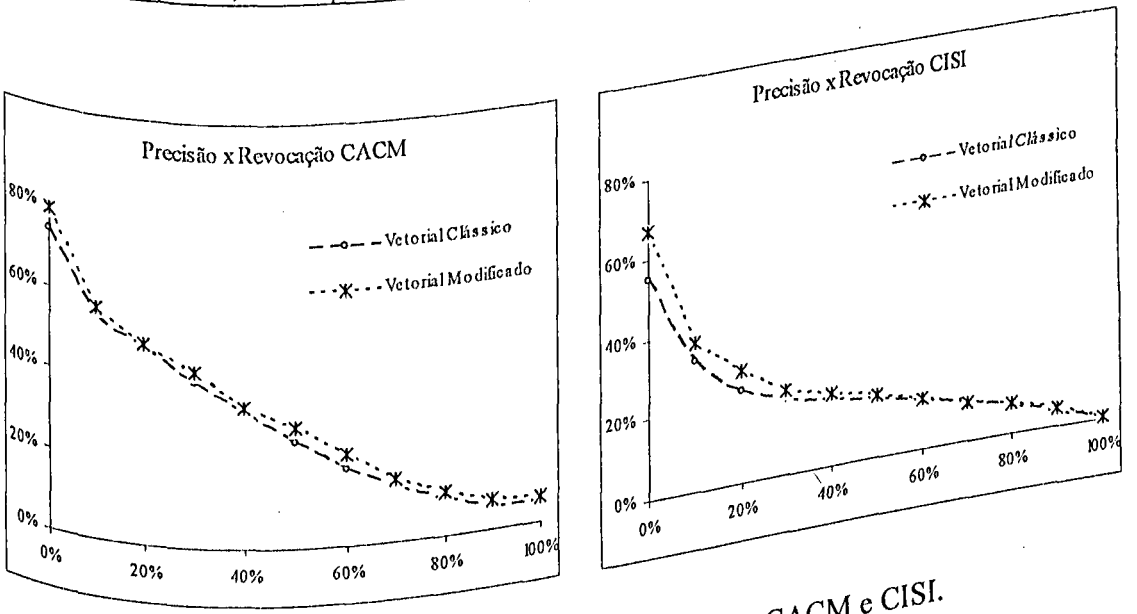

Figura 5.1 - Curvas de média da precisão para as coleções CACM e CISI.

Tabela 5.3 – Médias de Precisão para as coleções CFC e TREC-3 no Modelo Vetorial modificado por regras de associação e Modelo Vetorial clássico e ganhos obtidos.

Recall	CFC			TREC-3		
	Vetorial Clássico	Vetorial Modificado	Ganho	Vetorial Clássico	Vetorial Modificado	Ganho
0%	53,94%	53,65%	- 0,54%	52,12%	58,23%	+ 11,72%
10%	29,88%	37,38%	+ 25,10%	24,73%	26,99%	+ 9,14%
20%	17,36%	24,15%	+ 39,11%	18,51%	20,78%	+ 12,26%
30%	6,86%	15,03%	+ 119,10%	14,25%	16,35%	+ 14,74%
40%	2,33%	9,41%	+ 303,86%	7,17%	8,78%	+ 22,45%
50%	0,21%	4,67%	+ 2123,81%	5,13%	6,72%	+ 30,99%
60%	0,00%	1,04%	+ 999%	3,74%	5,06%	+ 35,29%
70%	0,00%	0,23%	+ 999%	3,06%	4,30%	+ 40,52%
80%	0,00%	0,05%	+ 999%	2,47%	3,61%	+ 46,15%
90%	0,00%	0,01%	+ 999%	1,38%	2,37%	+ 71,74%
00%	0,00%	0,00%	+ 999%	0,43%	1,29%	+ 200,00%
Média	10,05%	13,24%	+ 31,74%	12,09%	14,04%	+ 16,13%

