

BRUNO AUGUSTO VIVAS E PÔSSAS

**UM NOVO MODELO DE ORDENAÇÃO DE
DOCUMENTOS BASEADO EM CORRELAÇÃO ENTRE
TERMOS**

Belo Horizonte
22 de agosto de 2005

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM NOVO MODELO DE ORDENAÇÃO DE
DOCUMENTOS BASEADO EM CORRELAÇÃO ENTRE
TERMOS**

Tese apresentada ao Curso de Pós-Graduação
em Ciência da Computação da Universidade
Federal de Minas Gerais como requisito par-
cial para a obtenção do grau de Doutor em
Ciência da Computação.

BRUNO AUGUSTO VIVAS E PÔSSAS

Belo Horizonte
22 de agosto de 2005

FEDERAL UNIVERSITY OF MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
GRADUATE PROGRAM IN COMPUTER SCIENCE

**SET-BASED VECTOR MODEL: A NEW APPROACH
FOR CORRELATION-BASED RANKING**

Thesis presented to the Graduate Program in
Computer Science of the Federal University
of Minas Gerais in partial fulfillment of the re-
quirements for the degree of Doctor in Com-
puter Science.

BRUNO AUGUSTO VIVAS E PÔSSAS

Belo Horizonte
August 22, 2005



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Um novo modelo de ordenação de documentos baseado em
correlação entre termos

BRUNO AUGUSTO VIVAS E PÔSSAS

Tese defendida e aprovada pela banca examinadora constituída por:

Ph. D. NIVIO ZIVIANI – Orientador
Federal University of Minas Gerais

Ph. D. WAGNER MEIRA JR. – Co-orientador
Federal University of Minas Gerais

Ph. D. BERTHIER RIBEIRO-NETO
Federal University of Minas Gerais

Ph. D. RICARDO BAEZA-YATES
Universidad de Chile & Universidad Pompeu Fabra, ES

Ph. D. IMRE SIMON
IME/University of São Paulo

Ph. D. EDLENO SILVA DE MOURA
Federal University of Amazonas

Belo Horizonte, 22 de agosto de 2005

*À minha linda e especial esposa Erica,
pelo que este trabalho representou em termos,
de horas, dias e anos subtraídos de nosso convívio.*

Agradecimentos

A colaboração de diversas pessoas foi essencial durante a realização deste trabalho. Agradeço principalmente ao professor, orientador, e amigo Nivio Ziviani pela inestimável orientação e pela oportunidade de ter trabalhado com um excelente profissional. Um agradecimento especial também aos professores Berthier Ribeiro-Neto e Wagner Meira Jr. que contribuíram ativamente para a minha formação acadêmica.

Também agradeço aos integrantes dos laboratórios *Latin* e *Speed*, amigos da *Akwan Information Technologies* e da *Smart Rpice*, colegas de sala e professores do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais que me deram totais condições de alcançar todos os meus objetivos.

Finalmente, agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de estudos (141.269/02-2), além das bolsas de produtividade em pesquisa dos meus orientadores Nivio Ziviani (520.916/94-8) e Wagner Meira Jr. (30.9379/03-2), e claro do projeto de pesquisa GERINDO (MCT/CNPq/CT-INFO 552.087/02-5). Agradeço também ao Departamento de Ciência da Computação da Universidade Federal de Minas Gerais (DCC/UFMG) pelas condições de infraestrutura proporcionadas, sem as quais este trabalho não poderia ser realizado.

Resumo

Neste trabalho apresentamos uma nova abordagem para a ordenação de documentos a partir do modelo de espaço vetorial. A sua originalidade apresenta-se em dois pontos principais: Primeiro, padrões de correlação entre os termos são levados em consideração e processados de forma eficiente. Segundo, a ponderação dos termos é baseada em uma técnica de mineração de dados chamada de regras de associação. A partir desses pontos definimos um novo mecanismo de ordenação chamado *modelo de espaço vetorial baseado em conjuntos*. Os componentes desse modelo deixam de ser os termos, e passam a ser os conjuntos de termos. Os conjuntos de termos capturam a intuição que termos semanticamente relacionados aparecem próximos em um documento. Esses conjuntos podem ser eficientemente gerados limitando sua computação a pequenos trechos de texto. Uma vez computados os conjuntos de termos, a função de ordenação é calculada a partir da frequência de um conjunto no documento e sua raridade na coleção. Nossa abordagem provê uma forma simples, efetiva, eficiente e parametrizada para o processamento de consultas disjuntivas, conjuntivas, por frases, além de ser usada para a estruturação automática de consultas. Todas as abordagens conhecidas que levam em consideração a correlação entre os termos foram projetadas somente para o processamento de consultas disjuntivas. Resultados experimentais mostram que o nosso modelo aumenta a precisão média para todas as coleções e tipos de consultas avaliados, mantendo o custo computacional adicional aceitável. Para a coleção TREC-8 de 2 gigabytes, a utilização do nosso modelo implica em um ganho de precisão média de 14.7% e 16.4% para consultas disjuntivas e conjuntivas, respectivamente, em relação ao modelo de espaço vetorial padrão. Esses ganhos aumentam para 24.9% e 30.0%, respectivamente, quando a informação de proximidade é levada em consideração. Os tempos de processamento das consultas são maiores, mas continuam comparáveis com os tempos obtidos para o modelo de espaço vetorial (o crescimento no tempo médio de processamento varia de 30% a 300%). Os resultados experimentais também mostram o sucesso do nosso modelo para a estruturação automática de consultas. Por exemplo, utilizando a TREC-8, nosso modelo gera ganhos de precisão média de aproximadamente 28% em comparação com o mecanismo de ordenação baseado na fórmula de ponderação BM25. Nossos resultados sugerem que a fórmula de ordenação do modelo de espaço vetorial baseado em conjuntos é bastante efetiva e computacionalmente viável para coleções genéricas.

Abstract

This work presents a new approach for ranking documents in the vector space model. The novelty lies in two fronts. First, patterns of term co-occurrence are taken into account and are processed efficiently. Second, term weights are generated using a data mining technique called association rules. This leads to a new ranking mechanism called the *set-based vector model*. The components of our model are no longer index terms but index termsets, where a termset is a set of index terms. Termsets capture the intuition that semantically related terms appear close to each other in a document. They can be efficiently obtained by limiting the computation to small passages of text. Once termsets have been computed, the ranking is calculated as a function of the termset frequency in the document and its scarcity in the document collection. The application of our approach provides a simple, effective, efficient and parameterized way to process disjunctive, conjunctive, phrase queries, and automatically structured complex queries. All known approaches that account for correlation among index terms were initially designed for processing only disjunctive queries. Experimental results show that the set-based vector model improves average precision for all collections and query types evaluated, while keeping computational costs small. For the 2 gigabyte TREC-8 collection, the set-based vector model leads to a gain in average precision figures of 14.7% and 16.4% for disjunctive and conjunctive queries, respectively, with respect to the standard vector space model. These gains increase to 24.9% and 30.0%, respectively, when proximity information is taken into account. Query processing times are larger but, on average, still comparable to those obtained with the standard vector model (increases in processing time varied from 30% to 300%). The experimental results also show that the set-based model can be successfully used for automatically structuring queries. For instance, using the TREC-8 test collection, our technique led to gains in average precision of roughly 28% with regard to a BM25 ranking formula. Our results suggest that the set-based vector model provides a correlation-based ranking formula that is effective with general collections and computationally practical.

Lista de Publicações

Artigos em Periódicos

1. Fonseca, B. M.; Golgher, P. B.; M., E. S.; Pôssas, B. e Ziviani, N. (2004). Discovering search engine related queries using association rules. *Journal of Web Engineering*, 2(4):215–227.
2. Pôssas, B.; Ziviani, N.; Ribeiro-Neto, B. e Meira Jr., W. (2005). Set-based vector model: An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems*, 23(4).

Artigos em Conferências

1. Pôssas, B.; Ziviani, N.; Meira Jr., W. e Ribeiro-Neto, B. (2001). Modelagem vetorial estendida por regras de associação. In *XVI Simpósio Brasileiro de Banco de Dados*, pp. 65-79, Rio de Janeiro, RJ, Brazil.
2. Pôssas, B.; Ziviani, N.; Meira Jr., W. e Ribeiro-Neto, B. (2002). Set-based model: A new approach for information retrieval. In *The 25th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237, Tampere, Finland.
3. Veloso, A. A.; Meira Jr., W.; de Carvalho, M. B.; Pôssas, B. e Zaki, M. J. (2002). Mining frequent itemsets in evolving databases. In *Second SIAM International Conference on DATA MINING*, Arlington, VA.
4. Pôssas, B.; Ziviani, N. e Meira Jr., W. (2002). Enhancing the set-based model using proximity information. In *The 9th International Symposium on String Processing and Information Retrieval*, pp. 104–116, Lisbon, Portugal.
5. Pôssas, B.; Ziviani, N.; Meira Jr., W. e Ribeiro-Neto, B. (2002). Modeling co-occurrence patterns and proximity among terms in information retrieval systems. In *The First Seminar on Advanced Research in Electronic Business*, pp. 123–131 Rio de Janeiro, Brazil.

6. Pôssas, B.; Ziviani, N.; Ribeiro-Neto, B. e Meira Jr., W. (2004). Processing conjunctive and phrase queries with the set-based model. In *The 11th International Symposium on String Processing and Information Retrieval*, pp. 171–183, Padova, Italy.
7. Fonseca, B.; Golgher, P.; Pôssas, B.; Ribeiro-Neto, B. e Ziviani, N. (2005). Concept-based interactive query expansion. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-05)*, Bremen, Germany. To appear.
8. Pôssas, B.; Ziviani, N.; Meira Jr., W. e Ribeiro-Neto, B. (2005). Maximal termsets as a query structuring mechanism. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-05)*, Bremen, Germany. To appear. Poster paper.

Relatório Técnico

1. Pôssas, B.; Ziviani, N.; Meira Jr., W. e Ribeiro-Neto, B. (2005). Maximal termsets as a query structuring mechanism. Technical Report TR012/2005, Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil. Available at <http://www.dcc.ufmg.br/~nivio/papers/tr012-2005.pdf>.

Resumo Estendido

Introdução

Recuperação de Informação (IR) concentra-se em prover aos usuários acesso a informação armazenada digitalmente. Diferentemente da área de recuperação de dados, que estuda soluções e mecanismos para o armazenamento e recuperação eficiente de dados estruturados, recuperação de informação está diretamente relacionada à extração de informação de dados semi ou não estruturados.

Os modelos de recuperação de informação mais populares para a ordenação de documentos de uma coleção (não necessariamente uma coleção formadas por documentos Web) são: (i) o modelo de espaço vetorial (Salton and Lesk, 1968; Salton, 1971), (ii) os modelos probabilísticos baseados em relevância (Maron and Kuhns, 1960; van Rijsbergen, 1979; Robertson and Jones, 1976; Robertson and Walker, 1994) e (iii) os modelos baseados em linguagens estatísticas (Ponte and Croft, 1998; Berger and Lafferty, 1999; Lafferty and Zhai, 2001). As principais diferenças entre esses modelos correspondem a representação das consultas e dos documentos, aos esquemas de ponderação dos termos e à fórmula de ordenação dos documentos a partir de uma consulta.

A grande maioria dos esquemas de ponderação de termos desses modelos assume que os termos são mutuamente independentes — essa suposição freqüentemente é adotada para simplificação da implementação e conveniência matemática. Todavia, é geralmente aceito que a exploração das correlações entre os termos em um documento podem ser utilizadas para melhorar a efetividade de recuperação para coleções genéricas. Realmente, várias abordagens distintas que utilizam a co-ocorrência entre os termos foram propostas (Rijsbergen, 1977; Harper and Rijsbergen, 1978; Raghavan and Yu, 1979; Wong et al., 1987; Cao et al., 2004; Billhardt et al., 2002; Nallapati and Allan, 2002). Entretanto, anos e anos de investigação científica têm mostrado que não é tarefa simples utilizar informação de correlações entre termos para melhorar a qualidade dos resultados de forma consistente. De fato, até hoje não se conhece nenhum mecanismo prático e eficiente que seja capaz de levar em consideração as correlações entre os termos da consulta em cada documento da coleção. Todas as abordagens citadas possuem uma mesma limitação, elas são computacionalmente ineficientes para ter algum valor prático.

Neste trabalho apresentamos a definição de um novo modelo de recuperação de informação que leva em consideração a correlação entre os termos de uma consulta, através da utilização de uma técnica de mineração de dados conhecida como regras de associação (Agrawal et al., 1993b). Apresentamos também a aplicação deste modelo a duas distintas aplicações: (i) processamento de consultas disjuntivas, conjuntivas e por frases e (ii) estruturação automática de consultas. Nosso modelo é chamado de modelo de espaço vetorial baseado em conjuntos, ou simplesmente, modelo baseado em conjuntos. Seus componentes básicos são os conjuntos de termos (*termsets*) de uma coleção de documentos. Os conjuntos de termos são enumerados pelos algoritmos de geração de regras de associação.

O modelo baseado em conjuntos corresponde ao primeiro modelo de recuperação de informação que explora efetivamente a correlação entre os termos, provê ganhos significativos de precisão e tem custos de processamento próximos aos custos do modelo de espaço vetorial, independentemente da coleção e do tipo de consulta considerado. O modelo explora também a intuição de que termos semanticamente relacionados geralmente ocorrem próximos, através da implementação de uma estratégia de poda que restringe a computação a conjuntos “próximos” de termos. Resultados parciais deste trabalho foram publicados em (Póssas et al., 2002c,a, 2004, 2005c,a).

Modelando a Correlação entre os Termos

As correlações entre os termos podem ser computadas de forma automática a partir dos índices dos termos. A forma de correlação utilizada baseia-se na ocorrência simultânea dos termos em um conjunto de documentos, obtida através do conceito de conjunto de termos.

Conjuntos de Termos

Seja $T = \{k_1, k_2, \dots, k_t\}$ o vocabulário de uma coleção C de N documentos, ou seja, o conjunto de todos os termos distintos t que aparecem nos documentos em C . Existe uma ordenação entre os termos do vocabulário que é baseada na ordem lexicográfica dos termos, tal que $k_i < k_{i+1}$, para $1 \leq i \leq t - 1$.

Definimos um n -termset S como um conjunto ordenado de n termos distintos, tal que $S \subseteq T$ e a ordem dos termos segue a ordenação mencionada. Seja $V = \{S_1, S_2, \dots, S_{2^t}\}$ o conjunto de todos os 2^t termsets que podem aparecer em todos os documentos em C . Cada termset S_i , $1 \leq i \leq 2^t$, possui uma lista invertida lS_i com os identificadores dos documentos nos quais ele aparece. Definimos também a frequência dS_i de um termset S_i como o número de ocorrências de S_i em C , ou seja, o número de documentos d_j , tal que $d_j \in C$, $1 \leq j \leq N$ e $S_i \subseteq d_j$. A frequência dS_i de um termset S_i é equivalente ao tamanho da sua lista invertida associada ($|lS_i|$).

Um termset S_i é considerado um termset freqüente se sua freqüência dS_i é maior ou igual do que um dado limite, conhecido como *suporte* no escopo das regras de associação (Agrawal et al., 1993b), mas referido neste trabalho como freqüência mínima. Como apresentado no algoritmo *Apriori* original, se um n -termset é freqüente, então todos os seus subconjuntos de tamanho $n - 1$ também são freqüentes.

Conjuntos Próximos de Termos

O conceito de conjuntos de termos pode ser estendido para considerar a proximidade entre os termos nos documentos, o que corresponde a uma estratégia para a geração de conjuntos de termos que são mais significativos. Para armazenar a informação sobre a proximidade entre os termos, a estrutura das listas invertidas é estendida da seguinte forma: Para cada par termo-documento $[i, d_j]$, nós adicionamos a lista de posições de um determinado termo i em um documento d_j , representada por $rp_{i,j}$, onde a posição de um termo i corresponde ao número de termos que o precedem em um documento d_j . Desta forma, cada entrada em uma lista invertida de um determinado termo i torna-se uma tripla $\langle d_j, tf_{i,j}, rp_{i,j} \rangle$.

A informação de proximidade é utilizada como uma estratégia de poda que limita os conjuntos de termos aos formados por termos que ocorrem próximos entre si. Essa definição captura a noção de que termos semanticamente relacionados freqüentemente aparecem próximos em um documento. Verificar o requisito de proximidade é muito simples e consiste em rejeitar os conjuntos de termos que contém termos cuja distância é maior do que um limite dado, chamado de proximidade mínima.

Conjuntos Fechados de Termos

Cada conjunto de termos carrega informação semântica relacionada à correlação entre os termos. Entretanto, como algumas correlações podem sobrepor outras, o uso de todas as correlações podem introduzir ruído no modelo e por conseqüência reduzir a efetividade de recuperação. Os conjuntos de termos freqüentemente se sobrepõem porque se um conjunto de termos é freqüente, por definição, todos os seus subconjuntos também são freqüentes. Se a sobreposição entre os conjuntos for total, ou seja, os documentos nos quais cada um dos conjuntos de termos ocorrem forem os mesmos, podemos avaliar somente o conjunto com o maior número de termos, sem que ocorra perda de informação. Os conjuntos fechados de termos correspondem a esses conjuntos de termos. A seguir apresentamos sua definição formal.

O fechamento de um conjunto de termos S_i corresponde ao conjunto de todos os conjuntos freqüentes de termos que co-ocorrem com S_i no mesmo conjunto de documentos e que preservam o requisito de proximidade. Um conjunto fechado de termos CS_i corresponde ao maior conjunto de termos no fechamento do conjunto de termos S_i .

Os conjuntos fechados de termos permitem que os conjuntos freqüentes de termos que não agregam qualquer informação adicional de valor sejam automaticamente descartados. Esses conjuntos são interessantes porque representam uma redução na complexidade computacional e na quantidade de dados que precisa ser analisada pelos algoritmos de ordenação de documentos, sem perda de informação.

A determinação de conjuntos fechados de termos é uma extensão do problema de mineração de conjuntos freqüentes de termos. Nossa abordagem é baseada em um eficiente algoritmo chamado *CHARM* (Zaki, 2000). Nós adaptamos esse algoritmo para tratar termos e documentos em vez de itens e transações, respectivamente.

Conjuntos Maximais de Termos

Um conjunto maximal de termos MS_i corresponde a um conjunto freqüente de termos que não é subconjunto de nenhum outro conjunto freqüente de termos. Os conjuntos maximais de termos permitem que os conjuntos freqüentes de termos que não agregam nenhuma nova informação de correlação, ou seja, os conjuntos freqüentes de termos que são subconjuntos de qualquer conjunto maximal de termos, sejam automaticamente descartados.

Esses conjuntos são interessantes, assim como os conjuntos fechados de termos, porque representam uma redução ainda maior na complexidade computacional e na quantidade de dados que precisa ser analisada pelos algoritmos de ordenação de documentos, e podem ser utilizados quando padrões de co-ocorrência mais específicos são necessários.

A determinação de conjuntos maximais de termos também é uma extensão do problema de mineração de conjuntos freqüentes de termos. Nossa abordagem é baseada em um eficiente algoritmo chamado *GENMAX* (Gouda and Zaki, 2001). Nós adaptamos esse algoritmo para tratar termos e documentos em vez de itens e transações, respectivamente.

Seja FT o conjunto de todos os conjuntos freqüente de termos, CFT o conjunto de todos os conjuntos fechados de termos, e MFT o conjunto de todos os conjuntos maximais de termos. $MFT \subseteq CFT \subseteq FT \subseteq V$ corresponde à relação de cardinalidade entre esses conjuntos. O conjunto MFT é bem menor do que o conjunto CFT , que é bem menor que o conjunto FT , que é bem menor do que V . O conjunto MFT corresponde a menor quantidade de informação necessária para gerar todos os conjuntos freqüentes de termos para uma coleção de documentos (Gouda and Zaki, 2001).

Modelo Baseado em Conjuntos

A seguir, as características fundamentais do modelo proposto, chamado de modelo de espaço vetorial baseado em conjuntos, ou simplesmente modelo baseado em conjuntos, são apresentadas.

Representação de Documentos e Consultas

As consultas e os documentos ainda são representados por vetores, assim como no modelo de espaço vetorial original. Entretanto, os componentes desses vetores não são mais termos, mas sim conjuntos de termos, os *termsets*. Formalmente:

$$\begin{aligned}\vec{d}_j &= (w_{S_1,j}, w_{S_2,j}, \dots, w_{S_{2^t},j}) \\ \vec{q} &= (w_{S_1,q}, w_{S_2,q}, \dots, w_{S_{2^t},q})\end{aligned}$$

onde t corresponde ao número de termos distintos na coleção de documentos, $w_{S_i,j}$ corresponde ao peso de um termset S_i em um documento d_j , e $w_{S_i,q}$ corresponde ao peso de um termset S_i em uma consulta q .

Uma importante simplificação em nosso modelo é que o espaço vetorial é induzido somente para os termsets enumerados a partir dos termos presentes nas consultas. Documentos e consultas são representados por vetores em um espaço vetorial de 2^t dimensões, onde t corresponde ao número de termos distintos no vocabulário T . Entretanto, somente as dimensões correspondentes aos conjuntos de termos enumerados a partir dos termos da consulta são levadas em consideração.

Esquema de Ponderação dos *Termsets*

No modelo baseado em conjuntos, os pesos são associados aos termsets (em vez dos termos). Esses pesos são função do número de ocorrências de um termset em um documento e em toda a coleção de documentos, de forma equivalente ao esquema de ponderação de termos $tf \times idf$. Qualquer esquema de ponderação baseado nestas medidas podem ser facilmente adaptados para o modelo baseado em conjuntos, especialmente os esquemas de ponderação utilizados pelo modelo de espaço vetorial padrão e pelos modelos probabilísticos baseados em relevância. Nos experimentos deste trabalho utilizamos dois esquemas de ponderação de termsets: o primeiro baseado no esquema de ponderação largamente utilizado pelas implementações do modelo de espaço vetorial (Salton and Yang, 1973; Yu and Salton, 1976; Salton and Buckley, 1988) e o segundo baseado no esquema BM25 (Robertson et al., 1995) utilizado pelo modelos probabilísticos.