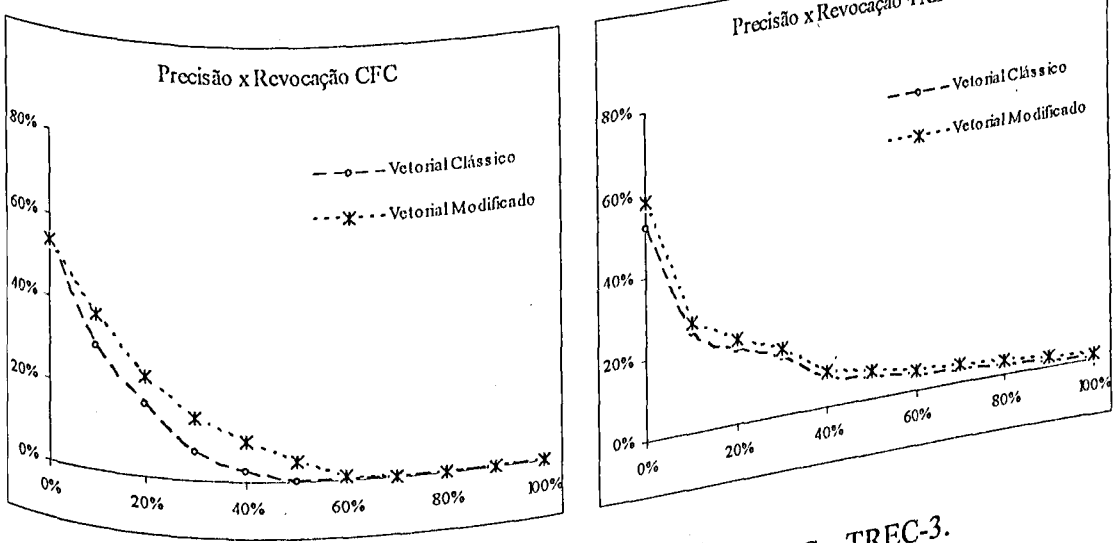

Figura 5.2 - Curvas de média da precisão para as coleções CFC e TREC-3.

Os resultados apresentados para o Modelo Vetorial modificado por regras de associação são os melhores, considerando a análise dos valores dos parâmetros feita na seção 5.2. Assim, para *min_sup* máximo de 4% a 5% e para *min_conf* alternando entre 45% a 70%, a variação dos resultados é mínima em relação à apresentada. Ao definir a

confiança mínima com um valor superior a 70%, poucas regras são geradas e, conseqüentemente, os resultados aproximam-se mais dos apresentados para o Modelo Vetorial clássico. As várias possibilidades de valores dos parâmetros foram testadas, contudo as coleções comportam-se de maneira semelhante na alteração deles.

Como pode ser visto nas TABELAS 5.2 e 5.3, em alguns níveis de revocação, os resultados foram inferiores ao do Modelo Vetorial clássico. Entretanto, para todas as coleções, a média de ganho na eficácia do modelo proposto em relação ao clássico é sempre positiva e varia entre 6,86% e 31,74%.

O Modelo Vetorial modificado por semelhança lexicográfica entre termos apresentou resultados um pouco inferiores aos apresentados acima. Os resultados são descritos nas TABELAS 5.3 e 5.4. Assim como para as regras de associação, a média de ganho na precisão foi positiva para todas as coleções avaliadas, obtendo um ganho entre 3,88% e 11,22%.

Em geral, ao gerar a dependência entre os termos empregando a semelhança lexicográfica de termos, o número de associações é maior do que quando as regras de associação são utilizadas. Para a coleção CISI, por exemplo, na semelhança lexicográfica, 4019 termos têm termos associados, sendo que a média de termos associados é aproximadamente 3. Já para as regras de associação, 783 termos têm outros associados, em que o número médio de termos associados é 5,9. Esse comportamento na geração de dependência entre os termos é análogo para as outras coleções com exceção da CFC.

A geração de termos semelhantes para a colcção CFC produziu 133 termos com associações, em que o número de médio de termos associados é 4,2. Por outro lado, para as regras de associação, 327 termos têm outros associados, e a média de termos associados é 35. Esta variância de termos associados está relacionada com as características da coleção CFC, uma coleção especializada de documentos da área médica. Além disso, essas diferenças apresentadas justificam a maior variação da média da precisão entre o Modelo Vetorial modificado por regras de associação e por semelhança de termos para a CFC.

Tabela 5.4 – Médias de Precisão para as coleções CACM e CISI no Modelo Vetorial modificado por semelhança de termos e Modelo Vetorial clássico e ganhos obtidos.