Algoritmo de Ordenação

No modelo baseado em conjuntos, computamos a similaridade entre um documento e uma consulta como o produto escalar normalizado entre o vetor que representa o documento $\vec{d}_j$, $1 \leq j \leq N$, e o vetor que representa a consulta do usuário $\vec{q}$, da seguinte forma:

$$sim(q, d_j) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{S_i \in S_q} w_{S_i,j} \times w_{S_i,q}}{|\vec{d}_j| \times |\vec{q}|},$$

onde $w_{S_i,j}$ corresponde ao peso de um termset S_i em um documento d_j , $w_{S_i,q}$ corresponde ao peso de um termset S_i em uma consulta q , e S_q corresponde ao conjunto de todos os conjuntos de termos gerados a partir da consulta q .

A normalização (ou seja, os fatores no denominador) é feita usando-se somente os 1-termsets, ou seja, os termos que compõem a consulta e os documentos. Essa simplificação reduz consideravelmente o custo computacional, pois o cálculo da norma dos documentos baseado em todos os *termsets* implicaria na geração de todos os *termsets* para a coleção. Apesar dessa simplificação, a normalização continua válida, uma vez que o objetivo de penalizar documentos grandes ainda persiste. Nossos resultados experimentais confirmam a validade dessa simplificação no cálculo da similaridade entre documentos e consultas.

Para ordenar os documentos a partir de uma consulta q , nós utilizamos o seguinte algoritmo. Primeiro cria-se uma estrutura para o armazenamento dos valores (acumuladores) A para as similaridades parciais dos documentos, calculadas para cada conjunto de termos em um documento d_j . Depois, para cada termo na consulta q , recupera-se sua lista invertida e determina-se os conjuntos freqüentes de termos de tamanho 1, aplicando o limite de freqüência mínima mf . O próximo passo é a enumeração de todos os conjuntos de termos baseados no limite de freqüência mínima e proximidade mínima. Depois de enumerar todos os conjuntos de termos, nós computamos a similaridade parcial para cada conjunto de termo S_i em relação ao documento d_j , utilizando um dos dois esquemas de ponderação discutidos anteriormente. A seguir, nós normalizamos as similaridades A , dividindo cada similaridade A_j pela norma do documento d_j correspondente. O passo final consiste em selecionar os k maiores valores para os acumuladores e retornar os documentos correspondentes.

Aplicações do Modelo Baseado em Conjuntos

Neste trabalho apresentamos um estudo detalhado da aplicação do modelo baseado em conjuntos, utilizando os conjuntos fechados de termos, para o processamento de consultas disjuntivas, conjuntivas e por frases. Uma das principais vantagens do modelo de espaço vetorial corresponde a sua estratégia de recuperação parcial, que permite a recuperação de documentos que aproximam as condições das consultas. Essa estratégia corresponde, conceitualmente, ao processamento de consultas disjuntivas. Diferentes máquinas de busca e grandes portais podem apresentar semânticas distintas para o processamento padrão de consultas formadas por mais de um termo. Entretanto, a maioria das maiores máquinas de busca assumem o comportamento conjuntivo para as suas consultas, ou seja, todos os termos da consulta devem aparecer em um documento para que esse seja incluído na ordenação final. Uma fração considerável das consultas submetidas na Web são compostas por frases, ou seja, uma seqüência de termos marcados com aspas duplas, que significa que a frase pesquisada deve aparecer em um documento para que ele sejam incluído na ordenação final.

Apresentamos também uma nova técnica para a estruturação automática de consultas baseada na distribuição dos vários componentes conjuntivos de uma determinada consulta em uma coleção de documentos. Os conjuntos maximais de termos são utilizados para modelar os componentes conjuntivos de uma consulta. O processamento dos conjuntos maximais de termos pelo modelo baseado em conjuntos promove a transformação automática de uma consulta conjuntiva em uma consulta disjuntiva, cujos componentes conjuntivos passam a ser “conceitos” com suporte na coleção de documentos. Essa estruturação é especialmente útil em substituição a consultas conjuntivas complexas, ou que não retornam um resultado aceitável.

Resultados Experimentais

Para a avaliação do modelo proposto, utilizamos cinco coleções de referência: CFC, CISI, TREC-8, WBR-99 e WBR-04. Cada coleção de referência possui um conjunto de consultas e, para cada consulta, os documentos relevantes (selecionados por especialistas) são indicados.

As medidas padrão de revocação (*recall*) e precisão (*precision*) foram utilizadas para a comparação do desempenho da efetividade de recuperação dos modelos avaliados. A eficiência computacional foi avaliada através dos tempos médios de resposta para o conjunto de consultas de cada coleção.

As consultas associadas às coleções avaliadas foram divididas em dois grupos. O primeiro, caracterizado como conjunto de treinamento, é formado por 15 consultas escolhidas de forma aleatória. Esse conjunto foi utilizado para determinar os melhores valores de frequência mínima e proximidade mínima para cada uma das coleções. Ele foi utilizado também para a avaliação e escolha da técnica de normalização a ser utilizada nos outros experimentos. O segundo grupo, formado pelas consultas restantes, foi utilizado para a comparação do modelo proposto, tanto para o processamento de consultas disjuntivas, conjuntivas e por frases, quanto para a abordagem de estruturação automática de consultas. Todos os experimentos foram executados em um PC com processador AMD-athlon 2600+ com 512 MBytes de memória principal com o sistema operacional Linux.

Efetividade de Recuperação

A Tabela 1 apresenta os resultados obtidos a partir de uma comparação de desempenho da efetividade de recuperação, através da precisão média, dos modelos avaliados para o processamento de consultas disjuntivas utilizando-se as coleções de referência CFC, CISI, TREC-8 e WBR-99. O modelo de espaço vetorial generalizado (GVSM), uma extensão do modelo de espaço vetorial que leva em consideração a correlação entre os termos, não pôde

ser avaliado para as coleções TREC-8 e WBR-99 devido ao seu custo exponencial no número de termos do vocabulário. Podemos observar que o modelo baseado em conjuntos (SBM) e o modelo baseado em conjuntos com informação de proximidade (PSBM) apresentam resultados melhores que o modelo de espaço vetorial (VSM) independentemente da coleção utilizada. Os ganhos variam de 2.36% a 20.78% para o modelo baseado em conjuntos, e de 10.38% a 25.86% para o modelo baseado em conjuntos com informação de proximidade. Podemos perceber que os ganhos apresentados para a coleção WBR-99 são menores. Isto ocorreu porque o número médio de termos por consulta é aproximadamente 2, o que limita o processamento das correlações entre os termos. O modelo baseado em conjuntos, com e sem a informação de proximidade, também apresenta resultados melhores do que o modelo de espaço vetorial generalizado, o que mostra que as correlações entre os termos podem ser utilizadas com sucesso para melhorar as qualidades das respostas.

Consultas Disjuntivas							
Coleção	Precisão Média (%)				Ganho (%)		
	VSM	GVSM	SBM	PSBM	GVSM	SBM	PSBM
CFC	27.37	29.05	33.06	34.45	6.13	20.78	25.86
CISI	17.31	17.40	20.18	21.20	0.51	16.58	22.47
TREC-8	25.44	-	29.17	31.76	-	14.66	24.84
WBR-99	24.85	-	25.44	37.43	-	2.36	10.38

Tabela 1: Precisão média dos modelos avaliados para as coleções de referência CFC, CISI, TREC-8 e WBR-99 para o processamento de consultas disjuntivas.

A seguir apresentamos os resultados obtidos para o processamento de consultas conjuntivas. Esses resultados são apresentados na Tabela 2. Podemos observar que o modelo baseado em conjuntos (SBM) e o modelo baseado em conjuntos com proximidade (PSBM) apresentam resultados melhores que o modelo de espaço vetorial (VSM) independentemente da coleção utilizada. Os ganhos variam de 7.29% a 16.38% para o modelo baseado em conjuntos, e de 11.04% a 29.96% para o modelo baseado em conjuntos com proximidade. Podemos perceber que os ganhos apresentados para a coleção TREC-8 são maiores. Isto ocorre devido ao maior número de termos por consulta, que permite que o nosso modelo compute um conjunto mais representativo de conjuntos fechados de termos.

Agora discutimos os resultados obtidos para o processamento de consultas por frases. Neste caso, o modelo baseado em conjuntos com proximidade corresponde ao modelo baseado em conjuntos, uma vez que, por definição, o limite de proximidade mínima deve ser igual a 1. Como podemos ver na Tabela 3, o modelo baseado em conjuntos (SBM) apresenta um resultado superior ao modelo de espaço vetorial (VSM), com ganhos de 8.93% para a coleção WBR-99 e de 17.51% para a coleção TREC-8.

Consultas Conjuntivas					
Coleção	Precisão Média (%)			Ganho (%)	
	<i>VSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>SBM</i>	<i>PSBM</i>
TREC-8	19.96	23.23	25.94	16.38	29.96
WBR-99	33.60	36.05	37.31	7.29	11.04

Tabela 2: Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-99 para o processamento de consultas conjuntivas.

Consultas por Frases			
Coleção	Precisão Média (%)		Ganho (%)
	<i>VSM</i>	<i>SBM</i>	
TREC-8	11.59	13.62	17.51
WBR-99	15.11	16.46	8.93

Tabela 3: Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-99 para o processamento de frases.

Finalmente apresentamos os resultados obtidos para o processamento de consultas estruturadas. A Tabela 4 apresenta os valores de precisão média para o modelo de espaço vetorial (VSM), para o modelo probabilístico (BM25), para o modelo baseado em conjuntos (SBM) utilizando o mesmo esquema de ponderação do modelo probabilístico, e o modelo baseado em conjuntos utilizando os conjuntos maximais de termos (SBM-MAX). Para a coleção TREC-8, o modelo baseado em conjuntos provê um bom ganho de 27.44% em relação ao modelo de espaço vetorial e de 15.86% sobre o modelo probabilístico, enquanto que nossa abordagem para estruturação automática de consultas aumentou esses ganhos para 34.71% e 22.47%, respectivamente. Para a coleção WBR-04, enquanto o modelo baseado em conjuntos provê um ganho de 29.48% em relação ao modelo de espaço vetorial e de 10.90% sobre o modelo probabilístico, nossa abordagem para estruturação automática de consultas provê ganhos de 41.42% e 21.13%, respectivamente.

Consultas Estruturadas							
Coleção	Precisão Média (%)				Ganho (%)		
	<i>VSM</i>	<i>BM25</i>	<i>SBM</i>	<i>SBM-MAX</i>	<i>VSM</i>	<i>BM25</i>	<i>SBM</i>
TREC-8	19.96	21.95	25.44	26.89	34.71	22.47	05.69
WBR-04	20.18	23.56	26.13	28.54	41.42	21.13	09.22

Tabela 4: Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-04 para o processamento de consultas estruturadas.

O modelo baseado em conjuntos é o primeiro modelo de recuperação de informação que utiliza correlações entre termos de forma eficiente e que produz melhoria consistente na qualidade das respostas, independentemente da coleção de referência utilizada e do tipo de consulta processado, além de prover um mecanismo eficiente e efetivo para a estruturação automática de consultas.

Desempenho Computacional

Nesta seção comparamos o modelo baseado em conjuntos com o modelo de espaço vetorial a partir dos tempos de resposta para cada uma das consultas, com o objetivo de avaliar sua viabilidade em termos de recursos computacionais. Uma das principais limitações dos modelos existentes que levam em consideração a correlação entre os termos está relacionada com a grande demanda de recursos computacionais. Muitos desses modelos não podem ser utilizados para coleções de documentos de tamanho médio a grande. A adição da determinação dos conjuntos de termos e o respectivo cálculo de similaridade para esses conjuntos de termos não afetam significativamente o tempo de execução das consultas.

O acréscimo médio no tempo total de execução das consultas no modelo baseado em conjuntos ficou entre 19.3% e 58.4% para consultas disjuntivas, entre 21.0% e 33.3% para consultas conjuntivas, entre 18.1% e 22.5% para consultas por frases e entre 9.9% e 21.0% para consultas estruturadas automaticamente. Esses resultados mostram a viabilidade prática do modelo baseado em conjuntos.

Conclusões e Trabalhos Futuros

Apresentamos um novo modelo para a recuperação de informação em coleções de textos que leva em consideração as correlações entre os termos através de sua ocorrência simultânea nos documentos. Mostramos que o novo modelo sempre apresenta resultados melhores mantendo um custo computacional adicional aceitável. A determinação das correlações entre os termos pela utilização dos conjuntos de termos enumerados pelos algoritmos para a geração de regras de associação permite uma extensão direta dos sistemas de recuperação de informação baseados no modelo de espaço vetorial.

Demonstramos, através das curvas de revocação e precisão, o aumento na efetividade de recuperação do modelo proposto em comparação com o modelo de espaço vetorial original para todas as coleções de referência avaliadas e todos os tipos de consultas considerados. Mostramos também que nosso mecanismo de estruturação automática de consultas é bastante efetivo. A viabilidade em termos computacionais, medidos através do tempo de resposta para as consultas, permite a utilização do modelo proposto em ambientes cujo tamanho das coleções de textos sejam maiores.

O desenvolvimento deste trabalho deixou ainda algumas questões em aberto, citadas aqui como sugestões para trabalhos futuros. Em primeiro lugar, a proximidade entre os termos pode também ser usada para melhorar a qualidade da nossa abordagem para a estruturação automática de consultas. Em segundo lugar, podemos utilizar o arcabouço dos conjuntos de termos em outros modelos de recuperação de informação, como os modelos probabilísticos e os modelos baseados em linguagens estatísticas. Finalmente, podemos apresentar uma fundamentação teórica para o modelo proposto utilizando a Teoria da Informação.

**SET-BASED VECTOR MODEL: A NEW APPROACH
FOR CORRELATION-BASED RANKING**

Contents

1	Introduction	1
1.1	Information Retrieval	1
1.2	Data Mining	2
1.3	Thesis Related Work	3
1.4	Thesis Contributions	6
1.5	Thesis Outline	7
2	Classical Information Retrieval Models	9
2.1	Boolean Models	10
2.2	Vector Space Models	11
2.2.1	Standard Vector Space Model	11
2.2.2	Generalized Vector Space Model	13
2.3	Probabilistic Models	14
2.3.1	Probabilistic Relevance Models	14
2.3.2	Inference-Based Models	15
2.3.3	Statistical Language Models	16
2.4	Set Oriented Models	18
2.5	Bibliography Revision	19
2.6	Summary	20
3	Modeling Correlation Among Terms	21
3.1	Termsets	21
3.2	Proximate Termsets	24
3.3	Termset Rules	24
3.4	Closed Termsets	26
3.5	Maximal Termsets	28
3.6	Bibliography Revision	31
3.7	Summary	32

4	Set-Based Vector Model	33
4.1	Documents and Queries Representations	33
4.2	Termset Weighting Schema	34
4.3	Ranking Computation	36
4.4	Computational Complexity	37
4.5	Indexing Data Structures and Algorithm	38
4.6	Set-Based Model Applications	39
4.6.1	Query Processing	39
4.6.2	Query Structuring	41
4.7	Set-Based Model Expressiveness	43
4.8	Summary	44
5	Experimental Results	45
5.1	Experimental Setup	45
5.1.1	Evaluation Metrics	45
5.1.2	The Reference Collections	46
5.2	Tuning of the Set-Based Model	50
5.2.1	Minimal Frequency Evaluation	50
5.2.2	Minimal Proximity Evaluation	53
5.2.3	Normalization Evaluation	56
5.3	Retrieval Effectiveness	62
5.3.1	Query Processing	62
5.3.2	Query Structuring	71
5.4	Computational Performance	77
5.4.1	Query Processing	77
5.4.2	Query Structuring	81
5.5	Bibliography Revision	82
5.6	Summary	83
6	Conclusions and Future Work	85
6.1	Thesis Summary	85
6.2	Future Work	86
	Bibliography	91

List of Figures

2.1	Vector space representation for the Example 1.	12
3.1	Sample document collection.	22
3.2	Frequent and closed termsets for the sample document collection of Example 1 for all valid minimal frequency values.	28
3.3	Frequent, closed, and maximal termsets for the sample document collection of Example 1 for all valid minimal frequency values.	30
4.1	Vector space representation for the Example 10.	34
4.2	The set-based model ranking algorithm.	37
4.3	The inverted file index structure.	38
4.4	The set-based model work-flow.	39
4.5	Information retrieval models expressiveness.	43
5.1	Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CFC test collection.	51
5.2	Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CISI test collection.	51
5.3	Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM) and the standard vector space model (VSM), in the TREC-8 test collection.	52
5.4	Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM) and the standard vector space model (VSM), in the WBR-99 test collection.	52
5.5	Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the maximal set-based model (SBM-MAX), the proba- bilistic model (BM25), and the standard vector space model (VSM) in the WBR- 04 test collection.	53

5.6	Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CFC test collection.	54
5.7	Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CISI test collection.	54
5.8	Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM) and the standard vector space model (VSM), in the TREC-8 test collection.	55
5.9	Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM) and the standard vector space model (VSM), in the WBR-99 test collection.	55
5.10	Normalization recall-precision curves for the CFC collection using a training set of 15 queries.	57
5.11	Normalization recall-precision curves for the CISI collection using a training set of 15 queries.	58
5.12	Normalization recall-precision curves for the TREC-8 collection using a training set of 15 queries.	59
5.13	Normalization recall-precision curves for the WBR-99 collection using a training set of 15 queries.	60
5.14	Normalization recall-precision curves for the WBR-04 collection using a training set of 15 queries.	61
5.15	Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the CFC test collection, using the test set of sample queries.	63
5.16	Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the CISI test collection, using the test set of sample queries.	63
5.17	Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the TREC-8 test collection, using the test set of sample queries.	64

5.18	Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the WBR-99 test collection, using the test set of sample queries.	64
5.19	Precision-recall curves for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) when conjunctive queries are used, with the TREC-8 test collection, using the test set of sample queries. .	69
5.20	Precision-recall curves for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) when conjunctive queries are used, with the WBR-99 test collection, using the test set of sample queries. .	69
5.21	Precision-recall curves for the vector space model (VSM) and the set-based model (SBM) when phrase queries are used, with the TREC-8 test collection, using the test set of sample queries	72
5.22	Precision-recall curves for the vector space model (VSM) and the set-based model (SBM) when phrase queries are used, with the WBR-99 test collection, using the test set of sample queries	72
5.23	Precision-recall curves for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structures queries are used, with the TREC-8 test collection, using the test set of sample queries.	75
5.24	Precision-recall curves for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structures queries are used, with the WBR-04 test collection, using the test set of sample queries.	75
5.25	Impact of query size on average response time in the WBR-99 for the set-based model (SBM).	79
5.26	Query size distribution for the WBR-99.	80

List of Tables

1	Precisão média dos modelos avaliados para as coleções de referência CFC, CISI, TREC-8 e WBR-99 para o processamento de consultas disjuntivas.	xviii
2	Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-99 para o processamento de consultas conjuntivas.	xix
3	Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-99 para o processamento de frases.	xix
4	Precisão média dos modelos avaliados para as coleções de referência TREC-8 e WBR-04 para o processamento de consultas estruturadas.	xix
3.1	Vocabulary-set for the query $q = \{a, b, c, d, f\}$	23
3.2	Examples of termset rules	25
3.3	Frequent and closed termsets for the sample document collection of Example 1.	27
3.4	Frequent, closed, and maximal termsets for the sample document collection of Example 1.	29
5.1	Characteristics of the five reference collections.	47
5.2	CFC document level average figures for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.	66
5.3	CISI document level average figures for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.	66
5.4	TREC-8 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.	67
5.5	WBR-99 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.	67

5.6	Comparison of average precision of the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.	68
5.7	TREC-8 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries.	70
5.8	WBR-99 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries.	70
5.9	Comparison of average precision of the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.	71
5.10	Document level average figures for the vector space model (VSM) and the set-based model (SBM) relative to the TREC-8 test collection, when phrase queries are used.	73
5.11	Document level average figures for the vector space model (VSM) and the set-based model (SBM) relative to the WBR-99 test collection, when phrase queries are used.	73
5.12	Comparison of average precision of the vector space model (VSM) and the set-based model (SBM) with phrase queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.	74
5.13	TREC-8 document level average figures for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structured queries are used.	76
5.14	WBR-04 document level average figures for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structured queries are used.	76

5.15	Comparison of average precision of the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) with structured queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.	77
5.16	Average number of closed termsets and inverted list sizes for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM).	78
5.17	Average response times and response time increases for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) for disjunctive query processing. . .	78
5.18	Average response times and response time increases for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) for conjunctive query processing.	80
5.19	Average response times and response time increases for the vector space model (VSM) and the set-based model (SBM) for phrase query processing.	81
5.20	Average response times and response time increases for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) with the TREC-8 and the WBR-04 test collections.	81
5.21	Average number of termsets for the set-based model (SBM) and the maximal set-based model (SBM-MAX) with the TREC-8 and the WBR-04 reference collections.	82

Chapter 1

Introduction

The field of data mining and information retrieval has been explored together in the last years. However, association rules mining, a well-known data mining technique, was not directly used to improve the retrieval effectiveness of information retrieval systems. This work concerns the use of association rules as a basis for the definition of a new information retrieval model that accounts for correlations among index terms. In this chapter, we develop and discuss the goals and contributions of our thesis.

1.1 Information Retrieval

Information Retrieval (IR) focuses on providing users with access to information stored digitally. Unlike data retrieval, which studies solutions for the efficient storage and retrieval of structured data, information retrieval is concerned with the extraction of information from non-structured or semi-structured text data. We can interpret the information retrieval problem as composed of three main parts: the user, the information retrieval system, and a digital data repository composed of the documents in a collection. The user has an *information need* that he/she translates to the information retrieval system as a *query*. Given a user's query, the goal of the information retrieval system is to retrieve from the data repository the documents that satisfy the user's information need, i.e., documents that are *relevant* to the user. Usually, this task consists of retrieving a set of documents and *ranking* them according to the likeliness that they will satisfy the user's query.

Traditionally, information retrieval was concerned with documents composed only of text. User queries were sets of keywords. Finding documents likely to satisfy a user's need consisted, basically, of finding documents that contained the words in the specified user's query. Several information retrieval models were proposed based on this general principle (Baeza-Yates and Ribeiro-Neto, 1999).

The most popular models for ranking the documents of a collection (not necessarily a Web document collection) are (i) the vector space models (Salton and Lesk, 1968; Salton, 1971), (ii) the probabilistic relevance models (Maron and Kuhns, 1960; van Rijsbergen, 1979; Robertson and Jones, 1976; Robertson and Walker, 1994), and (iii) the statistical language models (Ponte and Croft, 1998; Berger and Lafferty, 1999; Lafferty and Zhai, 2001). The differences between these models rely on the representation of queries and documents, on the schemes for term weighting, and on the formula for computing the ranking.

Designing effective schemes for term weighting is a critical step in a search system if improved ranking is to be obtained. However, finding good term weights is an ongoing challenge. In this work we propose a new term weighting schema that leads to improved ranking and is efficient enough to be practical.

The best known term weighting schemes use weights that are function of the number of times the index term occurs in a document and the number of documents in which the index term occurs. Such term weighting strategies are called $tf \times idf$ (term frequency times inverse document frequency) schemes (Salton and McGill, 1983; Witten et al., 1999; Baeza-Yates and Ribeiro-Neto, 1999). A modern variation of these strategies is the BM25 weighting scheme used by the Okapi system (Robertson and Walker, 1994; Robertson et al., 1995).

All practical term weighting schemes, to this date, assume that the terms are mutually independent — an assumption often made for mathematical convenience and simplicity of implementation. However, it is generally accepted that exploitation of the correlation among index terms in a document might be used to improve retrieval effectiveness with general collections. In fact, distinct approaches that take term co-occurrences into account have been proposed over time (Wong et al., 1985, 1987; Rijsbergen, 1977; Harper and Rijsbergen, 1978; Raghavan and Yu, 1979; Billhardt et al., 2002; Nallapati and Allan, 2002; Cao et al., 2004). However, after decades of research, it is well-known that taking advantage of index term correlations for improving the final document ranking is not a simple task. All these approaches suffer from a common drawback, they are too inefficient computationally to be of value in practice.

1.2 Data Mining

Data Mining and Knowledge Discovery in Databases (KDD) is a new interdisciplinary field merging ideas from statistics, machine learning, databases, and parallel computing. It has been engendered by the phenomenal growth of data in all spheres of human endeavor, and the economic and scientific need to extract useful information from the collected data. The key challenge in data mining is the extraction of knowledge from massive databases.

Data mining refers to the overall process of discovering new patterns or building models from a given dataset. There are many steps involved in the KDD enterprise which include data selection, data cleaning and preprocessing, data transformation and reduction, data-mining task and algorithm selection, and finally post-processing and interpretation of discovered knowledge (Fayyad et al., 1996b,a). This KDD process tends to be highly iterative and interactive.

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT) (Feldman and Dagan, 1995; Feldman and Hirsh, 1997), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining combines techniques of information extraction, information retrieval, natural language processing and document summarization with the methods of data mining. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value.

One of the most well-known and successful techniques of data mining and text mining is the association rules. The problem of mining association rules in categorical data presented in customer transactions was introduced by Agrawal et al. (1993b). This seminal work gave birth to several investigation efforts (Agrawal and Srikant, 1994; Park et al., 1995; Agrawal et al., 1996; Bayardo et al., 1999; Veloso et al., 2002; Srikant and Agrawal, 1996; Zhang et al., 1997; Pôssas et al., 2000) resulting in descriptions of how to extend the original concepts and how to increase the performance of the related algorithms.

The original problem of mining association rules was formulated as how to find rules of the form $set_1 \rightarrow set_2$. This rule is supposed to denote affinity or correlation among the two sets containing nominal or ordinal data items. More specifically, such association rule should translate the following meaning: customers that buy the products in set_1 also buy the products in set_2 . Statistical basis is represented in the form of minimum support and minimum confidence measures of these rules with respect to the set of overall customer transactions.

1.3 Thesis Related Work

As we shall see later on, the set-based vector model is the first information retrieval model that exploits term correlations and term proximity effectively and provides significant gains in terms of precision, regardless of the size of the collection, of the size of the vocabulary, and the query type. All known approaches that account for correlation among index terms were initially designed for processing only disjunctive queries. The set-based vector model provides a simple, effective, efficient, and parameterized way to process disjunctive, conjunctive, and phrase queries. Our approach was also used for automatically structuring a

user query into a disjunction of smaller conjunctive subqueries. Following we review some seminal works related to the use of correlation patterns in information retrieval models, and several query structuring mechanisms.

Correlation-Based Information Retrieval Models

Different approaches to account for co-occurrence among index terms have been proposed. The use of statistical analysis of a set of queries (considering relevant and non-relevant document sets) to establish positive and negative correlations among index terms was proposed by Raghavan and Yu (1979). The work by Rijsbergen (1977) introduces a probabilistic model that incorporates dependences among index terms. Experimental results were later presented in a companion paper by Harper and Rijsbergen (1978). The extent to which two index terms depend on one another is derived from the distribution of co-occurrences in the whole collection, in the relevant document set, and in the non-relevant document set, leading to a non-linear weighting function. As showed in Salton et al. (1982), the resulting formula for computing the dependency factors developed by Rijsbergen (1977); Harper and Rijsbergen (1978) does not seem to be computationally feasible, even for a relatively small number of index terms. The work in Bollmann-Sdorra and Raghavan (1998) presents a study of term dependence in a query space.

The work in Wong et al. (1985, 1987) presents an interesting approach to compute index term correlations based on automatic indexing schemes. It defines a new information retrieval model called *generalized vector space model*. The work shows that index term vectors can be explicitly represented in a 2^t -dimensional vector space, where t is the vocabulary size, such that index term correlations can be incorporated into the vector space model in a straightforward manner. It represents index term vectors using a basis of orthogonal vectors called min-terms, where each min-term represents one of the 2^t possible patterns of index term co-occurrence inside documents. Each index term k_i is represented by a term composed of all min-terms related to k_i . The model is not computationally feasible for moderately large collections because there are 2^t possible min-terms. Extensions of the generalized vector space model were presented in Alsaffar et al. (2000); Kim et al. (2000).

Billhardt et al. (2002) present a context vector model that uses term dependencies in the process of indexing documents and queries. The context vectors that represent the documents of a collection provide richer descriptions of their basic characteristics. The similarity calculation is based on a semantic-matching rather than on a simple word-matching approach.

Language modeling approaches to information retrieval usually do not capture correlation between terms. However, there have been attempts to represent correlation among index terms using *bigrams* or *bi-terms* (Song and Croft, 1999; Srikanth and Srihari, 2002). In these latter works, only adjacent words are assumed to be related. Nallapati and Allan (2002) and

Cao et al. (2004) present alternative language models that allow representing term correlations. These correlations are bounded by a document sentence, such that only the strongest word dependencies are considered in order to reduce estimation errors. Cao et al. (2005) proposed another dependency language model in which two types of word relationships are taken into account, one extracted from the WordNet ¹ and other based on term by term co-occurrence patterns.

Bookstein (1988) proposed a decision theoretic framework to outline a set oriented model for the information retrieval systems. It argued that accepting a set oriented viewpoint might enhance retrieval effectiveness, because structural relations, or correlation patterns, occurring within a collection could be used to broken down it into meaningful subsets of related documents. In spite of its theoretical appeal, the set oriented model was not properly instantiated and evaluated through experimentation. However, it clear defines the bounds for all correlation-based approaches.

Query Structuring Mechanisms

Structured queries containing operators such as AND, OR, and proximity can be used to describe accurate representations of information needs. Although boolean query languages may be difficult for people to use, there is considerable evidence that trained users may achieve good retrieval effectiveness using them. Experimental results with the extended boolean model (Salton et al., 1983) and the network model (Turtle and Croft, 1990, 1991) showed that structured queries were more effective than simpler queries consisting of a set of weighted terms.

There are studies on the best translation of linguistic relationships from queries into boolean operators. Using both syntactic and semantic information, Das-Gupta (1987) proposed an algorithm for deciding when the natural language conjunction “and” should be interpreted as a boolean AND or a boolean OR. Smith (1990) presented a complex algorithm for translating a full syntactic parse of a natural language query into a boolean form.

The aforementioned studies all focused on the connections between linguistic relationships and boolean operators. Natural language processing models view a query as an expression of different concepts and their relationships that are of interest to the user. Correctly identifying these concepts in queries and in documents results in improved retrieval performance (Srikanth and Srihari, 2003).

Phrases provide another means for structured queries to capture linguistic relationships, specially for natural language processing approaches. Both statistical (based on word co-occurrences) and syntactic (based on natural language processing techniques) methods have been successfully explored to identify indexing phrases. Mitra et al. (1997) compared sta-

¹<http://wordnet.princeton.edu/>

tistical and syntactic indexing phrases and observed a trade off between query accuracy and query coverage. Croft et al. (1991) used phrases identified in natural language queries to build structured queries for a probabilistic model.

Instead of processing documents through a natural language processing system to identify phrases for indexing, there have been efforts to use linguistic processing to get a better understanding of the user information needs. Experiments by Smeaton and van Rijsbergen (1988) implement a retrieval strategy that is based on syntactic analysis of queries. The work by Narita and Ogawa (2000) has examined the utility of phrases as search terms in information retrieval. They used single term selection and phrasal term selection in their query construction. Similar to Mitra et al. (1997), they experimented with different representations for multi-word phrases (more than 2 words) and decided to use two word phrases for query construction.

1.4 Thesis Contributions

This thesis focuses in the application of a data mining technique in the information retrieval domain to increase retrieval effectiveness. This work intends to provide answers to the following research questions:

- Is the exploitation of the correlation among index terms effective to improve retrieval precision for general document collections, including Web collections?
- Is there a practical and efficient mechanism, in terms of computational costs, that account for the correlations among index terms?

To answer the questions and also to overcome the standard vector space model problems and limitations, we propose a new model for computing index term weights that takes into account patterns of term co-occurrence and is efficient enough to be of practical value. Our model is referred to as *set-based vector model*. For simplicity, we also refer to it as set-based model. We evaluated and validated the set-based model through experiments using several test collections. The major contributions of this thesis are, therefore:

- An information retrieval model to compute term weights (set-based model), which is based on the set-theory, derived from association rules mining. We showed that it is possible to significantly improve retrieval effectiveness, while keeping extra computational costs small (Chapters 4 and 5).
- The use of term weighting schemes based on association rules theory. Association rules naturally provide for quantification of representative patterns of index term co-occurrences, something that is not present in other term weighting schemes, such as the $tf \times idf$ and the *BM25* schemes (Chapters 3 and 4).

- The formal framework we adopted allowed naturally to consider relevant patterns of term co-occurrence accounting for the information about the proximity among query terms in documents. In addition to assessing document relevance, the proximity information was successfully used in identifying phrases with a greater degree of precision (Chapter 3).
- The application of the set-based model provides a simple, effective, efficient, and parameterized way to process disjunctive, conjunctive, phrase, and automatically structured queries. All known approaches that account for correlation among index terms were initially designed for processing only disjunctive queries (Chapter 4).
- A detailed empirical evaluation of the set-based model for all query types considered in terms of retrieval and computational performance. The evaluation is based on a comparison with the standard vector space model, with the generalized vector space model, and with the BM25 probabilistic relevance model (Chapter 5).

Partial results have been published in Pôssas et al. (2002c,a,b, 2004, 2005c,a,b).

1.5 Thesis Outline

This work is organized as follows. Chapter 2 gives some background on information retrieval models, specially the models that we have used in our experiments, notably the vector space model, the probabilistic model with the BM25 weighting scheme, and the generalized vector space model. Chapter 3 discusses our method for computing co-occurrences among query terms based on a variant of association rules. The concept of termsets is introduced as a basis for computing term weights, where termset is simply an ordered set of terms extracted from the documents. In Chapter 4, the basic features of the set-based model are developed and justified. We also present two different types of applications that uses the set-based model. We show how different query types, such as conjunctive, disjunctive and phrase queries, can be modeled using the proposed model, and how to automatically structure user queries. Chapter 5 describes the reference collections and the evaluation metrics we used during experimentation. The tuning of the set-based model is also described, followed by experiments on retrieval effectiveness and computational performance. Finally, in Chapter 6 we draw our conclusions and suggestions for future research.

Chapter 2

Classical Information Retrieval Models

Research in information retrieval is based on several quite different paradigms. It is important to understand the foundations of the principal approaches in order to develop a more thorough appreciation of the relative strengths and weaknesses of the different models. The history of information retrieval research has showed that the development of models is often a combination of some theoretical modeling and a lot of experimentation guided by intuition and/or experience. This has the unfortunate result that not all of the motivations for the development of a term-weighting formula have been well-documented. In many cases, information is scattered over many different papers, sometimes with inconsistent notation. Therefore we will describe the intuitions of several important information retrieval models in some more detail, notably the models that we have used for our information retrieval experiments: the vector space model, the probabilistic model with the BM25 weighting scheme, and the generalized vector space model. For a complete description and taxonomy of the information retrieval models, see Baeza-Yates and Ribeiro-Neto (1999).

The classic models in information retrieval consider that each document is described by a set of representative keywords called *index terms*. An *index term* is simply a (document) word whose semantics helps in remembering the document's main themes. Thus, index terms are mainly used to index and summarize the documents content. However, all the distinct words in a document are considered as index terms, specially for Web search engines. Given a set of index terms, we notice that not all terms are equally useful for describing the document contents. The importance of term for a document is captured through the assignment of numerical *weights*.

This chapter provides the necessary theoretical background material which serves as a starting point for our work which is presented in later chapters.

2.1 Boolean Models

The earliest information retrieval systems were Boolean systems. Even today, a lot of commercial information retrieval systems are based on the Boolean model. The popularity among users is largely based on the clear set-theoretic semantics of the model. In a Boolean system, documents are represented by a set of index terms. An index term is seen as a propositional constant. If the index term occurs in the document, it is true for the document, and following the closed world assumption, it is false if the index term does not occur in the document. Queries consist of logical combinations of index terms using AND, OR or NOT and braces. Thus a query is a propositional formula. Every propositional formula can be rewritten as a disjunctive normal form which can be efficiently evaluated for each document. The ranking function is thus a binary decision rule: if the formula holds for a document, it is considered relevant and retrieved. The Boolean retrieval model is very powerful, since in theory a query could be constructed which only retrieves the relevant documents, provided that each document is indexed by a unique set of index terms. However, without knowledge of the document collection it is impossible for a user to create such a query.

The conceptual clarity of Boolean systems is important for users. They know exactly how a query is evaluated, because the resulting documents will satisfy the Boolean constraint of the query. This gives the user a feeling of tight control of the retrieval function. However, Boolean systems also have considerable disadvantages: (i) Since documents are modeled as either relevant or non-relevant, retrieved documents are not ordered with respect to relevance and documents that contain most query terms are not retrieved. (ii) It is difficult for users to compose good queries. As a result, the retrieved set is often too large or completely empty. (iii) The model does not support query term weighting or relevance feedback. (iv) Boolean systems display inferior retrieval effectiveness on standard information retrieval test collections.

The extended Boolean model (Salton and McGill, 1983) integrates term-weighting and distance measures into the Boolean model. Firstly, index terms can be weighted between 0 and 1. Secondly, the Boolean connectives have a new semantics, they are modeled as similarity measures based on non-Euclidean distances in a t -dimensional space, where t is equal to the number of different index terms in the document collection. The extended Boolean model has been further generalized in the p -norm model. Here the semantics of the OR and AND connective contains a parameter p . By varying the parameter p between 1 and infinity, the p -norm ranking function varies between a vector space model like ranking and a Boolean ranking function. In principle p can be set for every connective.

Despite their conceptual appeal, extended Boolean models have not become popular. One of the reasons could be that the models are less perspicuous for the user. Queries still have the form of a Boolean formula, but with changed semantics. Many users prefer not to

spend a lot of time to compose a structured query. For long queries, a vector space or probabilistic system is to be preferred. For two-word queries a Boolean AND query is usually, but not always, sufficient. Extended Boolean systems in combination with sophisticated user interfaces which give feedback on term statistics might be attractive especially for a more robust handling of short queries.

2.2 Vector Space Models

In this section we will discuss the standard vector space model and its assumptions. Following, we will present a more advanced vector based model, called generalized vector space model, which takes into account correlation among index terms.

2.2.1 Standard Vector Space Model

The notion of similarity which is characteristic for the vector space model (VSM) approach. In contrast with the Boolean model where the matching function is based on an exact match, the vector space approach starts from a more fine grained view on relevance estimation.

In the vector space model the algebraic representation of the set of index terms for both documents and queries correspond to vectors in a t -dimensional Euclidean space, where t is equal to the number of different index terms in the document collection (Salton and Lesk, 1968). A vector space model based system determines the similarity between a query representation and a document taking a vector distance measure as similarity metric and thus as relevance predictor. The similarity is assumed to be correlated with the probability of relevance of the document.

Example 1 Consider a vocabulary of three terms $T = \{k_1, k_2, k_3\}$, and a collection C of two documents d_j , $1 \leq j \leq 2$, given by $C = \{(k_1, k_2, k_3, k_1, k_2, k_3, k_2, k_3, k_2, k_3, k_3), (k_1, k_2, k_3, k_1, k_2, k_1, k_2, k_2, k_2, k_2)\}$, and a user query $q = \{k_1\}$. Figure 2.1 shows the vector space defined for the documents of the collection C and the specified user query q . For simplicity, we are considering the weights of a terms in a document, as its number of occurrences.

The term weights can be calculated in many different ways (Salton and Yang, 1973; Yu and Salton, 1976; Sparck, 1972). The best known term weighting schemes for the vector space model use weights that are given by (i) $tf_{i,j}$, the number of times that an index term i occurs in a document d_j and (ii) df_i , the number of documents that an index term i occurs in the whole document collection. Thus, the weight of an index term i in a document d_j is given by:

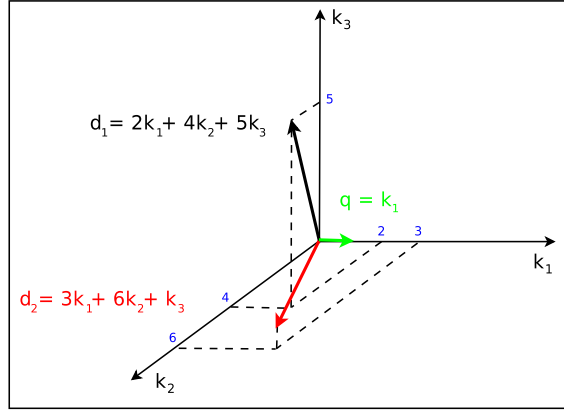


Figure 2.1: Vector space representation for the Example 1.

$$w_{i,j} = tf_{i,j} \times idf_i = tf_{i,j} \times \log \frac{N}{df_i}, \quad (2.1)$$

where N corresponds to the number of documents in the collection and idf_i corresponds to the inverse document frequency for term i . Such term-weighting strategy is called $tf \times idf$ (term frequency times inverse document frequency) scheme.

Similarly, the weight of a term i in a query q is formally defined as:

$$w_{i,q} = f(tf_{i,q}) \times idf_i = (1 + \log tf_{i,q}) \times \log \left(1 + \frac{N}{df_i}\right) \quad (2.2)$$

where N is the number of documents in the collection, $tf_{i,q}$ is the number of occurrences of the term i in the query q and idf_i is the inverse frequency of occurrence of the term i in the collection, scaled down by a log function.

One of the most successful ranking formula for the vector space model is the cosine measure. It assigns a similarity measure to every document containing any of the query terms, defined as the scalar product between the set of document vectors $\vec{d}_j$, $1 \leq j \leq N$, and the query vector $\vec{q}$. This measure is equivalent to the angle between the query vector and any document vector. Thus, the similarity between a document d_j and a query q is given by:

$$sim(q, d_j) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}}, \quad (2.3)$$

where $w_{i,q}$ corresponds to the weight of term i in query q , whose definition is equivalent to the weight of a term in a document, i.e., $w_{i,q} = tf_{i,q} \times idf_i$. The factors $|\vec{d}_j|$ and $|\vec{q}|$ correspond to the norm of document and query vectors, respectively. The ranking calculation is not affected by $|\vec{q}|$ because its value is the same for all documents. The factor $|\vec{d}_j|$ represents the length of document d_j .

In spite of its success, the vector space model has the disadvantage that index terms are assumed to be mutually independent, an assumption often made as a matter of mathematical convenience and simplicity of implementation. This is clearly a simplification because occurrences of index terms in a document are not independent.

2.2.2 Generalized Vector Space Model

The main idea of generalized vector space model (GVSM) (Wong et al., 1985, 1987) is to incorporate index term correlations, represented as elements of the Boolean algebra, into a vector space. In this mapping, terms are represented as a linear combination of vectors associated with atomic expressions (or concepts) that are pairwise orthogonal.

The index term vectors can be explicitly represented in a 2^t -dimensional vector space, where t is the vocabulary size, such that index term correlations can be incorporated into the vector space model in a straightforward manner. It represents index term vectors using a basis of orthogonal vectors called min-terms.

Definition 1 A *min-term* is an atomic expression $\vec{m}_r$, that represents one of the 2^t possible patterns of index term co-occurrence inside documents. Each index term vector $\vec{k}_i$ is composed of all min-terms that contains the index term k_i .

Pairwise orthogonality among the generated min-terms vectors does not imply independence among the index terms. On the contrary, index terms correlations arise from the min-terms vectors, i.e., common patterns of term co-occurrence are shared among the min-terms.