Recall	CACM			CISI		
	Vetorial Clássico	Vetorial Modificado	Ganho	Vetorial Clássico	Vetorial Modificado	Ganho
0%	74,11%	77,02%	+ 3,93%	54,81%	61,15%	+ 11,57%
10%	55,32%	62,33%	+ 12,67%	32,50%	31,79%	- 2,18%
20%	48,51%	53,51%	+ 10,31%	22,99%	25,54%	+ 11,09%
30%	40,71%	42,58%	+ 4,59%	19,06%	21,76%	+ 14,17%
40%	33,37%	32,91%	- 1,38%	16,74%	18,59%	+ 11,05%
50%	25,99%	26,55%	+ 2,15%	14,49%	17,43%	+ 20,29%
60%	18,97%	20,44%	+ 7,75%	12,13%	14,18%	+ 16,90%
70%	13,74%	15,34%	+ 11,64%	10,13%	12,19%	+ 20,34%
80%	9,29%	9,94%	+ 7,00%	7,02%	8,05%	+ 14,67%
90%	5,53%	5,91%	+ 6,87%	3,57%	4,69%	+ 31,37%
100%	4,82%	4,97%	+ 3,11%	0,61%	0,41%	- 32,79%
Média	30,03%	31,95%	+ 6,39%	17,64%	19,62%	+ 11,22%

Tabela 5.5 - Médias de Precisão para as coleções CFC e TREC-3 no Modelo Vetorial modificado por semelhança de termos e Modelo Vetorial clássico e ganhos obtidos.

Recall	CFC			TREC-3		
	Vetorial Clássico	Vetorial Modificado	Ganho	Vetorial Clássico	Vetorial Modificado	Ganho
0%	53,94%	54,11%	+ 0,32%	52,12%	60,83%	+ 16,71%
10%	29,88%	32,31%	+ 8,13%	24,73%	24,91%	+ 0,73%
20%	17,36%	18,11%	+ 4,32%	18,51%	18,29%	- 1,19%
30%	6,86%	7,46%	+ 8,75%	14,25%	15,18%	+ 6,53%
40%	2,33%	2,59%	+ 11,16%	7,17%	7,30%	+ 1,81%
50%	0,21%	0,21%	+ 0,00%	5,13%	5,53%	+ 7,80%
60%	0,00%	0,00%	0,00%	3,74%	4,40%	+ 17,65%
70%	0,00%	0,00%	0,00%	3,06%	3,79%	+ 23,86%
80%	0,00%	0,00%	0,00%	2,47%	3,12%	+ 26,32%
90%	0,00%	0,00%	0,00%	1,38%	1,88%	+ 36,23%
100%	0,00%	0,00%	0,00%	0,43%	0,91%	+ 111,63%
Média	10,05%	10,44%	+ 3,88%	12,09%	13,28%	+ 9,84%

A análise dos resultados mostra que, em geral, embora as regras de associação produzam menor número de termos com outros associados, a eficácia do Modelo Vetorial modificado por regras de associação é maior, se comparado a eficácia do modelo utilizando semelhança de termos. Isso resulta que incluir informações sobre co-ocorrência de termos no Modelo Vetorial modificado por dependência entre os

termos gera resultados melhores do que incluir termos lexicograficamente semelhantes.

5.4 Considerações Finais

Neste capítulo, descrevemos experimentos realizados com o objetivo de verificar a eficiência em se adicionar informações sobre dependência entre os termos no Modelo Vetorial clássico. Expusemos as coleções de referência utilizadas nos experimentos e os resultados do modelo proposto utilizando duas técnicas de dependência entre os termos distintas.

Os experimentos mostraram que o modelo proposto tem média de precisão superior ao modelo original para todas as coleções avaliadas. Além do mais, a precisão média obtida não foi prejudicada pelo aumento de revocação ocorrido ao expandir as consultas.

No próximo capítulo, apresentamos as conclusões deste trabalho e os trabalhos futuros a serem realizados.

CONCLUSÕES

Neste trabalho, apresentamos uma extensão ao Modelo Vetorial para contemplar a dependência entre os termos da coleção. No modelo proposto, a dependência entre os termos é representada geometricamente no espaço vetorial.