The ranking formula for the generalized vector space model is also the cosine measure. It assigns a similarity measure to every document containing any of the query terms, defined as the scalar product between the set of document vectors $\vec{d}_j$, $1 \leq j \leq N$, and the query vector $\vec{q}$. Formally, the similarity between a document d_j and a query q is given by:

$$\text{sim}(q, d_j) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^t \sum_{r=1}^{2^t} c_{i,r}^j \times c_{i,r}^q}{\sqrt{\sum_{i=1}^t \sum_{r=1}^{2^t} (c_{i,r}^j)^2} \times \sqrt{\sum_{i=1}^t \sum_{r=1}^{2^t} (c_{i,r}^q)^2}}, \quad (2.4)$$

where $c_{i,r}^j$ corresponds to the sum of the weights of all terms k_i contained in a document d_j for each min-term m_r . Analogously, $c_{i,r}^q$ corresponds to the sum of the weights of all terms k_i contained in a query q for each min-term m_r . The weight of a term k_i in a document or query is the same used by the standard vector space model, presented in Eq. 2.1. The factors $|\vec{d}_j|$ and $|\vec{q}|$ correspond to the norm of document and query vectors, respectively.

The generalized vector space model is more complex than the standard vector space model, and is not computationally feasible for moderately large collections because there are 2^t possible min-terms. Further, it is not clear whether this model yields effective improvement in retrieval effectiveness for general collections (Baeza-Yates and Ribeiro-Neto, 1999). Despite these drawbacks, its main contribution relies in its theoretical point of view.

2.3 Probabilistic Models

In the previous section we have seen that term statistics can serve as an effective means to weight the importance of a term. However, the specific term weighting schemes of the vector space model have a rather heuristic basis. Probability theory has proved to be a more principled avenue to deal with uncertainty. The (classical) probabilistic takes the relevance relation as starting point, and uses term statistics for the estimation of parameters in the model. We will discuss three classes of probabilistic models in the following sections: (i) Probabilistic relevance models try to estimate the relevance of a document directly based on the idea that query terms have different distributions in relevant and non-relevant documents. (ii) Inference based models apply Bayesian inference for the computation of a relevance score. (iii) Generative probabilistic models, also called statistical language models as usually applied in automatic speech recognition systems, can also very fruitfully be applied for information retrieval.

2.3.1 Probabilistic Relevance Models

The probabilistic model (Robertson and Jones, 1976) estimates the probability that the user will find a document d_j relevant for a user query q . The model assumes that this probability of relevance depends on the query and the document representations only. Further, it assumes that there is a subset of all documents that the user prefers as the answer set for the query q . Such an *ideal* answer set is labeled R and should maximize the overall probability of relevance to user. Documents in the set R are predicted to be relevant to the query, while documents not in this set are predicted to be non-relevant.

The BM25 measure (Robertson et al., 1995) is one of the most successful ranking formula for the probabilistic model. The BM25 weighting scheme is a function of the number of occurrences of the term in a document, in the whole collection, and a function of the document length. Formally, the weight of a term i in a document d_j is defined as:

$$w_{i,j} = \frac{k_1 \times tf_{i,j}}{tf_{i,j} + k_1 \times \left(1 - b + b \times \frac{|\vec{d_j}|}{|\vec{d_j}|}\right)} \times \log \frac{N - df_i + 0.5}{0.5}, \quad (2.5)$$

where $tf_{i,j}$ is the number of occurrences of the term i in the document d_j , k_1 and b are parameters, that depends on the collection and possibly on the nature of the user queries, $|\vec{d_j}|$ corresponds to a document length function, $|\vec{d_j}|$ is the average document length, N is the number of documents in the collection, and df_i is the number of documents containing the term i .

The BM25 scheme also defines a weight of a term i in a query q . This weight is formally defined as:

$$w_{i,q} = \frac{(k_3 + 1) \times tf_{i,q}}{k_3 + tf_{i,q}}, \quad (2.6)$$

where $tf_{i,q}$ is the number of occurrences of the term i in the query q , and k_3 is a parameter, that depends on the collection and possibly on the nature of the user queries.

The probabilistic model computes the similarity between a document and the user query as the scalar product between the document vector $\vec{d}_j$, $1 \leq j \leq N$, and the query vector $\vec{q}$, as follows:

$$sim(q, d_j) = \vec{d}_j \bullet \vec{q} = \sum_{i \in q} w^{(1)} \times w_{i,j} \times w_{i,q} + k_2 \times \frac{|\vec{d}_j| - |\vec{q}|}{|\vec{d}_j| + |\vec{q}|}, \quad (2.7)$$

where $w_{i,j}$ is the weight associated with the term i in the document d_j , $w_{i,q}$ is the weight associated with the term i in the query q , $|\vec{d}_j|$ corresponds to a document length function, $|\vec{d}_j|$ is the average document length, k_2 is another parameter, that also depends on the collection and possibly on the nature of the user queries, and $w^{(1)}$ is the Robertson-Sparck Jones weight (Robertson and Jones, 1976), which is defined as:

$$w^{(1)} = \log \frac{(r + 0.5) / (R - r + 0.5)}{(df_i - r + 0.5) / (N - df_i - R + r + 0.5)}, \quad (2.8)$$

where N is the number of documents in the collection, df_i is the number of documents containing the term i , R is the number of relevant documents for the query q , and r is the number of relevant documents containing the term i .

2.3.2 Inference-Based Models

Bayesian Networks (also known as Inference or Belief Networks) are a modeling language within which many probabilistic relationships can be expressed as part of a common representation, and used as part of a unified inference procedure (Pearl, 1988). A Bayesian Network is a graph in which nodes correspond to propositions and links correspond to conditional probabilistic dependencies between these propositions. A directed link from node p to node q is used to model the fact that p causes q , although other semantics (e.g. logical implication) are sometimes also used.

When representing interactions among n propositions, we must in general consider the possible dependency of each proposition on every other. To do this completely requires an exponential number of statistics which is impractical for most situations, and certainly if we attempt to model interactions between all the documents in our corpora and their keywords. Within Bayesian Networks this full set of statistics, the joint probability distribution,

is replaced by a sparse representation only among those variables directly influencing one another. Interactions among indirectly-related variables are then computed by propagating inference through a graph of these direct connections.

The key integration of probabilistic information across interacting variables is accomplished by specifying how each child node depends on the set of its parent's values. A table of conditional dependency probabilities specifies, for each possible value of each parent node, the probability of each of child variable's value. With these conditional relationships specified for each node, querying a Bayesian network corresponds to placing prior probabilities on some elements of the network, and then asking for the probability at other nodes.

The first application of Bayesian Network representations to information retrieval problems was presented by Turtle and Croft (1990, 1991). In the *inference network model*, index terms, documents and user queries are seen as events and are represented as nodes in a Bayesian network. The model takes the viewpoint that the observation of a document induces belief on its set of index terms, and that specification of such terms induces belief in a user query or information need. This model was showed to perform better than traditional probabilistic models and used to effectively combine different sources of information for the task of document ranking. The sources of information are not limited to the query formulation, but can also include knowledge about the user, the domain etc...

Later, a second information retrieval model, called *belief network model*, was proposed by Ribeiro-Neto and Muntz (1996), where the elements of an information retrieval system are formally defined as concepts in a sample space. Their work not only provides a probabilistic justification for the model, but also demonstrates that the combination of evidence from past queries with evidence from the vector space model yields better results than the use of a vector ranking alone.

2.3.3 Statistical Language Models

A statistical language model is a probability distribution over all possible sentences or other linguistic units in a language. It can also be viewed as a statistical model for generating text. The task of language modeling, in general, answers the question: how likely the i^{th} word in a sequence would occur given the identities of the preceding $(i - 1)$ words? In most applications of language modeling, such as speech recognition and information retrieval, the probability of a sentence is decomposed into a product of n -gram probabilities.

Definition 2 Let $S = \{k_1, k_2, \dots, k_t\}$ be a specified sequence of t words. An n -gram language model considers the word sequence S to be a Markov process with probability:

$$P_n(S) = \prod_{i=1}^t P(k_i | k_{i-1}, k_{i-2}, k_{i-3}, \dots, k_{i-n+1}), \quad (2.9)$$

where n refers to the order of the Markov process.

When $n = 2$ we call it a *bigram* language model which is estimated using information about the co-occurrence of pairs of words. In the case of $n = 1$, we call it a unigram language model which uses only estimates of the probabilities of individual words. For applications such as speech recognition or machine translation, word order is important and higher-order (usually *trigram*) models are used. In information retrieval, the role of word order is less clear and *unigram* models have been used extensively.

To establish the word n -gram language model, probability estimates are typically from frequencies of n -gram patterns in the training data. It is common that many possible word n -gram patterns would not appear in the actual data used for estimation, even if the size of the data is huge and the value of n is small. As a consequence, for rare or unseen events the likelihood estimates that are directly based on counts become problematic. This is often referred to as the data sparseness problem. Smoothing is used to address this problem and has been an important part in any language model. A discussion of this problem is not beyond the scope of this work, but we give some useful references to it in Section 2.5.

The basic approach for using language models for information retrieval assumes that the user has a reasonable idea of the terms that are likely to appear in the “ideal” document that can satisfy his/her information need, and that the query terms the user chooses can distinguish the “ideal” document from the rest of the collection (Ponte and Croft, 1998). The query is thus generated as the piece of text representative of the “ideal” document. The task of the system is then to estimate, for each of the documents in the collection, which is most likely to be the ideal document. That is, we calculate:

$$P(d_j|q) = P(q|d_j) P(d_j), \quad (2.10)$$

where q is a query and d_j is a document. The prior probability $P(d_j)$ is usually assumed to be uniform and a language model $P(q|d_j)$ is estimated for every document. In other words, we estimate a probability distribution over words for each document and calculate the probability that the query is a sample from that distribution. Documents are ranked according to this probability. This is generally referred to as the *query-likelihood retrieval model* and was first proposed by Ponte and Croft (1998).

This work takes a multi-variate Bernoulli approach to approximate $P(q|d_j)$. There are two main assumptions behind this approach: First, a query q is represented as a vector of binary attributes, one for each unique term in the vocabulary, indicating its presence or absence. The number of times that each term occurs in the query is not captured. Second, the occurrence of each term in a document is considered independently. Based on these assumptions, the query likelihood $P(q|d_j)$ is thus formulated as the product of the probability of producing the query terms and the probability of not producing other terms. Formally:

$$P(q|d_j) = \prod_{i=1}^{|q|} P(k_i|d_j) \prod_{i=1}^{|q|} (1.0 - P(k_i|d_j)), \quad (2.11)$$

where $P(k_i|d_j)$ is calculated by a non-parametric method that makes use of the average probability of k_i in documents containing it and a risk factor. For non-occurring terms, the global probability of k_i in the collection is used instead. It is worth mentioning that collection statistics such as term frequency and document frequency are integral parts of the language model and not used heuristically as in traditional probabilistic and other approaches. In addition, document length normalization does not have to be done in an ad hoc manner as it is implicit in the calculation of the probabilities. This approach to retrieval, although very simple, has demonstrated superior effectiveness to traditional vector space and probabilistic models (Ponte and Croft, 1998).

2.4 Set Oriented Models

Formalization of the information retrieval process were usually based on Boolean logic and were explicitly set oriented. However, for various practical and theoretical reasons, most recent probabilistic-based attempts to model information retrieval and to develop retrieval algorithms have tended to be single-item oriented. Bookstein (1988) proposed a decision theoretic framework to outline a set oriented model for the information retrieval systems. It argued that accepting a set oriented viewpoint might enhance retrieval effectiveness, because structural relations occurring within a collection could be used to broken down it into meaningful subsets of related documents. Also, since item oriented retrieval is a special case of set oriented retrieval it is expected that a set oriented approach could allow us to better understand the functioning of the current models.

The set oriented model corresponds to a theoretical framework for representing concepts or co-occurrence patterns that can be used to retrieve any subset of the documents of the collection. The constraint on the number of possible sets that can be retrieved, which is identical to the number of possible independent Boolean requests, corresponds to the expressiveness of this approach. Since there are 2^{2^t} independent Boolean expressions that can be formed from t distinct index terms, and a set of N documents can be broken up into 2^N subsets, a fundamental constraint for every subset of documents to be retrieval is that 2^{2^t} be at least as large as 2^N , or that t be greater than $\log_2 N$. This formal constraint defines a lower bound on the number of index terms needed to retrieve every set of documents. Moreover, correlation among index terms in documents and in the queries relax this constraint, since they limit the number of possible Boolean requests, because not all co-occurrence patterns are valid.

In spite of its theoretical appeal, the set oriented model was not properly instantiated and evaluated through experimentation. However, it clearly defines the bounds for all correlation based approaches, including the set-based vector model.

2.5 Bibliography Revision

The vector space model was proposed by Salton (Salton and Lesk, 1968; Salton, 1971), and different weighting schemes were presented in the following works (Salton and Yang, 1973; Yu and Salton, 1976; Salton and Buckley, 1988). In the vector space model, index terms are assumed to be mutually independent. The independence assumption leads to a linear weighting function which, although not necessarily realistic, is easy to compute. Simple term weighting was used early on by Salton and Lesk (1968). Spark Jones introduced the *idf* factor (Sparck, 1972), and Salton and Yang verified its effectiveness for improving retrieval (Salton and Yang, 1973).

The first explicit probabilistic model was due to Maron and Kuhns (1960). While it is concerned with probability of relevance, it starts from the opposite end: user queries are assumed fixed, but document indexing requires optimization. No real experiments have ever been done with this model. An attempt has been made to unify the Maron/Kuhns model with the classic probabilistic model proposed by Robertson and Jones (1976). Robertson et al. (1982) suggests that this unification gives the possibility of using relevance feedback both locally (for the immediate query) and globally (to modify the document indexing for subsequent queries). However, this model has not been evaluated experimentally.

Following the original Robertson/Sparck Jones model, with its assumption of independence of terms, a substantial amount of work was done on formal models which made some attempt to avoid or relax such assumptions (Rijsbergen, 1977; Harper and Rijsbergen, 1978; Raghavan and Yu, 1979). These models were evaluated, but accounting for dependence among index terms did not lead to any substantial improvements in retrieval effectiveness.

Statistical Language Modeling has been used in different Natural Language Processing tasks including Speech Recognition (Jelinek, 1998), Machine Translation (Brown et al., 1990), Information Extraction (Srihari et al., 1999), and, finally, Information Retrieval (Ponte and Croft, 1998). Different researchers have showed that the language modeling approach to information retrieval using smoothed unigram language models performs better than the vector space and classical probabilistic retrieval models (Berger and Lafferty, 1999; Hiemstra, 1998; Miller et al., 1999). Improved retrieval has been demonstrated using higher order language models, e.g. bigram or trigram language models (Miller et al., 1999; Song and Croft, 1999). While these higher order models are derived on term co-occurrence or statistical basis, more linguistic information that capture the user's information need can be encoded in the language models.

Alternative approaches have been proposed to the Query-Likelihood method for ranking documents Ponte and Croft (1998): Document-likelihood or Relevance Models, a language model is associated with the query or topic of interest and documents are ranked based on the probability of them being generated by the query language model (Lavrenko and Croft, 2001) and Query-Document Model Similarity based on Risk Minimization Framework, documents are ranked based on the similarity between language models associated with the query and a document (Lafferty and Zhai, 2001).

2.6 Summary

This chapter has discussed the main elements and the intuition related to several classical information retrieval models: (i) the Boolean based models, (ii) the vector space based models, (iii) the probabilistic based models, and (iv) the set oriented models. These models provide the theoretical foundations of our work and will be used to evaluate and validate the results found for the set-based model, which will be presented later. We also showed a detailed bibliographic discussion for each of the discussed models.

Chapter 3

Modeling Correlation Among Terms

In this chapter we introduce the concept of termsets as a basis for modeling dependences among index terms in the set-based model. We also present three special types of termsets: proximate, closed, and maximal termsets.

One of the key features of our approach is that we only compute the set of termsets associated with the query terms. The generalized vector space model (Wong et al., 1985, 1987), on the other hand, requires the computation of weights for all subsets of correlated terms in the document space, which is hard to compute with large collections. As we shall see, the set-based model computation becomes simpler and faster.

3.1 Termsets

Definition 3 Let $T = \{k_1, k_2, \dots, k_t\}$ be the vocabulary of a collection C of N documents, that is, the set of t unique terms that appear in all documents in C . There is a total ordering among the vocabulary terms, which is based on the lexicographical order of terms, so that $k_i < k_{i+1}$, for $1 \leq i \leq t - 1$.

Definition 4 An n -termset S , $S \subseteq T$, is a set of n terms. When the number of terms is not important, we refer simply to the termset S .

Definition 5 Let $V = \{S_1, S_2, \dots, S_{2^t}\}$ be the vocabulary-set of a collection C of documents, that is, the set of 2^t unique termsets that may appear in any document from C . The frequency dS_i of a termset S_i is the number of occurrences of S_i in C , that is, the number of documents where $S_i \subseteq d_j$ and $d_j \in C$, $1 \leq j \leq N$.

Definition 6 With each termset S_i , $1 \leq i \leq 2^t$, we associate an inverted list lS_i composed of the identifiers of the documents containing that termset. The frequency dS_i of a termset S_i can be computed as the length of its associated inverted list ($|lS_i|$). Further, we also use lS_i to refer to the set of documents in the list.

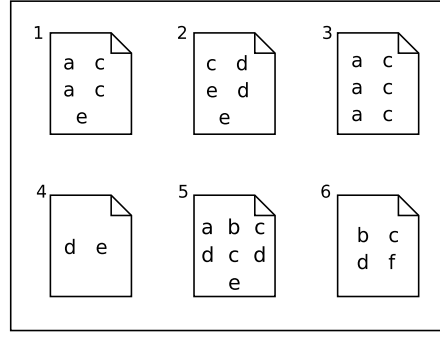


Figure 3.1: Sample document collection.

Definition 7 A termset S_i is a frequent termset if its frequency dS_i is greater than or equal to a given threshold, known as **support** in the scope of association rules (Agrawal et al., 1993b) and referred to as **minimal frequency** in this work. As presented in the original Apriori algorithm (Agrawal and Srikant, 1994), if an n -termset is frequent, all of its $(n - 1)$ -termsets are also frequent.

Example 2 Consider a vocabulary of six terms $T = \{a, b, c, d, e, f\}$, and a collection C of six documents d_j , $1 \leq j \leq 6$, given by $C = \{(a, c, a, c, e), (c, d, e, d, e), (a, c, a, c, a, c), (d, e), (a, b, c, d, c, d, e), (b, c, d, f)\}$, as depicted in Figure 3.1. Consider also the user query $q = \{a, b, c, d, f\}$. There are 32 termsets associated with q , but only 23 occur in our sample collection, as depicted in Table 3.1. The 1-termsets are S_a, S_b, S_c, S_d , and S_f , the 2-termsets are $S_{ab}, S_{ac}, S_{ad}, S_{bc}, S_{bd}, S_{bf}, S_{cd}, S_{cf}$, and S_{df} , the 3-termsets are $S_{abc}, S_{abd}, S_{acd}, S_{bcd}, S_{bcf}, S_{bdf}$, and S_{cdf} , and the two 4-termsets are S_{abcd} and $S_{bcd f}$.

Generating Termsets

Our procedure for generating termsets is an adaptation of a well-known algorithm for determining termsets (Agrawal and Srikant, 1994). As mentioned, the main challenge in determining the frequent termsets is that the number of termsets increases exponentially with the number of distinct terms of the query, making naive or exhaustive approaches infeasible.

To search for frequent termsets, we use a simple and powerful principle: for an n -termset to be frequent, all $(n - 1)$ -termsets that are subsets of it must be frequent. Several of the most efficient data mining algorithms for association rules are based on this principle. They start by verifying which single terms are frequent and then combine them into 2-termsets. With each 2-termset is associated an inverted list of documents which is used to determine whether the 2-termset is frequent or not. The process iterates for termsets of size 3 and up, until there are no more frequent termsets to be found.

Termsets	Elements	Documents
S_a	$\{a\}$	$\{d_1, d_3, d_5\}$
S_b	$\{b\}$	$\{d_5, d_6\}$
S_c	$\{c\}$	$\{d_1, d_2, d_3, d_5, d_6\}$
S_d	$\{d\}$	$\{d_2, d_4, d_5, d_6\}$
S_f	$\{f\}$	$\{d_6\}$
S_{ab}	$\{a, b\}$	$\{d_5\}$
S_{ac}	$\{a, c\}$	$\{d_1, d_3, d_5\}$
S_{ad}	$\{a, d\}$	$\{d_5\}$
S_{bc}	$\{b, c\}$	$\{d_5, d_6\}$
S_{bd}	$\{b, d\}$	$\{d_5, d_6\}$
S_{bf}	$\{b, f\}$	$\{d_6\}$
S_{cd}	$\{c, d\}$	$\{d_2, d_5, d_6\}$
S_{cf}	$\{c, f\}$	$\{d_6\}$
S_{df}	$\{d, f\}$	$\{d_6\}$
S_{abc}	$\{a, b, c\}$	$\{d_5\}$
S_{abd}	$\{a, b, d\}$	$\{d_5\}$
S_{acd}	$\{a, c, d\}$	$\{d_5\}$
S_{bcd}	$\{b, c, d\}$	$\{d_5, d_6\}$
S_{bcf}	$\{b, c, f\}$	$\{d_6\}$
S_{bdf}	$\{b, d, f\}$	$\{d_6\}$
S_{cdf}	$\{c, d, f\}$	$\{d_6\}$
S_{abcd}	$\{a, b, c, d\}$	$\{d_5\}$
S_{bcdf}	$\{b, c, d, f\}$	$\{d_6\}$

Table 3.1: Vocabulary-set for the query $q = \{a, b, c, d, f\}$.

To determine whether a termset is frequent or not, a three-steps procedure is executed: (i) verify whether its subsets are frequent; if so, (ii) generate its inverted list and count its size; and (iii) check whether its inverted list size is above the minimum frequency. As expected, the most expensive task is the second one, which is implemented as an intersection of the inverted lists of the $(n - 1)$ -termsets that are subsets of the termset being generated.

Example 3 Consider our example document collection in Figure 3.1 and a minimum frequency equal to 2. To determine whether S_{bcd} is frequent, we first check whether S_{bc} , S_{bd} , and S_{cd} are frequent. Since they are indeed frequent, we generate lS_{bcd} by intersecting the lists for S_{bc} , S_{bd} , and S_{cd} . The resulting list contains the documents $\{d_5, d_6\}$. We can conclude that S_{bcd} is frequent, because its frequency is greater than or equal to the minimum frequency.

3.2 Proximate Termsets

We extend the concept of termsets to consider the proximity among the terms in the documents, as a strategy for generating termsets that are more meaningful. To store information on proximity among terms in a document, we extend the structure of the inverted lists as follows. For each term-document pair $[i, d_j]$, we add a list of occurrence locations of the term i in the document d_j , represented by $rp_{i,j}$, where the location of a term i is equal to the number of terms that precede i in document j . Thus, each entry in the inverted list for term i becomes a triple $\langle d_j, tf_{i,j}, rp_{i,j} \rangle$.

To compute proximate termsets, we modify the algorithm for computing termsets by adding a new constraint: two terms are considered close when their distance is bounded by a proximity threshold, called minimum proximity. This technique is equivalent to the concept of intra-document passages (Zobel et al., 1995; Kaszkeil and Zobel, 1997; Kaszkeil et al., 1999).

Proximity information works as a pruning strategy that limits termsets to those formed by proximate terms. This captures the notion that semantically related terms often occur close to each other. Verifying the proximity constraint is quite straightforward and consists of rejecting the termsets that contain terms whose distance is larger than the given threshold.

Example 4 *To illustrate how proximity affects the determination of termsets, consider the termsets S_a and S_c in Example 1 and a minimum proximity threshold of 1. To verify whether S_{ac} is frequent, it is necessary to consider the proximity of the occurrences of a and c . Terms a and c co-occur in documents d_1 , d_3 , and d_5 . We then calculate $rp_{a,1} = \{1, 3\}$, $rp_{c,1} = \{2, 4\}$, $rp_{a,3} = \{1, 3, 5\}$, $rp_{c,3} = \{2, 4, 6\}$, $rp_{a,5} = \{1\}$, and $rp_{c,5} = \{3, 5\}$. Following, we verify for each document whether the occurrences of the termsets S_a and S_c are within the proximity threshold. This is the case for documents 1 and 3, but not for document 5. Thus, the frequency of S_{ac} is set to 2. Clearly, the application of this new criterion tends to reduce the total number of termsets. Most important, the termsets that are computed represent stronger correlations, which tends to improve the retrieval effectiveness. Our experimental results (see Section 5.3) confirm such observations.*

3.3 Termset Rules

Each termset embeds semantic information on term correlations. However, since some correlations may subsume others, the use of all of them may introduce noise into the model and reduce the retrieval precision. Termsets frequently overlap because a frequent termset implies that its subsets are also frequent. In Example 1, termset S_{ac} is a subset of termset S_{abc} , both are frequent, and there is an overlap in the set of documents in which they occur,

Rules	Confidence (%)
$S_a \rightarrow S_{ab}$	33
$S_{ab} \rightarrow S_{abc}$	100
$S_{ac} \rightarrow S_{ab}$	33
$S_{abcd} \rightarrow S_{bcdf}$	0

Table 3.2: Examples of termset rules

more specifically $\{d_5\}$. One issue in this case is whether both termsets should be considered for retrieving information, since discarding one of them may result in information loss. We distinguish two scenarios where information loss may occur. First, if we discard S_{abc} , we lose information on the correlation among the terms a , b , and c . Second, if we discard S_{ac} , we also lose information on a correlation that is “popular” (it occurs in 3 documents) and thus, more meaningful for retrieval purposes.

In summary, whenever two termsets overlap and one termset is a subset of the other, discarding the larger termset results in losing correlation information. Discarding the smaller termset results in losing popularity information. To better understand this information loss process, we introduce the use of “rules”, which are good for identifying precedence relations.

Definition 8 *In the context of termsets, a rule is an implication $X \rightarrow Y$, where X and Y are termsets. The rule is characterized by a confidence degree, which is the probability that Y appears in a document given that X has appeared.*

Example 5 *To illustrate, consider the rules presented in Table 3.2. Discarding either of the termsets that compose the first rule will result in information loss. However, discarding S_{ab} in the second rule, while keeping S_{abc} , will not result in any loss because the information carried by S_{abc} is exactly the same information carried by S_{ab} . This discarding strategy reduces the number of termsets to be considered while yielding better retrieval results. The third rule confirms the intuition that both termsets S_{ac} and S_{ab} cannot be discarded without information loss. Finally, the last rule indicates that S_{abcd} and S_{bcdf} do not share any information and can not be discarded.*

Whenever a termset rule has 100% confidence, the “smaller” termset may be discarded without information loss. This can be accomplished by enumerating all termset rules and then selecting those with 100% confidence. But, since enumerating all termset rules is expensive, it is necessary to devise a strategy for selecting termsets to be discarded. On the other hand, whenever a termset rule has 0% confidence, it makes no sense to discard either of the associated termsets, since the information they carry is mutually exclusive.

As we will be seen, closed termsets automatically identify the 100% confidence rules, while maximal termsets identify the 0% confidence rules. Closed and maximal termsets also define the limits of the spectrum of sets of termsets. Each point in the spectrum is characterized by a minimum confidence that is satisfied by the rules among all termsets that should be enumerated from a user query. Notice that going beyond this spectrum in any direction does not make sense. Considering less than the maximal termsets will result in clear information loss, while taking more than the closed termsets result in clear information redundancy and distortion. This tradeoff will be investigated later on in this work.

3.4 Closed Termsets

In this section we introduce the idea of closed termsets, an extension to the concept of frequent termsets.

Definition 9 The closure of a termset S_i is the set of all frequent termsets that co-occur with S_i in the same set of documents and that preserve the proximity constraint.

Definition 10 The closed termset CS_i is the largest termset in the closure of a termset S_i . More formally, given a set $D \subseteq C$ of documents and a set S_D of termsets that occur in all documents from D and only in these, a closed termset CS_i satisfies the property that $\nexists S_j \in S_D | (CS_i \subset S_j \wedge lS_i \equiv lS_j)$.

Closed termsets allow one to automatically discard termsets that do not aggregate any additional information of value. In fact, closed termsets encapsulate termsets that are the consequent in 100%-confidence rules. Closed termsets are interesting because they represent a reduction in the computational complexity and in the amount of data that has to be analyzed for ranking purposes, without loss of information.

Example 6 Consider the dataset of Example 1. Table 3.3 shows all frequent and closed termsets and their respective frequencies. If we define that a frequent termset must have a minimum frequency of 50%, the number of termsets is reduced from 23 to 5. Notice that the number of frequent termsets, although potentially very large, is usually small in natural language texts. Regarding the closed termsets, even in this small example, we see that the number of closed termsets (7) is considerably smaller than the number of frequent termsets (23), for a minimum frequency of 17%.

A major advantage of using closed termsets, instead of frequent termsets, is that they can be generated very efficiently. Since the number of closed termsets is at most equal to the number of frequent termsets, we may also use this bound as an upper limit to the number of closed termsets. As discussed in Section 5.4, in practical situations, the number of closed termsets is significantly smaller than the number of frequent termsets.

Frequency (ds)	Frequent Termsets	Closed Termsets
83% (5)	S_c	S_c
67% (4)	S_d	S_d
50% (3)	S_a, S_{ac}	S_{ac}
50% (3)	S_{cd}	S_{cd}
33% (2)	$S_b, S_{bc}, S_{bd}, S_{bcd}$	S_{bcd}
17% (1)	$S_{ab}, S_{ad}, S_{abc}, S_{abd}, S_{acd}, S_{abcd}$	S_{abcd}
17% (1)	$S_f, S_{bf}, S_{cf}, S_{df}, S_{bcf}, S_{bdf}, S_{cdf}, S_{bcdf}$	S_{bcdf}

Table 3.3: Frequent and closed termsets for the sample document collection of Example 1.

Generating Closed Termsets

Determining closed termsets is an extension of the problem of mining frequent termsets. Our approach is based on an efficient algorithm called *CHARM* (Zaki, 2000). We adapt that algorithm to handle terms and documents instead of items and transactions, respectively.

The starting point of the algorithm is the set of frequent termsets for a document collection. Following a total ordering criterion, the lexicographic order in our case, we determine all possible closures, testing whether each termset is closed or not. Whenever a termset is subsumed by other within a closure, that termset is removed from the set of closed termsets.

It is proven that the number of enumerated closed maximal termsets does not depend on the total ordering criterion chosen. However, this is not the case for its enumeration algorithm efficiency. For a complete study of the efficiency of several total ordering criteria please refer to (Zaki et al., 1997). For results reported in Chapter 5, we choose the best total ordering criteria, which is based on the frequency of the termsets.

Formally, assume two termsets S_i and S_j , where $S_i \leq S_j$ under our total ordering criterion. The comparison between lS_i and lS_j , the lists of documents associated with S_i and S_j , respectively, leads to one of the following two situations:

1. if $lS_i = lS_j$, we verify whether the termset S_j is a subset of S_i . If it is, we can discard S_j without information loss. If it is not, we can remove S_i and S_j , replacing them by $S_{i \cup j}$ in the set of current closed termsets, since the closure of S_i and S_j is equal to the closure of $S_{i \cup j}$.
2. if $lS_i \neq lS_j$, we cannot remove or discard anything, because the two termsets lead to different closures. There is only one possible action, that is to add S_i and S_j to the set of closed termsets.

Example 7 Consider our sample collection of Example 1. A lexicographic total ordering of the frequent termsets would be $S_a < S_{ab} < S_{abc} < S_{abcd} < S_{abd} < S_{ac} < S_{acd} < S_{ad} < S_b < S_{bc} < S_{bcd} < S_{bcdf} < S_{bcf} < S_{bd} < S_{bdf} < S_{bf} < S_c < S_{cd} < S_{cdf} < S_{cf} < S_d < S_{df} < S_f$.

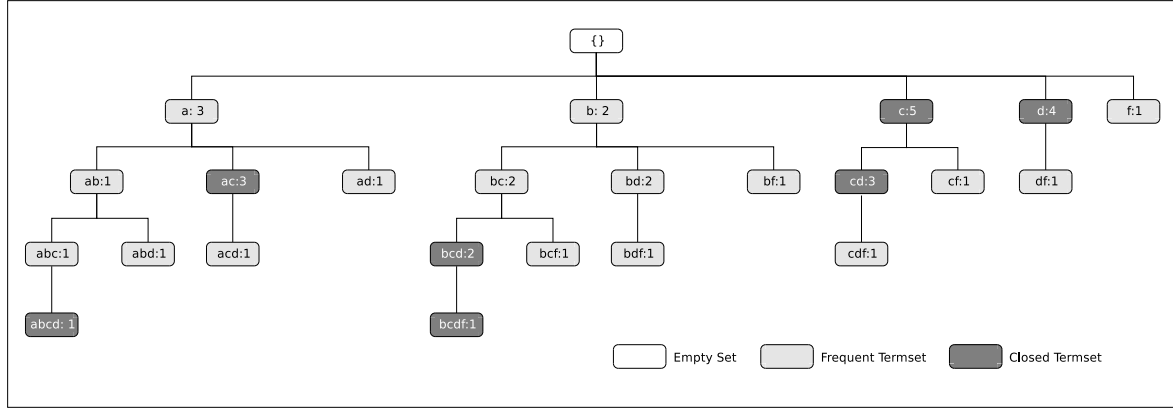


Figure 3.2: Frequent and closed termsets for the sample document collection of Example 1 for all valid minimal frequency values.

The starting point of the algorithm is the set of frequent termsets. The set of closed termsets, denoted C , is initially set to the empty set. To determine the closed termsets, we start by comparing S_a with the termset S_{ab} that comes after it. Since $lS_a \neq lS_{ab}$, both termsets are added to C . Following, we compare S_{ab} with S_{abc} . Since $lS_{ab} = lS_{abc}$ and S_{abc} is not a subset of S_{ab} , we replace S_{ab} and S_{abc} by $S_{ab \cup abc}$, i.e., S_{abc} . These comparisons proceed to the following termsets analogously, until we compare S_{abcd} with S_{ac} . Since $lS_{abcd} \neq lS_{ac}$, we add S_{ac} to C . This process continues until there are no termsets in the set of frequent termsets to be evaluated. Figure 3.2 shows the lattice of the frequent termsets with the closed ones highlighted.

3.5 Maximal Termsets

In this section we use the concept of frequent termset to introduce the maximal termsets. The main differences between the maximal and closed termsets are also explained.

Definition 11 A maximal termset MS_i is a frequent termset that is not a subset of any other frequent termset. That is, given the set $S_D \subseteq S$ of frequent termsets that occur in all documents from D , a maximal termset MS_i satisfies the property that $\nexists S_j \in S_D | MS_i \subset S_j$.

Let FT be the set of all frequent termsets, CFT be the set of all closed termsets, and MFT be the set of all maximal termsets. It is straightforward to see that the following relationship holds: $MFT \subseteq CFT \subseteq FT \subseteq V$. The set MFT is much smaller than the set CFT , which itself is much smaller than the set FT , which itself is much smaller than the vocabulary set V . It is proven that the set of maximal termsets associated with a document collection is the minimum amount of information necessary to derive all frequent termsets associated with that collection (Gouda and Zaki, 2001). It is also proven that the number

Frequency (ds)	Frequent Termsets	Closed Termsets	Maximal Termsets
83% (5)	S_c	S_c	
67% (4)	S_d	S_d	
50% (3)	S_a, S_{ac}	S_{ac}	
50% (3)	S_{cd}	S_{cd}	
33% (2)	$S_b, S_{bc}, S_{bd}, S_{bcd}$	S_{bcd}	
17% (1)	$S_{ab}, S_{ad}, S_{abc}, S_{abd}, S_{acd}, S_{abcd}$	S_{abcd}	S_{abcd}
17% (1)	$S_f, S_{bf}, S_{cf}, S_{df}, S_{bcf}, S_{bdf}, S_{cdf}, S_{bcdf}$	S_{bcdf}	S_{bcdf}

Table 3.4: Frequent, closed, and maximal termsets for the sample document collection of Example 1.

of enumerated maximal termsets does not depend on the total ordering criterion chosen. However, this is not the case for its enumeration algorithm efficiency. For a complete study of the efficiency of several total ordering criteria please refer to (Zaki et al., 1997). For results reported in Chapter 5, we choose the best total ordering criteria, which is based on the frequency of the termsets.

Maximal termsets automatically discard the termsets that does not aggregate any new correlation information, that is, those termsets that are subsets of the maximal termsets. For sake of retrieval, maximal termsets are interesting because they represent a significant reduction on the computational complexity and on the amount of data that has to be analyzed, and can be used when more specific co-occurrence patterns are needed.

Example 8 We use the same dataset of Example 1, where $q = \{a, b, c, d, f\}$ and C is the whole collection of documents. Table 3.4 shows all frequent, closed and maximal termsets for the sample document collection and their respective frequencies. As mentioned before, it is possible to vary the number of frequent termsets by changing the minimum frequency. Regarding the maximal termsets, even in this small example, we can see that the number of maximal termsets is significantly smaller than the number of closed and frequent termsets. We have 23 frequent termsets, 7 closed termsets and just 2 maximal termsets.

Generating Maximal Termsets

Determining maximal termsets is also an extension of the problem of mining frequent termsets, and these frequent termsets are the starting point of our algorithm. Our approach is based on an efficient algorithm called *GenMax* (Gouda and Zaki, 2001), which has been adapted to handle terms and documents instead of items and transactions, respectively.

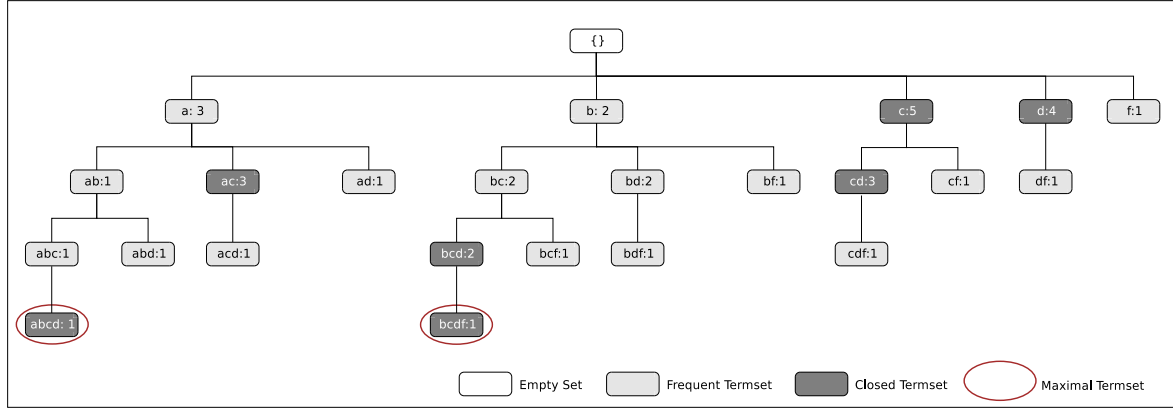


Figure 3.3: Frequent, closed, and maximal termsets for the sample document collection of Example 1 for all valid minimal frequency values.

The *GenMax* algorithm utilizes a backtracking search for efficiently enumerating all maximal termsets. Several other optimizations are also used to quickly prune away a large portion of the subset search space. It is not in the scope of this work the complete description of all the proposed optimizations. Only the main feature of the *GenMax* algorithm is covered. The termsets are verified for being maximal according to a total ordering criteria, which is the lexicographic order in our case. The starting point of the algorithm is the set of frequent termsets for a document collection. A termset X is represented by the terms that compose it and the list of documents ls_X where the termset occurs. The algorithm considers that all frequent termsets are potentially maximal and verifies whether the premise applies for any of them. Whenever a frequent termset is subsumed by other frequent termset, the termset is removed. The *MFT* corresponds to all frequent termsets that does not have any frequent termsets as its superset.

Example 9 Considering our sample collection, the total ordering of the frequent termsets would be $S_a < S_{ab} < S_{abc} < S_{abcd} < S_{abd} < S_{ac} < S_{acd} < S_{ad} < S_b < S_{bc} < S_{bcd} < S_{bcdf} < S_{bcf} < S_{bd} < S_{bdf} < S_{bf} < S_c < S_{cd} < S_{cdf} < S_{cf} < S_d < S_{df} < S_f$. The starting point of the algorithm is the set of potentially maximal termsets, denoted C , that is initialized with all frequent termsets. The determination of the maximal termsets starts by comparing S_a with the termsets that come after it. The comparison between S_a and S_{ab} shows that S_a is a subset of S_{ab} , resulting in its removal from C . The next comparisons results in the removal of all termsets, except S_{abcd} and S_{bcdf} , which are not a subset of any other termset. Figure 3.3 shows the lattice of the frequent termsets with the maximal ones highlighted.

3.6 Bibliography Revision

There are several proposals for mining association rules from transaction data. Some of these proposals are constraint-based in the sense that all rules must fulfill a predefined set of conditions, such as support and confidence (Agrawal et al., 1993a, 1996; Bayardo et al., 1999; Veloso et al., 2002). The second class identify just the most interesting rules (or optimal) in accordance to some interestingness metric, including confidence, support, gain, chi-square value, gini, entropy gain, Laplace, lift, and conviction (Webb, 1995; Liu et al., 1999; Bayardo and Agrawal, 1999). However, the main goal common to all of these algorithms is to reduce the number of generated rules. There are some other efforts that exploit quantitative information present in transactions for generating association rules (Srikant and Agrawal, 1996; Aumann and Lindell, 1999; Miller and Yang, 1997; Zhang et al., 1997; Pôssas et al., 2000).

In this context, many algorithms for efficient generation of frequent itemsets have been proposed in the literature since the problem was first introduced in Agrawal et al. (1993b). The *DHP* algorithm (Park et al., 1995) uses a hash table in pass k to perform efficient pruning of $(k + 1)$ -itemsets. The *Partition* algorithm (Savasere et al., 1995) minimizes I/O by scanning the database only twice. In the first pass it generates the set of all potentially frequent itemsets, and in the second pass the support for all them is measured. The above algorithm is based on specialized techniques which do not use any database operations. Algorithms using only general purpose DBMS systems and relational algebra operations have also been proposed (Holsheimer et al., 1995; Houtsma and Swami, 1995).

The closed itemset mining, initially proposed in Pasquier et al. (1999), mines only those frequent itemsets having no proper superset with the same support. Mining closed itemsets, as show in Zaki (2000), can lead to orders of magnitude smaller result set while retaining the completeness. In the last several years, extensive studies have proposed fast algorithms for mining closed itemsets such as *A-close* (Pasquier et al., 1999), *CLOSET* (Pei et al., 2000) and *CHARM* (Zaki and Hsiao, 2002). *A-close* is an Apriori-like algorithm that directly mines frequent closed itemsets. *CLOSET* uses a novel frequent pattern tree (FP-tree) structure, which is a compressed representation of all the transactions in the database. It uses a recursive divide-and-conquer and database projection approach to mine long patterns. *CHARM* uses a dual itemset search tree, using a efficient hybrid search that skips many levels. It also uses a fast hash-based approach to remove any "non-closed"sets found during computation.

Methods for finding the maximal elements include *All-MFS* (Gunopulos et al., 1997), which works by iteratively attempting to extend a working pattern until failure. *MaxMiner* (Bayardo, 1998) uses efficient pruning techniques to quickly narrow the search. It employs a breadth-first transversal of the search space. It also reduces database scanning by employing a lookahead pruning strategy, i.e., if a node with all its extensions can determined to be

frequent, there is no need to further process that node. The *Pincer-Search* (Lin and Kedem, 1998) constructs the candidates in a bottom-up manner like *Apriori*, but also starts a top-down search at the same time, maintaining a candidate set of maximal patterns, which is a superset of the maximal patterns. *Depth-Project* (Agrawal et al., 2000) finds long itemsets using a depth-first search of a lexicographic tree of itemsets, and uses a counting method based on transaction projections along its branches. It returns a superset of the MFI and would require post-pruning to eliminate non-maximal patterns. *MAFIA* (Burdick et al., 2001) uses three pruning strategies to remove non-maximal sets. The first is the lookahead pruning first used in MaxMiner. The second is to check if a new set is subsumed by an existing one. The last technique combines two sets if there is a inverted list which is a subset set of other. Its also mines a superset of the MFI, and requires a post-pruning step to eliminated non-maximal patterns. *GenMax* (Gouda and Zaki, 2001) uses a backtrack search algorithm for mining maximal itemsets. It uses progressive focusing to perform “maximality” checking and diffset propagation to perform fast frequency computation.