O Modelo Vetorial assume que os termos aparecem na coleção de documentos sem qualquer relação entre si. Os termos são representados por vetores de termos ortogonais entre si, porque não é conhecida a priori a correlação entre os termos. Entretanto, essa é uma simplificação que não corresponde à realidade. O algoritmo proposto neste trabalho, para lidar com esse problema tem como principal fundamento a rotação dos vetores de termos no espaço, de forma que suas representações reflitam a dependência entre os termos. A base $\{k_1, k_2, \ldots, k_t\}$ do Modelo Vetorial clássico é modificada para $\{k_1', k_2', \ldots, k_t'\}$, que não é ortonormal.

Em conseqüência á modificação da base do espaço vetorial, vetores de documentos e consultas são alterados, pois são combinações lineares dos vetores de termos da nova base. Observamos que, no novo modelo, o vetor correspondente ao documento d_j é definido por $\mathbf{d_j}' = \sum_{i=1}^{t} w_{ij}\, k_i'$. Assim, a representação de documentos e consultas inclui a informação de correlação entre os termos contida nos vetores $\{k_1', k_2', \ldots, k_t'\}$. Isto se reflete diretamente no cálculo da similaridade e ordenação dos documentos.

Mudanças de base do Modelo Vetorial também são consideradas em outros modelos da literatura. No Modelo Vetorial Generalizado, por exemplo, a base $\{k_1, k_2, \ldots, k_t\}$ é modificada para a base formada pelo conjunto de mintermos $\{m_1, m_2, \ldots, m_t\}$. Neste caso, não é clara a relação dos vetores da nova base com aqueles da base original. No modelo proposto, esta relação da nova base com a base original decorre

da semântica definida pelas regras de associação ou semelhança lexicográfica de termos.

O modelo proposto baseia-se na rotação dos vetores de termos, de acordo com a dependência entre os termos. Esta rotação pode ser feita com base em técnicas, que gerem informações sobre a dependência entre termos da coleção. Neste trabalho, empregamos as regras de associação e a geração de termos semelhantes como técnicas para a obtenção da correlação entre termos.

A geração de regras de associação é uma conhecida técnica da mineração dados, que permite encontrar padrões freqüentes em grandes bases de dados. No contexto deste trabalho, elas são utilizadas para encontrar pares de termos que ocorrem simultaneamente na coleção de documentos. Essas informações são úteis para modificar os vetores de termos, a fim de que reflitam a semântica de co-ocorrência definida pelas regras de associação.

A técnica de obtenção de termos lexicograficamente semelhantes é uma estratégia semelhante à extração de radicais. Ela considera dois termos semelhantes se um termo está contido no outro.

Neste trabalho, apresentamos uma extensão ao Modelo Vetorial que contempla a dependência entre os termos de uma forma clara, flexível e nova. É clara porque a incorporação de dependência entre os termos é feita caso a caso e a base do espaço vetorial reflete a semântica definida pela técnica adotada. O modelo proposto é flexível porque permite a incorporação de dependência entre os termos da coleção obtida de várias maneiras. E finalmente, a proposta é nova porque na literatura não há uma extensão ao Modelo Vetorial que modifica a base do espaço como foi feito neste trabalho.

Avaliamos a eficácia do modelo proposto, considerando as duas técnicas apresentadas de dependência entre os termos. As medidas padrão de eficácia de recuperação precisão e revocação foram empregadas para avaliar o Modelo Vetorial modificado por dependência entre os termos. Os experimentos foram feitos com quatro coleções de referência denominadas CFC, CACM, CISI e TREC-3. Houve um aumento na efetividade de recuperação do modelo proposto em comparação ao Modelo Vetorial clássico para todas as coleções de referência avaliadas.

Como trabalhos futuros, a eficácia do modelo proposto será comparada à eficácia do Modelo Vetorial Generalizado. Além disso, pesquisaremos outros métodos de obtenção de correlação entre os termos de uma coleção de documentos. Estes métodos serão incorporados de forma geométrica ao modelo proposto neste trabalho. Pretendemos também avaliar o modelo proposto para coleções maiores formadas por documentos *Web*.

Referências Bibliográficas

[1] Adriaans, P.; Zantige, D. **Data Mining.** Inglaterra, Addison-Wesley, 1996.

[2] Agrawal, R.; Imielinski, T.; Swami, A. **Mining association rules between sets of items in large databases.** *Proceedings of the ACM SIGMOD Conference.* Washington, DC, USA, p. 207-216, maio 1993.

[3] Agrawal, R.; Srikant, R. **Fast algorithms for mining association rules.** *Proceedings of the 20th Int'l Conference on Very Large Databases.* Santiago, Chile, September 1994.

[4] Agrawal, R.; Mannila, R.; Srikant, R.; Toivonen, H.; Verkamo, A. **Fast discovery of association rules.** *Advances in Knowlwdge Discovery and Data Mining,* AAAI Press, Menlo Park, CA, 1996, p. 307-328.

[5] Baeza-Yates, R.; Ribeiro-Neto, B. **Modern information retrieval.** *ACM/Addison-Wesley,* 1999.

[6] Bayardo, R. J. **Efficiently mining long patterns from databases.** *Proceedings of ACM SIGMOD Conference in Management of Data,* June 1998.

[7] Becker, J.; Kuropka, D. **Topic-based vector space model.** *Proceedings of the 6th International Conference on Business Information Systems,* Colorado Springs, June 2003, p. 7-12.

[8] Berry, J. A.; Linnof, G. **Data mining techniques for marketing, sales and customer support.** *Wiley Computer Publishing,* Canadá, 1997.

[9] Bollmann-Sdorra, P.; Raghavan, V. V. **On the necessity of term dependence in a query space for weighted retrieval.** *Journal of the American Society of Information Science*, 49(13): 1161-1168, 1998.

[10] Brin, S.; Motwani, R.; Ullman, J.; Tsur, S. **Dynamic itemset counting and implication rules for market basket data.** *Proceedings of ACM SIGMOD Conference in Management of Data*, May 1997.

[11] Buckley, C.; Salton, G.; Allan, J.; Singhal, A. **Automatic query expansion using SMART : TREC 3.** *In D. K. Harmon, editor, NIST Special Publication 500-225: The Third Text Retrieval conference (TREC 3),* 1995, p. 69-80.

[12] **CAM-Collection.** *ftp://ftp.cs.cornell.edu/pub/smart/cacm.*

[13] Fox, E. **Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts.** *Techinical Report* 83-561, Cornell University, Computer Science, 1983, 64p.

[14] Gordon, M.; Pathak, P. **Finding information on the world wide web: The retrieval effectiveness of search engines.** *Information Processing and Management*,35 (2), 1999, p.141-180.

[15] Han, J.; Kamber, M. **Data mining Concepts and techniques.** *San Diego: Academic Press*, 2001, p.335-393.

[16] Harman, D. **Overview of the third Text Retrieval Conference.** *Proceedings of the third Text Retrieval Conference (TREC-3),* Gaithersburg, MD,USA,1995, p. 1-20.

[17] Hawking, D.; Craswell, N.; P. Bailey, K; Korfhage R. G. **Measuring search engine quality.** *Journal of Information Retrieval, published by Kluwer Academic*, November 2001.

[18] Korfhage R. R.. **Information Storage and Retrieval.** *Wiley Computer Publishing*, 1997.

[19] Kowalski G. **Information Retrieval Systems. Theory and Implementation**, *Kluwer Academic Publishers*, 1997.

[20] Mandala, R. ; Tokunaga, T.; Tanaka, H. M. **Combining multiple evidence from different types of thesaurus for query expansion.** *Proceedings of the 22th annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, United States, August 1999, p. 191-197.

[21] Lin, D. I.; Kedem, Z. M. **Pincer-search: A new algorithm for discovering the maximum frequent set.** *Proceedings of the 6th International on Extending Database Technology*, March 1998.

[22] Nie, J. Y.; Jin, F. **Integrating logical operators in query expansion in Vector Space Model.** *Workshop on Mathematical/Formal Methods in Information Retrieval, 25th ACM-SIGIR,* Tampere, Finland, August 2002.

[23] Pôssas, B; Ziviani, N.; Meira-Jr, W.; **Enhancing the set-based model using proximity information.** *Proceedings of the 9th International Symposium of String Processing and Information Retrieval,* Lisbon, Portugal, September 2002, p. 104-116.