Zaki and Ogihara (1999) present a formal complexity analysis of the association rules mining problem based on the connection between frequent itemsets and bipartite cliques. This work provides the reasons why all current association rules mining algorithms exhibit linear scalability in database size.

3.7 Summary

This chapter shows how to model dependences among index terms using a association rules based framework. We present the concept of termsets, that quantify the correlations as the simultaneous occurrence of terms in a set of documents. Three special types of termsets, the proximate, closed, and maximal termsets, were also presented with its corresponding generating algorithms and properties. We also showed a detailed bibliographic discussion providing the well known algorithms for generating all presented termsets types. In the next chapter we describe how we use termsets as a basis for a new information retrieval model that retrieves documents efficiently.

Chapter 4

Set-Based Vector Model

To use termsets for ranking purposes, we propose a variant of the classic vector space model. This new information retrieval model is referred to as *set-based vector model*, or simply set-based model. In this chapter we discuss its fundamental features and its ranking algorithm.

4.1 Documents and Queries Representations

A document d_j and a user query q are represented as vectors of weighted termsets as follows:

$$\begin{aligned}\vec{d}_j &= (w_{S_1,j}, w_{S_2,j}, \dots, w_{S_{2^t},j}) \\ \vec{q} &= (w_{S_1,q}, w_{S_2,q}, \dots, w_{S_{2^t},q})\end{aligned}$$

where t corresponds to the number of distinct terms in the collection, $w_{S_i,j}$ is the weight of termset S_i in the document d_j , and $w_{S_i,q}$ is the weight of termset S_i in the query q .

Example 10 Consider a vocabulary of two terms $T = \{1, 2\}$, and a collection C of two documents d_j , $1 \leq j \leq 2$, given by $C = \{(1, 2, 1, 2, 2, 2), (1, 2, 1, 2, 1, 2, 2, 2)\}$, and a user query $q = \{1, 2\}$. Figure 4.1 shows the termsets vector space defined for the documents of the collection C and the specified user query q . There are 3 termsets associated with q . The 1-termsets are S_1 and S_2 , and the 2-termset is S_{12} . For simplicity, we are considering the weights of a termset in a document, as its number of occurrences.

One important simplification in our model is that the vector space is induced just for the termsets generated from the query terms. Documents and queries are represented by vectors in a 2^t -dimensional space, where t is the number of unique index terms in the vocabulary. However, only the dimensions corresponding to termsets enumerated for the query terms are taken into account. This is important because the number of termsets induced by the queries is usually small (see Section 5.4). Also, we can use proximity information among patterns

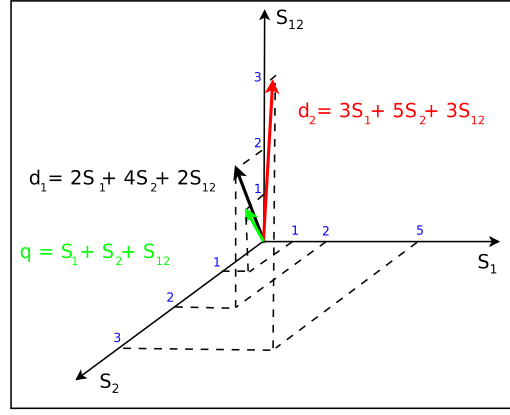


Figure 4.1: Vector space representation for the Example 10.

of term co-occurrence to further reduce the number of termsets to be considered. Proximity information, as we shall see in Section 5.3, further provides significant gains in retrieval effectiveness.

4.2 Termset Weighting Schema

Good term weighting schemes are usually based on three basic criteria. First, they take into account the number of times that an index term occurs in a document. Second, they emphasize term scarcity by reducing the weight of terms that appear in many documents. Third, they penalize long documents because these are naturally more likely to contain any given query term. This is usually done with the introduction of a normalization factor to discount the contribution of long documents (Salton and Buckley, 1988; Robertson et al., 1995).

Index term weights can be calculated in many different ways (Salton and Yang, 1973; Yu and Salton, 1976; Salton and Buckley, 1988). To abide by the first two basic criteria above, the best known term weighting schemes use weights that are a function of (i) $tf_{i,j}$, the number of times that an index term i occurs in a document j , and (ii) df_i , the number of documents of the collection that contain an index term i . Since scarce terms are more selective, an inverse function of the document frequency df_i is used, the inverse document frequency idf_i . The resulting term weighting strategy is called a $tf \times idf$ scheme.

In the set-based model, weights are associated with termsets (instead of terms). These weights are a function of the number of occurrences of the termset in a document and in the whole collection, analogously to $tf \times idf$ term weights. Any terms weights schemas based on the described criteria, mainly those used in the standard vector space model and in the probabilistic relevance model, can be easily adapted for the set-based model. Following, we describe two termset weighting schemas: a standard vector space based and a BM25 based weighting schema.

VSM-Based Termset Weighting Schema

The standard vector space term weighting schema can be directly adapted to the set-based model, extending the tf and the idf functions with its corresponding in the termsets framework already presented. Formally, the weight of a termset S_i in a document d_j is defined as:

$$w_{S_i,j} = f(Sf_{i,j}) \times idS_i = (1 + \log Sf_{i,j}) \times \log \left(1 + \frac{N}{dS_i} \right) \quad (4.1)$$

where N is the number of documents in the collection, $Sf_{i,j}$ is the number of occurrences of the termset S_i in the document d_j and idS_i is the inverse frequency of occurrence of the termset S_i in the collection, scaled down by a log function. The factor $Sf_{i,j}$ subsumes $tf_{i,j}$ in the sense that it counts not only single terms but also co-occurring term subsets. The component idS_i subsumes the idf_i factor.

Similarly, the weight of a termset S_i in a query q is formally defined as:

$$w_{S_i,q} = f(Sf_{i,q}) \times idS_i = (1 + \log Sf_{i,q}) \times \log \left(1 + \frac{N}{dS_i} \right) \quad (4.2)$$

where N is the number of documents in the collection, $Sf_{i,q}$ is the number of occurrences of the termset S_i in the query q and idS_i is the inverse frequency of occurrence of the termset S_i in the collection, scaled down by a log function.

BM25-Based Termset Weighting Schema

The BM25 weighting scheme is also defined as a function of the number of occurrences of the term in a document, in the whole collection. Adapting this weighting scheme to the termsets its quite straightforward. Formally, the weight of a termset S_i in a document d_j is defined as:

$$w_{S_i,j} = \frac{k_1 \times Sf_{i,j}}{Sf_{i,j} + k_1 \times \left(1 - b + b \times \frac{|\vec{d}_j|}{|\vec{d}|} \right)} \times \log \frac{N - dS_i + 0.5}{0.5}, \quad (4.3)$$

where $Sf_{i,j}$ is the number of occurrences of the termset S_i in the document d_j , k_1 and b are parameters, that depends on the collection and possibly on the nature of the user queries, $|\vec{d}_j|$ corresponds to a document length function, $|\vec{d}|$ is the average document length, N is the number of documents in the collection, and dS_i is the number of documents containing the termset S_i .

The weight of a termset S_i in a query q is formally defined as:

$$w_{S_i,q} = \frac{(k_3 + 1) \times Sf_{i,q}}{k_3 + Sf_{i,q}} \quad (4.4)$$

where $Sf_{i,q}$ is the number of occurrences of the termset i in the query q , and k_3 is a parameter, which depends on the collection and possibly on the nature of the user queries.

4.3 Ranking Computation

In the set-based model, we compute the similarity between a document and the user query as the normalized scalar product between the document vector $\vec{d}_j$, $1 \leq j \leq N$, and the query vector $\vec{q}$, as follows:

$$sim(q, d_j) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{S_i \in S_q} w_{S_i,j} \times w_{S_i,q}}{|\vec{d}_j| \times |\vec{q}|}, \quad (4.5)$$

where $w_{S_i,j}$ is the weight associated with the termset S_i in the document d_j , $w_{S_i,q}$ is the weight associated with the termset S_i in the query q , and S_q is the set of all termsets generated from the query terms. That is, our ranking computation is restricted to the termsets generated by the query.

The norm of d_j , represented as $|\vec{d}_j|$, is hard to compute because of the large number of termsets generated by a document. To speed up computation, we consider only the 1-termsets in the document, i.e., we use only single terms. Thus, our normalization procedure does not take into account term co-occurrences. Despite that, it addresses the third ranking criterion in Section 4.2 because it accomplishes the effect of penalizing large documents, the major objective of ranking normalization. We validate this 1-termset normalization procedure through experimentation (see Section 5.2.3).

To compute the ranking with regard to a user query q , we use the algorithm of Figure 4.2. First, we initialize the data structures (line 4) used for computing partial similarities between each termset S_i and a document d_j . For each query term, we retrieve its inverted list and determine the frequent termsets of size 1, applying the minimal frequency threshold mf (lines 5 to 10). The next step is the enumeration of all termsets based on the 1-termsets, filtered by the minimal frequency and proximity thresholds (line 11). After enumerating all termsets, we compute the partial similarity of each termset S_i with regard to the document d_j (lines 12 to 17). Following, we normalize the document similarities A by dividing each document similarity A_j by the norm of the document d_j (line 18). The final step is to select the k largest similarities and return the corresponding documents (line 19).

```

SBM ( $q, mf, mp, k$ )
   $q$  : a set of query terms
   $mf$  : minimum frequency threshold
   $mp$  : minimum proximity threshold
   $k$  : number of documents to be returned
1. Let  $A$  be a set of accumulators
2. Let  $C_q$  be a set of 1-termsets
3. Let  $S_q$  be a set of termsets
4.  $A = \emptyset, C_q = \emptyset, S_q = \emptyset$ 
5. for each query term  $t \in q$  do begin
6.   if  $df_t \geq mf$  then begin
7.     Obtain the 1-termset  $S_t$  from term  $t$ 
8.      $C_q = C_q \cup \{S_t\}$ 
9.   end
10. end
11.  $S_q = \text{Termsets\_Gen}(C_q, mf, mp)$ , see Secs. 3.2, 3.4, and 3.5
12. for each termset  $S_i \in S_q$  do begin
13.   for each  $[d_j, Sf_{i,j}]$  in  $lS_i$  do begin
14.     if  $A_j \notin A$  then  $A = A \cup \{A_j\}$ 
15.      $A_j = A_j + w_{S_i,j} \times w_{S_i,q}$ , from Eqs. (4.1 or 4.3).
16.   end
17. end
18. for each accumulator  $A_j \in A$  do  $A_j = A_j \div |\vec{d}_j|$ 
19. determine the  $k$  largest  $A_j \in A$  and return the corresponding documents
20. end

```

Figure 4.2: The set-based model ranking algorithm.

4.4 Computational Complexity

Different approaches to account for co-occurrence among index terms during the information retrieval process have been proposed. However, one major limitation of existing approaches is their computational cost. Several of the proposed models cannot be applied to large or even mid-size collections since their costs increase exponentially with the vocabulary size.

The complexity of the standard vector space model and the set-based model depends on the number of query terms, while the complexity of the generalized vector space model exponentially depends on the number of terms of the vocabulary. The upper bound on the number of operations performed for satisfying a query in the vector space model is $O(|q|)$, where $|q|$ is the number of terms in the query. The computational complexity for the set-based model is $O(2^{|q|})$, where $2^{|q|}$ corresponds to the upper bound for the number of termsets that can be enumerated for a query containing $|q|$ distinct terms. An immediate consequence

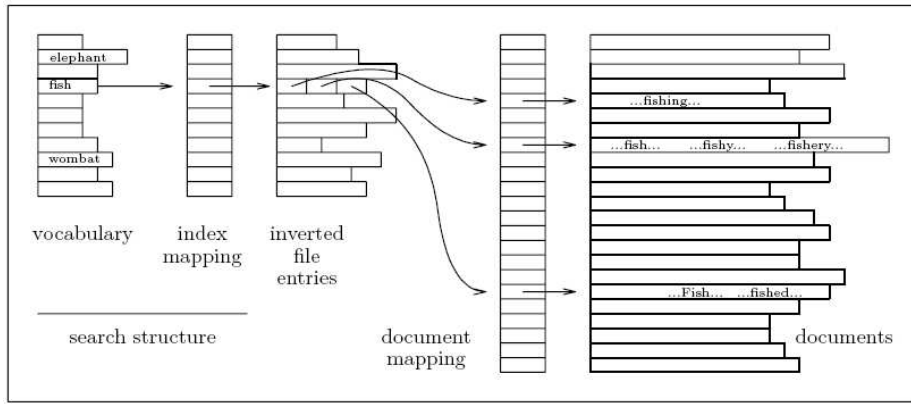


Figure 4.3: The inverted file index structure.

of this complexity analysis is that the implementation of the proposed model is practical and efficient for queries containing up to 30 terms, with processing times close to those of the standard vector space model (see Section 5.4).

4.5 Indexing Data Structures and Algorithm

The index structure used by the set-based model corresponds to the widely discussed inverted files (Witten et al., 1999). A general inverted file index for a text collection consists of two main components: (i) a set of inverted file entries, one entry per index term, each entry composed of the identifiers of the documents containing the corresponding index term, an intra-document frequency, and, optionally, a list of ordinal positions at which the term occurs in the document; and (ii) a data structure for identifying the location of the inverted file entry for each term, composed of a vocabulary of query terms and of an index mapping (that maps ordinal term numbers onto disk addresses in the inverted index). This arrangement is illustrated in Figure 4.3. The index mapping can either be stored on disk as a separate file or can be held in memory with the vocabulary. We assume that inverted file entries store ordinal document numbers rather than addresses. Thus, to map the resulting document identifiers to disk addresses there must also be a document mapping.

We store the contents of each inverted file entry contiguously, in contrast to other schemes in which entries are often stored as linked lists with each node randomly placed on disk. The inverted list entries are sorted by the document identifiers, and are compressed using Elias Gamma (Elias, 1975), a non-parameterized bitwise method of coding integers. Queries are processed according to the ranking algorithm discussed in the previous section, which uses the vocabulary and index mapping to find the location of the inverted file entry for each query term.

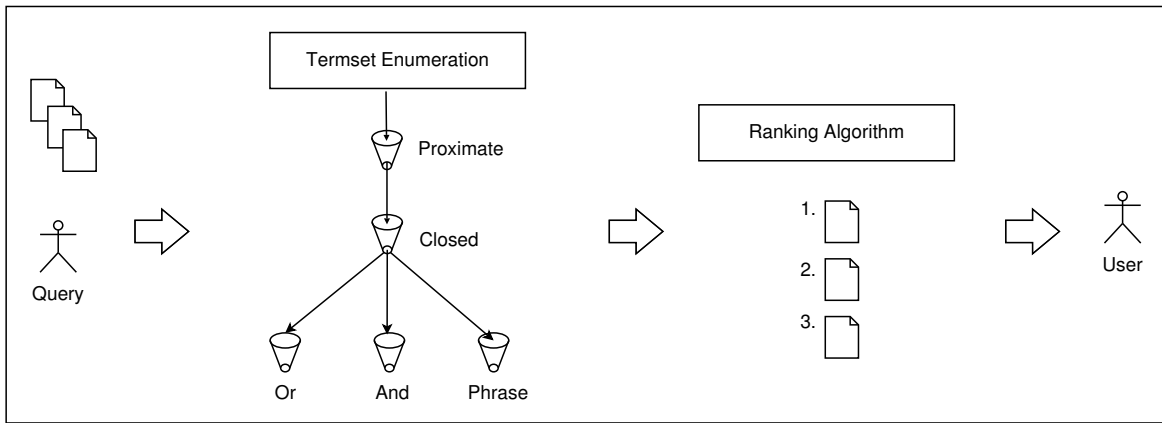


Figure 4.4: The set-based model work-flow.

We use a sort-based compressed multi-way merge algorithm, extensively discussed in Witten et al. (1999), to build the index structure for the 1-termsets. The inverted lists of high order termsets, that is, termsets with more than one term, are build during query processing and do not have to be stored on disk.

4.6 Set-Based Model Applications

In this section we describe some applications of the set-based model. This applications includes query processing and automatically query structuring.

4.6.1 Query Processing

In Boolean retrieval systems Buell (1981); Paice (1984), the terms in the user query are connected by the Boolean operators AND, OR and NOT. Boolean connectives are useful for specialized users who, knowing the document collection well, can use them to provide a more selective structure to their queries.

Figure 4.4 shows the set-based model work-flow for query processing. The first step consists in the specification of a user query. Next, the set-based model enumerates all closed termsets according to the query type (disjunctive, conjunctive and phrase queries) and the frequency and proximity thresholds. As we shall see in detail in the following sections, the evaluation of the enumerated closed termsets is quite different depending on the query type being considered. Finally, the documents are ranked according to their similarities to the enumerated termsets. The ranked documents are returned to the user.

Disjunctive Queries

One of the main advantages of the vector space model is its partial matching strategy, which allows the retrieval of documents that approximate the query conditions. This strategy corresponds, conceptually, to the processing of disjunctive queries.

Given a user query, the minimal frequency, and the proximity thresholds, the enumeration algorithm determines all closed termsets. Since the closed termsets represent all query-related patterns of term co-occurrence, partial matching between the query and the documents is allowed.

Example 11 Consider our collection of Example 1 and the user query $q = \{a, b, c, d\}$. Assume that the minimal frequency and minimal proximity threshold values are set to 1 and 10, respectively. Then, the termsets enumeration algorithm finds 6 closed termsets associated with q , S_c , S_d , S_{ac} , S_{cd} , S_{bcd} , and S_{abcd} , all of which occur in our sample collection.

Conjunctive Queries

Different search engines and portals might have different default semantics for handling a multi-word query. Despite that, all major search engines assume conjunctive queries as a default querying strategy. That is, all query words must appear in a document that is included in the ranking.

The main modification of the set-based model for the processing of conjunctive queries is related to the termset enumeration algorithm. Since all query terms must occur in a document retrieved, we check if the document includes a closed termset that contains all query terms. If so, just the inverted list of these closed termset is evaluated by our ranking algorithm. Another important constraint is related to the minimal frequency threshold. We set this threshold to 1 because all documents containing all the query terms must be returned.

Example 12 We use the same dataset of Example 1, where $q = \{a, b, c, d\}$ and C is the collection of documents. We first check if the set of closed termsets contains a termset that has all query terms. In this simple example, this is the termset S_{abcd} . Its inverted list is then evaluated using our ranking algorithm. As a result, the document d_5 is returned.

Phrase Queries

A fraction of the queries in the Web include phrases, i.e., a sequence of terms enclosed in quotation marks, which means that the phrase must appear in the documents retrieved. A standard way to evaluate phrase queries is to use an extended index that includes information on the positions at which a term occurs in a document. Given information on the positions of the terms, we can determine which documents contain a phrase declared in a query.

The set-based model can be easily adapted to handle phrase queries. To achieve this, we enumerate the set of closed termsets using the same restrictions applied for conjunctive queries. If there is a closed termset containing all query terms, we just need to verify if the query terms are adjacent. This is done by checking whether the ordinal word positions in the index are adjacent. The minimal proximity threshold is set to 1 to select only the adjacent termsets.

Example 13 Consider again the dataset of Example 1, where $q=\{“a\ b\ c\ d”\}$. The closed termset S_{abcd} matches the requirements for phrase query processing. Thus, just its inverted list is evaluated by the ranking algorithm.

4.6.2 Query Structuring

The huge volume of information now available on the Web has posed challenges to the users. Any short query presents the user with thousands of answers. If the first 10-20 answers are not satisfactory, the user has to sift the answers of his interest among dozens, even hundreds, of answers. Frequently, he gets impatient and takes one of two actions: he rewrites his query in a different form or he simply gives up.

If our user is more persistent, he will rewrite his query trying to make it more specific. It is our believe that, as users of Web search engines become more knowledgeable, they will write more specific, longer, and complex queries (we consider here that a *long* or *complex* query is one that contains two or more terms). Let us proceed with one example.

Consider the query “*tylenol drogaria belo horizonte*” formulated by a Brazilian student who has headache and seeks for a drugstore in the city of Belo Horizonte, where he lives. He wrote his query thinking of tylenol, the brand name of a popular pain killer. It is a simple request. Let us look at the results.

At *UOL Busca*, search engine of UOL¹, largest paid ISP in the Brazil, only one answer is returned: Tcafarma, a company that distributes medicines to hospitals and pharmacies. It does not solve the problem of our student. At Yahoo², three answers are returned: Tcafarma as before, Mercado Mineiro, an institute that runs market research, and Drogaria Pacheco, a small drugstore located far away from the student’s home. Again, our student would have obtained frustrating answers. At Google³, twenty-two answers are returned, which would seem better, but that none of them are relevant.

Curiously, had our user formulated his query as “*drogaria belo horizonte*”, instead, he would have gotten pointers to large chains of drugstores in Belo Horizonte, with stores close to his home, right in the first top ten answers in all three search engines. He did not obtain

¹<http://www.uol.com.br>

²<http://www.yahoo.com.br>

³<http://www.google.com.br>

this information with his original query because all search engines process the user queries as a *conjunction* of the query terms.

The problem that we face can then be formulated as follows:

Given a conjunctive user query, is it reasonable to structure the query in smaller conjunctive components? When should this be attempted? Is it possible to improve precision through one such mechanism?

Our proposal to solve these questions is a new technique for automatically structuring queries based on *maximal termsets*. Our technique is referred to as *SBM-MAX*, i.e., maximal set-based model, and the key idea is that information derived from the distributions of the query conjunctive components in the document collection can be used to guide the query structuring process. Given a user query, the effect is that, we can provide the user with the “best” set of answers that is possible to produce by directly matching the query against the documents in the collection. “Best” in the sense that the largest query components are used, not necessarily that they lead to higher precision figures. It is intuitive though that, if the user query makes sense, the query components best supported by the document collection are more likely to produce more relevant answers – a conjecture confirmed by our experimentation. Once this best set of answers has been produced, Web ranking techniques such as Page ranking (Brin and Page, 1998) can be applied.

Example 14 *We instantiate the problem with the Example 1. Our example collection C is composed of just 6 documents, none of them containing the following 5 terms $\{a, b, c, d, f\}$ of the vocabulary. Thus, the conjunctive user query $q = \{a, b, c, d, f\}$ returns the empty set. However, it is clear in this case that there are two answers that are far better than returning an empty set. These are documents d_5 and d_6 . Notice that they could have been returned had we processed the user query as $q' = \{a, b, c, d\} \vee \{b, c, d, f\}$, which are two maximal termsets for the original user query in the context of our example document collection.*

Our approach allows to naturally produce answers to queries that otherwise would lead to empty result sets. This is accomplished by finding all maximal termsets derived from the user query that have support in the document collection. That is, maximal termsets provide a simple and elegant formalism for naturally structuring conjunctive user queries formed by an arbitrary number of terms.

The set-based model can be easily adapted to automatically structuring user queries. To achieve this, as mentioned before, we restrict the use of termsets to maximal termsets. Given a user query q , we enumerate its related maximal termsets and compute the partial similarities between each maximal termset S_i and a document d_j , according to the set-based model ranking formula.

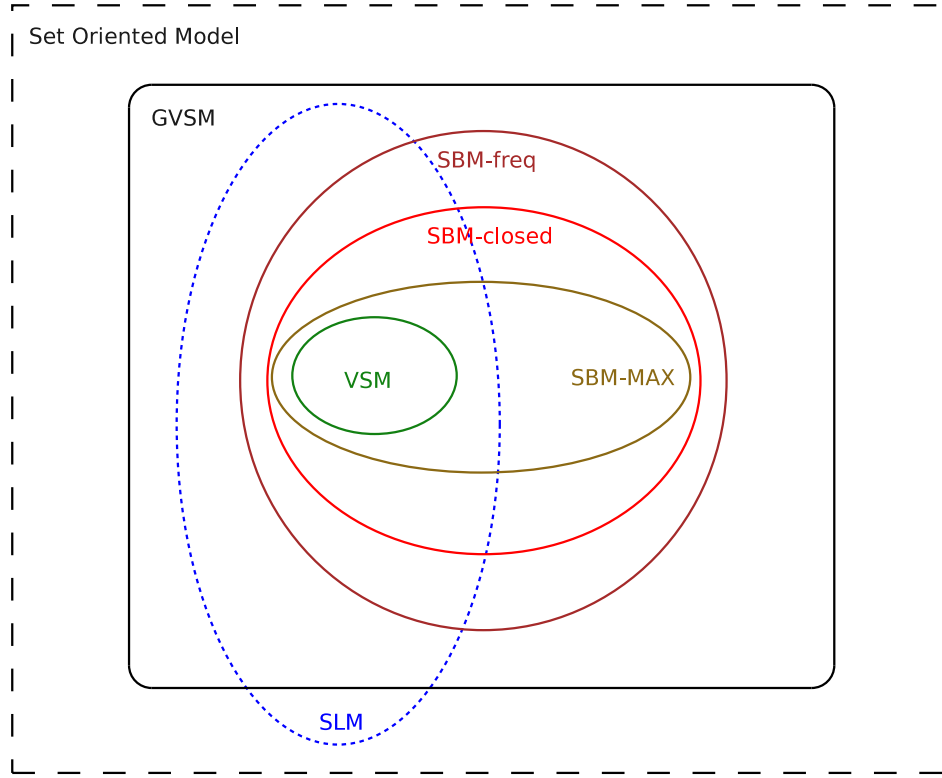


Figure 4.5: Information retrieval models expressiveness.

Example 15 To illustrate, let us consider our Example 1. For the query $q = \{a, b, c, d, f\}$, the related maximal termsets are S_{abcd} and S_{bcdf} . We process the enumerated maximal termsets and rank the retrieved documents using Eq. (4.5) with the termsets weighting scheme presented in Eq. (4.3).

4.7 Set-Based Model Expressiveness

The information retrieval models that take into account the correlation among index terms can be classified in terms of its expressiveness. In this work we are considering the expressiveness as the number of correlations used for each retrieval model. The expressiveness relation between some of the presented models and our model is showed in Figure 4.5.

As stated before, the set-based model, independently of the type of termsets used, corresponds to a generalization of the standard vector space model (VSM), since the single index terms are represented by the 1-termsets. The cardinality of the set-based model regarding frequent, closed, and maximal termsets is represented by its respective instances (SBM-freq, SBM-closed, and SBM-MAX). The generalized vector space model (GVSM) is more complex than both the standard vector space model and the set-based model because all valid co-occurrence patterns derived from the 2^t possible min-terms are used.

A statistical language model (SLM) is a probability distribution over all possible sentences or other linguistic units in a language. However, for efficiency reasons, the published statistical language models limit the number of terms in correlation patterns to 2 or 3 terms, or limit the correlations to sentence bounded words. A recent work (Cao et al., 2005) have expanded the correlation space with the use of a knowledge base, which explains why this class of models contains correlations not represented by the generalized vector space model.

The set oriented model corresponds to a theoretical framework for representing concepts, which can be modeled as co-occurrence patterns, as knowledge based entries, etc. Due to its nature, the set oriented model sets the upper bounds for the number of correlations taken into account for all correlation-based models.

4.8 Summary

In this chapter the basic features of the model being proposed were developed and justified. The justification for the set-based model comprises the termsets, a framework for representing correlation patterns between terms, which concepts and algorithms were already presented in Chapter 3. We showed how to overcome the independence assumption associated with the vector space and several others well know information retrieval models using the termsets. The building blocks of a complete information retrieval model, such as (i) documents and queries representation, (ii) the index terms weighting schema, (iii) the ranking computation and its computational cost, and (iv) the index structure and algorithm, were also described.

We also showed how the different query processing types, such as conjunctive, disjunctive, and phrase queries, can be modeled using the set-based model and the closed termsets. We also presented SBM-MAX, a formalism for automatically structuring a user query into a disjunction of smaller conjunctive subqueries using the maximal termsets. A comparison of the expressiveness of our model and the other models that take into account the correlation among query terms was also presented. In the next chapters we will describe experimental setup and results for the evaluation of the set-based model in terms of both effectiveness and computational performance using several reference collections.

Chapter 5

Experimental Results

In this chapter we discuss the experimental results for the set-based model regarding retrieval effectiveness and computational performance.

5.1 Experimental Setup

In this section we describe our experimental environment. We present the evaluation metrics and the reference collections used to evaluate the set-based model in comparison with the vector space model, generalized vector space model, and the BM25 probabilistic relevance model.

5.1.1 Evaluation Metrics

In this section, we introduce the most common evaluation metrics which are necessary for understanding the results showed in the following sections. We quantify the retrieval effectiveness of the various approaches through standard measures of average recall and precision (Baeza-Yates and Ribeiro-Neto, 1999). Computational performance is evaluated through query response times.

Consider a user query q and its set R of relevant documents. The relevance judgments are accomplished through human evaluations, made by specialists in the query domain, or by a group of system users. Assume that the retrieval method being evaluated process the query q and returns a document answer set A . Let $|A|$ be the number of documents in this set and $|R|$ be the total number of relevant documents. The higher the overlap between the set of documents A and R them, the better is considered the result. Recall and precision are defined as a means to characterize this overlap, as follows.

Definition 12 *Recall* is the fraction of correct answers that were properly retrieved in A , formally:

$$recall = \frac{|A \cap R|}{|R|}$$

Definition 13 *Precision* is the fraction of all answers in A that are correct, formally:

$$precision = \frac{|A \cap R|}{|A|}$$

Frequently, we want to evaluate average precision at given recall levels. The standard 11-point average precision measure returns precision at 0%, 10%, 20%, ..., 100% of recall level. For instance, precision at 10% recall is the precision when 10% of the relevant documents in the set R have been seen in the ranking, starting from the top. Average precision at 10% recall is the average precision for all test queries, taken at 10% recall. Plotting the precision at the 11 standard recall points allows us to easily evaluate and compare the quality of ranking algorithms.

One additional approach is to compute average precision at given *document cutoff values*. For instance, we can compute the average precision when 5, 10, 15, 20, 30, 100, 200, 500, or 1000 documents have been seen. The procedure is analogous to the computation of average precision but provides additional information on the effectiveness of the ranking algorithm.

We have employed four aggregate metrics for measuring retrieval effectiveness in our experiments: (i) standard 11-point average precision figures, (ii) average precision over the retrieved documents, (iii) average precision at 10, i.e., average precision computed over the first ten documents retrieved, and (iv) document level averages, which corresponds to the precision at 9 cutoff values.

Measuring differences in precision and recall between retrieval systems is only indicative of the relative effectiveness. It is also necessary to establish whether the difference is statistically significant. Per-query recall-precision figures can be used in conjunction with statistical significance tests to establish the likelihood that a difference is significant. We use the “Wilcoxon’s rank test”, which has been showed by Zobel (Zobel, 1998) and others to be suitable for this task. In our comparisons, a 95% level of confidence is used to find whether the results are statistically significant.

5.1.2 The Reference Collections

A test, or a reference, collection consists of a set of documents, a set of topics, and a set of relevance judgments. A topic is a description of the information being sought. The relevance judgments specify the documents that should be retrieved in response to each topic. In this paradigm, the effectiveness of different retrieval mechanisms can be directly compared on the common task defined by the test collection (Salton, 1992).

Characteristics	Collection				
	<i>CFC</i>	<i>CISI</i>	<i>TREC-8</i>	<i>WBR-99</i>	<i>WBR-04</i>
Number of Documents	1,239	1,460	528,155	5,939,061	15,240,881
Number of Distinct Terms	2,105	10,869	737,833	2,669,965	4,217,897
Number of Available Topics	100	76	450	100,000	1,733,087
Number of Topics Used	66	50	50 (401-450)	50	100
Avg. Terms per Topic (1)	3.82	9.44	10.80	1.94	-
Avg. Terms per Topic (2)	-	-	4.38	1.94	-
Avg. Terms per Topic (3)	-	-	10.80	-	5.95
Avg. Relevant Docs. per Topic	29.04	49.84	94.56	35.40	8.40
Size	1.3 (MB)	1.3 (MB)	2 (GB)	16 (GB)	80 (GB)

- (1) Used for disjunctive queries evaluation.
(2) Used for conjunctive and phrase queries evaluation.
(3) Used for structured queries evaluation.

Table 5.1: Characteristics of the five reference collections.

In our evaluation we use five reference collections. Table 5.1 presents the main features of these collections.

CFC

The cystic fibrosis (CFC) collection (Shaw et al., 1991) is composed of 1,240 documents indexed by the term “cystic fibrosis” in the National Library of Medicine’s MEDLINE database. This collection includes 100 sample queries. However, only 66 of these queries has its corresponding relevant documents. The average number of relevant documents for each query is approximately 29. The CFC collection, despite its small size, has two important characteristics. First, its sets of relevant documents were generated directly by human experts through a careful evaluation strategy. Second, it includes a good number of queries (relative to the collection size) and, as a result, the queries overlap among themselves. The mean number of keywords per query is 3.82.

CISI

The documents in the CISI collection were selected from a previous collection created by Small (Small, 1981) in the Information Science Institute. The selected documents refer to information science. This collection also includes 76 queries, 35 of them expressed through boolean logic and the remaining 41 expressed in natural language. However, only 50 of these queries has its corresponding relevant documents. Since the queries are quite general, the average number of relevant documents for each query is approximately 50. The mean number of keywords per query is 9.44.

TREC-8

The TREC reference collections have been growing steadily over the years. At TREC-8 (Voorhees and Harman, 1999), which is used in our experiments, the collection size is roughly 2 gigabytes (disks 4 and 5, excluding the Congressional Record sub-collection). The documents in the TREC-8 collection come from the following sources: The Financial Times (1991-1994), Federal Register (1994), Foreign Broadcast Information Service, and LA Times.

Each TREC collection includes a set of example *information requests* that can be used for testing a new ranking algorithm. Each information request is a description of an information need in natural language. The TREC-8 has a total of 450 such requests, usually referred to as topics. Our experiments are performed with the 50 topics numbered 401–450. All queries were generated automatically in the following way: the disjunctive queries were generated using the title, description and narrative of each topic and the conjunctive and phrase queries were generated using just the title and description of the topics. Disjunctive query generation is different from conjunctive and phrase queries generation because the latter ones require that the terms represent valid relationships or phrases. The mean number of keywords per query is 10.80 for disjunctive query processing and 4.38 for conjunctive and phrase query processing.

In TREC-8, the set of relevant documents for each information request was obtained as follows. For each information request, a pool of candidate documents was created by taking the top k documents in the ranking generated by various retrieval systems participating in the TREC conference. The documents in the pool were then showed to human assessors who ultimately decided on the relevance of each document. The average number of relevant documents per topic is 94.56.

WBR-99

The WBR-99 reference collection is composed of a database of Web pages, a set of example Web queries, and a set of relevant documents associated with each example query. The WBR-99 databases is composed of 5,939,061 pages of the Brazilian Web, under the domain “.br”, respectively. The pages were automatically collected by the document crawler described in (Silva et al., 1999).

For the WBR-99 collection, a total of 50 example queries were selected from a log of 100,000 queries submitted to the *TodoBR* search engine¹. The queries selected were the 50 most frequent ones, excluding queries related to sex. The mean number of keywords per query is 1.94 (disjunctive, conjunctive, and phrase queries). Of the 50 selected queries, 28

¹<http://www.todobr.com.br>

were quite general, like “tabs”, “movies”, or “mp3”. Other 14 queries were more specific, but still on a general topic, like “transgenic food” or “electronic commerce”. Finally, 8 queries were quite specific, consisting mainly of music band names. The mean number of keywords per query is 1.94.

For each of the 50 example queries of the WBR-99 collection we composed a query pool formed by the top 20 documents, retrieved by each of the following eight ranking variants we considered: disjunctive queries with the vector, the set-based, and the proximity set-based models; conjunctive queries with the vector, the set-based, and the proximity set-based models; phrase queries with the vector and the set-based models. The pool was expanded with several executions of the set-based and the proximity set-based models varying the minimal frequency and minimal proximity thresholds. Each query pool contained an average of 83.26 pages. All documents in each query pool were submitted to a manual evaluation by a group of 10 users, all of them familiar with Web searching and with a Computer Science background. Users were allowed to follow links and to evaluate the pages according not only to their textual content, but also according to their linked pages and graphical content (flash, or dynamic HTML animations). The average number of relevant pages per query pool is 35.4. We adopted the same pooling method used for the Web-based collection of TREC (Hawking et al., 1998, 1999).

WBR-04

The WBR-04 reference collection is also composed of a database of Web pages, a set of example Web queries, and a set of relevant documents associated with each example query. The WBR-04 databases is composed of 15,240,881 pages of the Brazilian Web, under the domain “.br”, respectively. These pages were collected using the same crawler used in the WBR-99 collection.

For the WBR-04 collection, a total of 100 example queries were selected from a log containing 1,733,087 queries submitted to the *UOL Busca* search engine². Since this collection will be used to evaluate our query structuring mechanism, we are interested in *complex queries*, we selected those composed of four or more terms, which corresponds to 23% of the processed log. Among all queries with four or more terms in our log, we selected two sets of queries: the 50 most frequent ones and 50 random queries, excluding those related to the topic sex. The mean number of keywords per query is 5.95.

For each of the 100 selected queries of the WBR-04 collection we composed a query pool formed by the top 10 ranked documents, as given by each of the ranking variants we considered, i.e., the maximal set-based model, the set-based model, the probabilistic model using the BM25 weighting scheme, and the standard vector space model. The pool was

²<http://busca.uol.com.br>

expanded with several executions of the set-based and the maximal set-based models varying the minimal frequency threshold. Each query pool contained an average of 29.62 documents. We adopted the same pooling method used for the WBR-99 collection. The average number of relevant documents per query pool is 8.40.

5.2 Tuning of the Set-Based Model

In this section we show experimental results for tuning the set-based model, the proximity set-based model, i.e., the set-based model using proximity information, and the maximal set-based model, i.e., the set-based model using the maximal termsets. Our analysis verifies the impact of frequency and proximity thresholds in the retrieval effectiveness, and also presents a normalization evaluation using several well known normalization techniques.

For each test collection evaluated we built a training set composed of 15 queries, randomly selected from the available sample queries. These training sets were used for tuning of the minimal frequency and of the minimal proximity thresholds. They were also used for selecting a normalization technique among the five we had available. The remaining sample queries were used as the test set for each evaluated reference collection in the experiments reported in Sections 5.3 and 5.4.

Our experiments were performed in a Linux-based PC with a AMD-athlon 2600+ 2.0 GHz processor and 512 MBytes of RAM. We used the other remaining queries in the set of sample queries for each of the evaluated reference collection.

5.2.1 Minimal Frequency Evaluation

One of the key features of the set-based model is the possibility of controlling the minimal frequency threshold. By varying the minimal frequency, we can exploit a trade-off between precision and the number of termsets taken into consideration. The higher the minimum frequency, the smaller the number of termsets to consider, and faster the computation. However, the impact on average precision needs to be analyzed.

To illustrate, we performed a series of executions of the set-based model with the 15 disjunctive queries in our training set. For the CFC and the CISI test collections we varied the minimal frequency threshold from 1 to 50 documents. For the TREC-8 test collection, we varied the minimal frequency threshold from 1 to 90 documents. For the WBR-99 and WBR-04 test collections we varied it from 1 to 190 documents. The results are presented in Figures 5.1, 5.2, 5.3, 5.4, and 5.5. We observe that the set-based model average precision is significantly affected by the minimal frequency threshold, and the behavior is similar for all collections. Initially, an increase in the minimal frequency results in better precision.

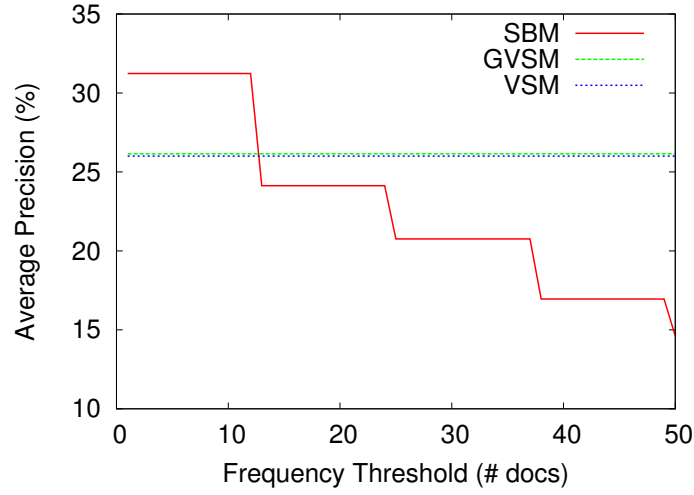


Figure 5.1: Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CFC test collection.

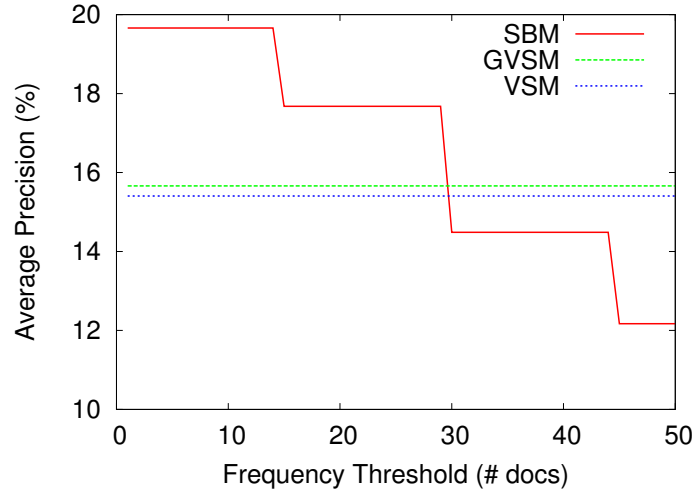


Figure 5.2: Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CISI test collection.

Maximum precision is reached for a minimal frequency threshold equal to 12 documents for the CFC, 14 for the CISI, 15 for the TREC-8, 60 for the WBR-99, and 66 for the WBR-04. These are the values used in our experimentation in Section 5.3.

For larger threshold values, precision decreases as the threshold value increases. This behavior can be explained as follows. First, an increase in the minimal frequency causes irrelevant termsets to be discarded, resulting in better precision. When the minimal frequency becomes larger than its best value, relevant termsets start to be discarded, leading to a reduction in overall precision.

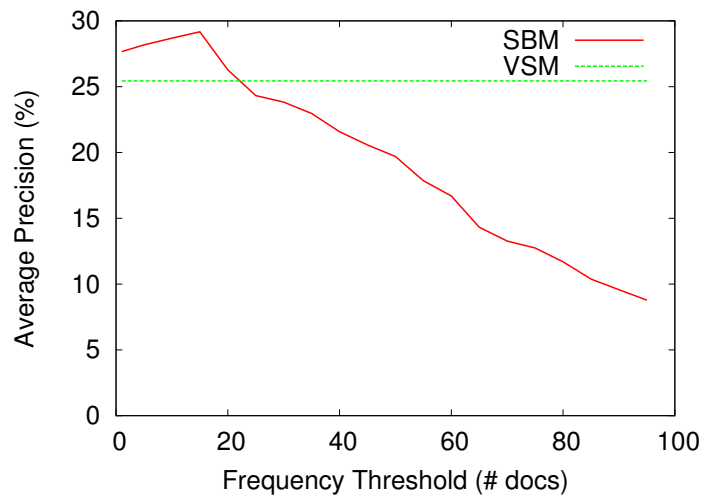


Figure 5.3: Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM) and the standard vector space model (VSM), in the TREC-8 test collection.