[24] Pôssas, B; Ziviani, N.; Meira-Jr, W.; Ribeiro-Neto, B. **Modelagem vetorial estendida por regras de associação.** *XVI Simpósio Brasileiro de Banco de Dados*, Rio de Janeiro, Brasil, 2001.

[25] Pôssas, B; Ziviani, N.; Meira-Jr, W.; Ribeiro-Neto, B. **Set-based model: A new approach for information retrieval.** *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval* , Tampere, Finland, August 2002.

[26] Rijsbergen, C. J. V. **Information Retrieval, 2nd ed.** *Butterworths,* London , *1979.*

[27] Salton, G. (ed) **The SMART retrieval system – experiments in automatic document processing.** Englewood Cliffs, *NJ: Prentice Hall*; 1971.

28] Salton, G.; Buckley, C. **Term weighting approaches in automatic text retrieval.** *Technical Report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management,* Vol.32 (4), 1996, p. 431-443.

9] Salton, G.; Lesk, M. E. **Computer evaluation of indexing and text processing.** *Journal of the ACM,* 15(1):8-36, Janeiro 1968.

)] Salton, G.; McGill M. J. **Introduction to modern information retrieval.** *MacGraw Hill,* New York, 1983.

] Savasere, A.; Omieccinski, E.; Navathe, S. **An efficient algorithm for mining association rules in large databases.** *Proceedings of 21th VLDB Conference,* 1995.

[32] Shaw, W. M.; Wood, R. E; Tiboo, H. R. **The cystic fibrosis database: Content and research opportunities.** *Library and Information Science Research,*13:347-366, 1991.

[33] Silva I., **Bayesian networks for information retrieval systems.** *Tese de Doutorado.* Banca: Ribeiro-Neto, B.; Baeza-Yates, R.; Milidú, R. L.; Laender, A. H. F; Luna, H. P. L; Ziviani, N. Universidade Federal de Minas Gerais, Julho 2000.

[34] Voorhees E. M. **Query expansion using lexical-semantic relations.** *Proceedings of the 17th ACM- SIGIR Conference,* 1993, p. 171-180.

[35] Voorhees E. M. **Variations in relevance judgments and the measurement of retrieval effectiveness.** *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 1998, p. 315-323.

36] Wong, S. K.M.; Raghavan, V. V. **The vector space model of information retrieval – A reevaluation.** *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval,* Cambridge, England, 1984 .

7] Wong, S. K.M.; Ziarko, W.; Raghavan, V. V.; Wong, P. C.N. **On Extending the Vector Space Model for Boolean Query Processing.** *Proceedings of the 9th ACM-SIGIR Conference on Research and Development in Information Retrieval.,* Pisa, Italy, p. 175 – 185.

] Wong, S. K.M.; Ziarko, W.; Raghavan, V. V.; Wong, P. C.N. **On modeling of information retrieval concepts in vector spaces.** *Proceedings of the ACM Transactions on Database Systems Volume 12* , New York, NY, USA, Junho 1987, p. 299 – 321.

[39] Wong, S. K. M.; Ziarko W.; Wong, P. C. N. **Generalized vector space model in information retrieval.** *Proceedings of the 8th ACM-SIGIR Conference on Research and Development in Information Retrieval.* New York, USA, 1985, p. 18-25.

[40] Yang K. **Literature review**, *Doctoral Dissertation* Committee: Robert M. Losee, Gary Marchionini, Gregory B. Newby ,Paul Solomon, Ellen Voorhees. *School of Information and Library Science University of North*, Janeiro 2001.

[41] Zaki, M. J.; Hsiao, C. J. **CHARM: An efficient algorithm for closed association rule mining.** *Technical Report 99-10,* Computer Science Department, Rensselaer Polytechnic Institute, October 1999.

[42] Zaki, M. J. **Generating non-redudant association rules.** *Proceedings of the 6th ACM SIGKDD International Conference on knowledge Discovery and Data Mining*, Boston, USA, August 2000, p. 34-43.

[43] Zaki, M. J.; Parthasarathy, S.; Ogihara, M.; Li, W. **New algorithms for fast discovery of association rules.** *Proceedings of the 3th International Conference on Knowledge Discovery and Data Mining*, August 1997.