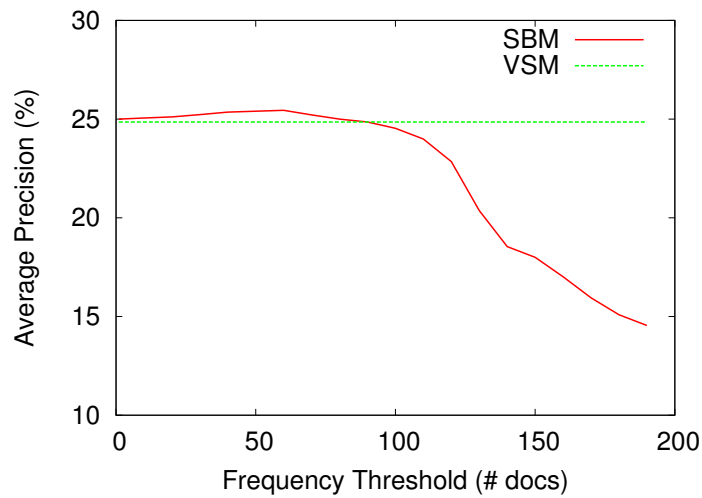


Figure 5.4: Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM) and the standard vector space model (VSM), in the WBR-99 test collection.

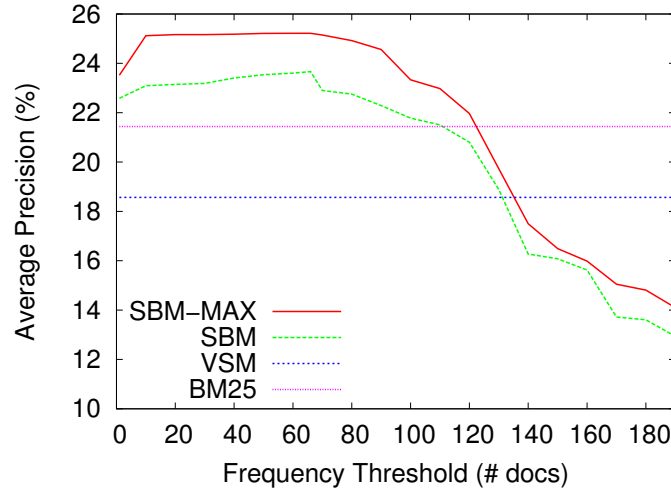


Figure 5.5: Impact on average precision of varying the minimal frequency threshold for the set-based model (SBM), the maximal set-based model (SBM-MAX), the probabilistic model (BM25), and the standard vector space model (VSM) in the WBR-04 test collection.

5.2.2 Minimal Proximity Evaluation

We also verified how variations in the minimal proximity threshold affect average precision. We performed a series of executions of the proximity set-based model with the 15 disjunctive queries in our training set in which the minimal proximity threshold was varied from 1 to 190 for the following test collections: CFC, CISI, TREC-8 and WBR-99. The WBR-04 test collection was used only for the evaluation of our approach for automatically query structuring, which does not take into account the proximity information. The results are illustrated in Figures 5.6, 5.7, 5.8, and 5.9. We observe that the proximity set-based model is significantly affected by the minimal proximity threshold, and the behavior is quite similar for all collections. Initially, an increase in the minimal proximity results in better precision, with maximum precision being achieved for minimal proximity values of 57 for the CFC, 81 for the CISI, 70 for the TREC-8, and 60 for the WBR-99 collection. These are the values used in our experimentation in Section 5.3.

When we increase the minimal proximity beyond those values, we observe that the precision decreases until it reaches almost the mean average precision obtained by the set-based model. This behavior can be explained as follows. First, an increase in the minimal proximity implies an increase in the number of termsets evaluated, what yields better precision. When the minimal proximity increases beyond the best found values, the number of termsets representing weak correlations (i.e., correlations among terms that are separated far apart from each other) increases, leading to a reduction in average precision figures.

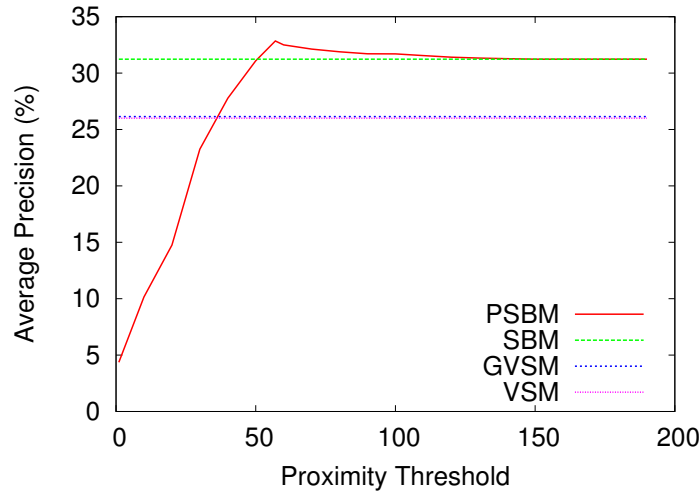


Figure 5.6: Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CFC test collection.

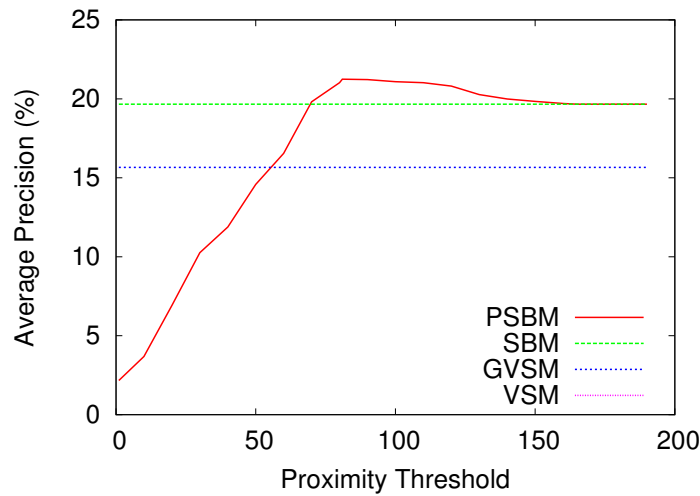


Figure 5.7: Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM), the generalized vector space model (GVSM), and the standard vector space model (VSM), in the CISI test collection.

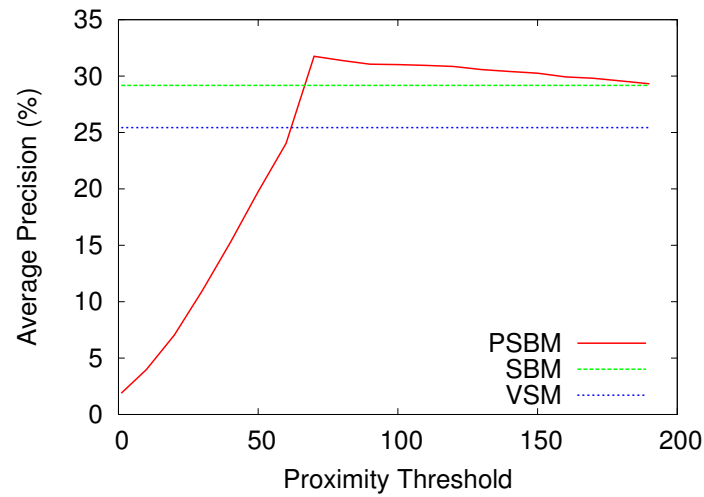


Figure 5.8: Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM) and the standard vector space model (VSM), in the TREC-8 test collection.

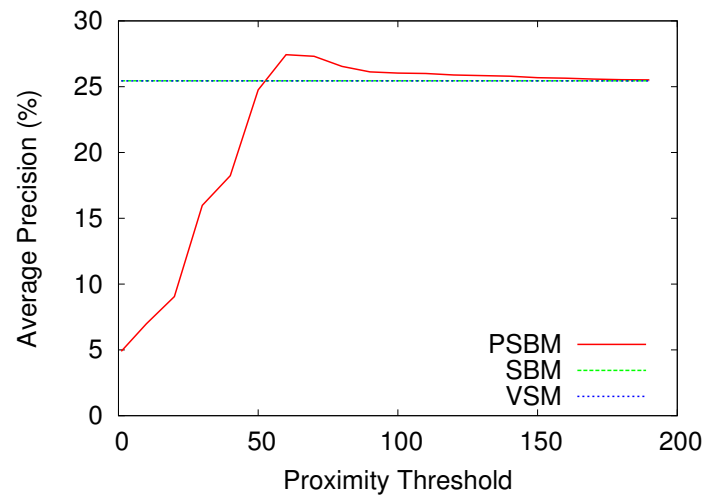


Figure 5.9: Impact on average precision of varying the minimal proximity threshold for the proximity set-based model (PSBM), for the set-based model (SBM) and the standard vector space model (VSM), in the WBR-99 test collection.

5.2.3 Normalization Evaluation

Long and verbose documents include repeated references to a same term. As a result, term frequencies tend to be higher for these documents. This increases the likelihood that a long document will be retrieved by a user query. To avoid this undesirable effect, document length normalization is used. It provides a way of penalizing long documents. Various normalization techniques have been used in information retrieval systems, especially with the standard vector space model.

We discuss experimental results of applying several popular term-based normalization techniques, and a termset-based normalization technique to the set-based model. These normalization techniques are as follows.

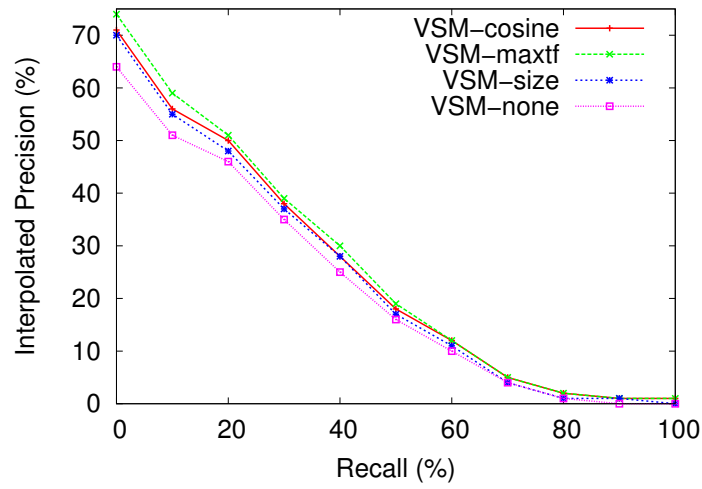
Technique 1 *Cosine* normalization is the most commonly used normalization technique in the vector space model. In our case, every termset weight in a document is divided by the Euclidean weighted norm of the document vector. The cosine normalization factor for a document j is computed as $\sqrt{w_{S_1,j}^2 + w_{S_2,j}^2 + \dots + w_{S_t,j}^2}$, where $w_{S_i,j}$ is the weight of 1-termset S_i in document d_j . Termsets absent from a document are considered to have zero weight.

Technique 2 Another popular normalization technique is to first normalize individual tf weights by the maximum tf . The SMART retrieval system Salton and Buckley (1988) augmented tf factor, given by $0.5 + 0.5 \times \frac{tf}{\max_{tf}}$, and the tf weights used in the INQUERY system Turtle and Croft (1990), given by $0.4 + 0.6 \times \frac{tf}{\max_{tf}}$, are examples of such normalization. Using normalized tf coefficients, we apply the cosine normalization factor given by $\sqrt{w_{S_1,j}^2 + w_{S_2,j}^2 + \dots + w_{S_t,j}^2}$.

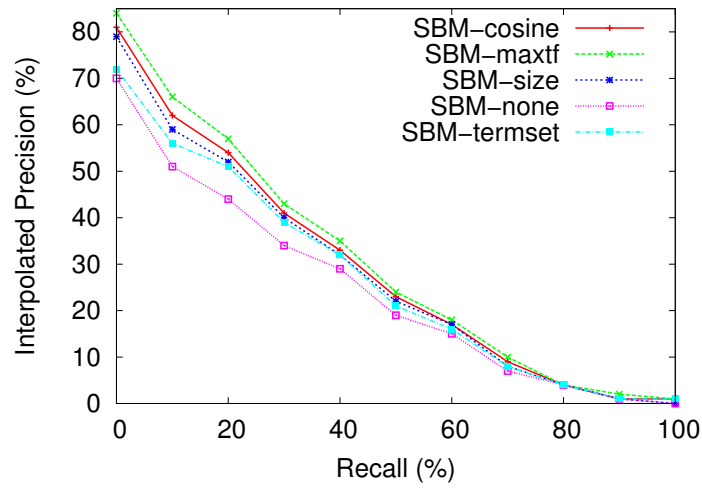
Technique 3 More recently, a length normalization scheme has been used in the Okapi system Robertson et al. (1995), which is based on the byte size of the documents. This scheme does not introduce mutual dependences among the term weights in a document.

Technique 4 We also evaluated the retrieval effectiveness of the vector space and set-based models, when no normalization scheme was used.

Technique 5 *Termset* normalization is one of the possible normalization techniques defined using information on the enumerated termsets for each document and the overall collection. In this case, a document length corresponds to its number of unique termsets $|S_{d_j}|$ divided by the number of distinct termsets for the document collection $|S|$. This technique is evaluated only for the CFC and CISI test collections, due to the exponential cost associated with the enumeration of the termsets.

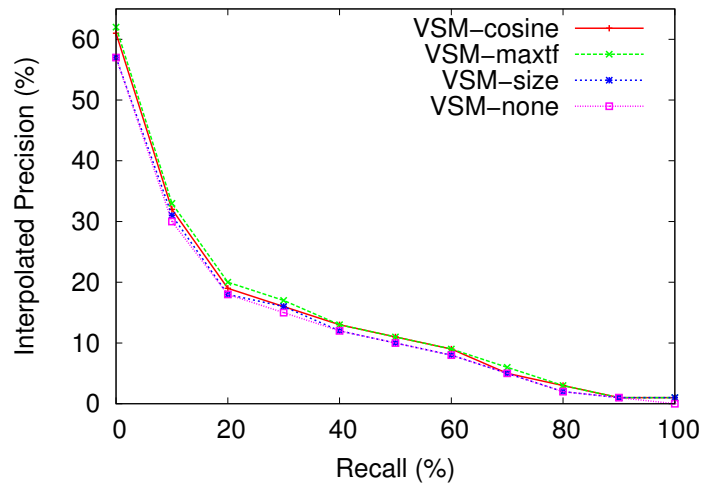


(a) Vector space model (VSM) results.

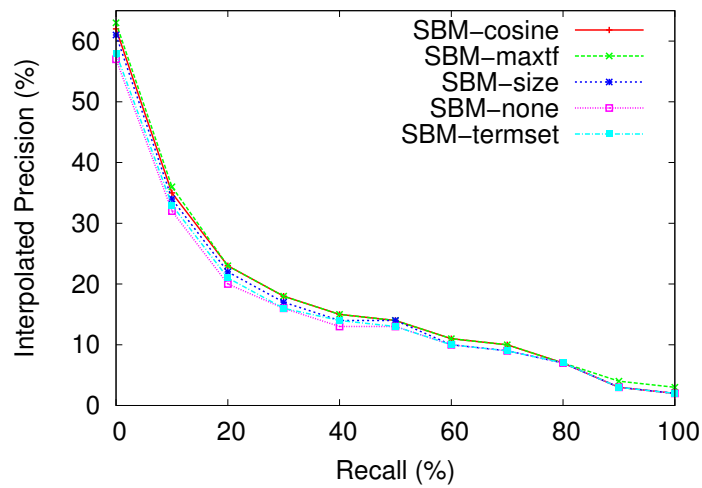


(b) Set-based model (SBM) results.

Figure 5.10: Normalization recall-precision curves for the CFC collection using a training set of 15 queries.

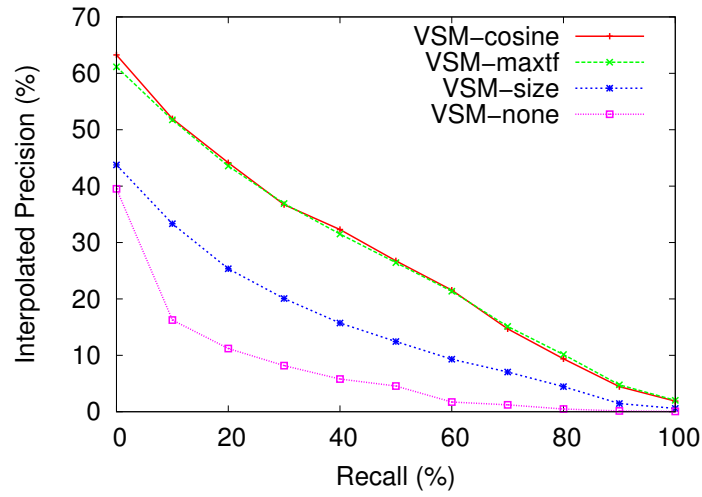


(a) Vector space model (VSM) results.

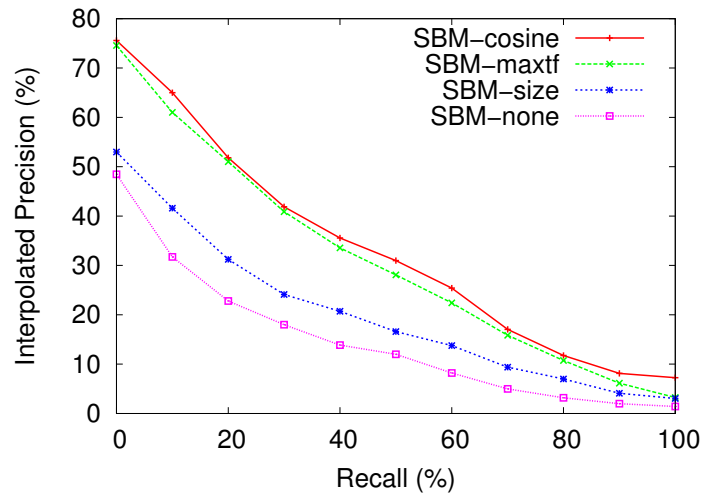


(b) Set-based model (SBM) results.

Figure 5.11: Normalization recall-precision curves for the CISI collection using a training set of 15 queries.

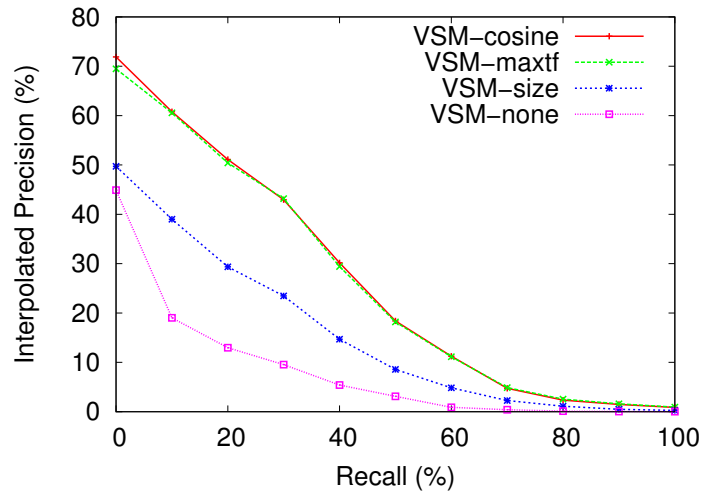


(a) Vector space model (VSM) results.

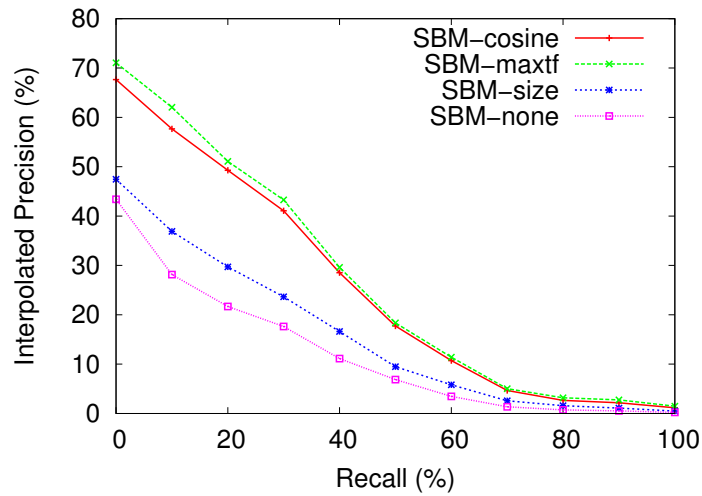


(b) Set-based model (SBM) results.

Figure 5.12: Normalization recall-precision curves for the TREC-8 collection using a training set of 15 queries.

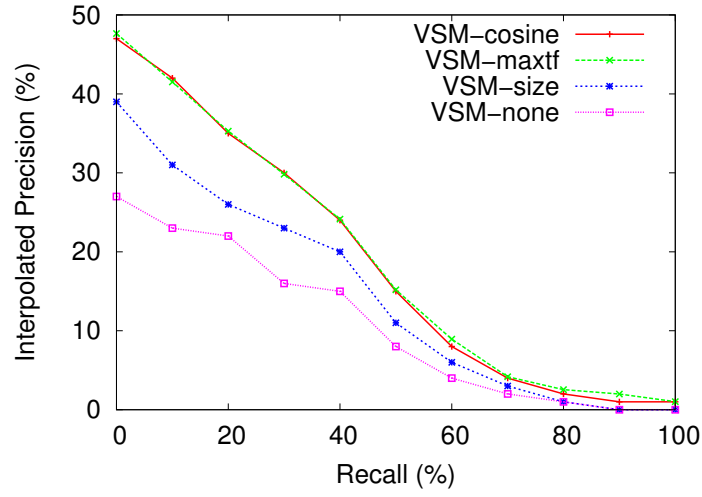


(a) Vector space model (VSM) results.

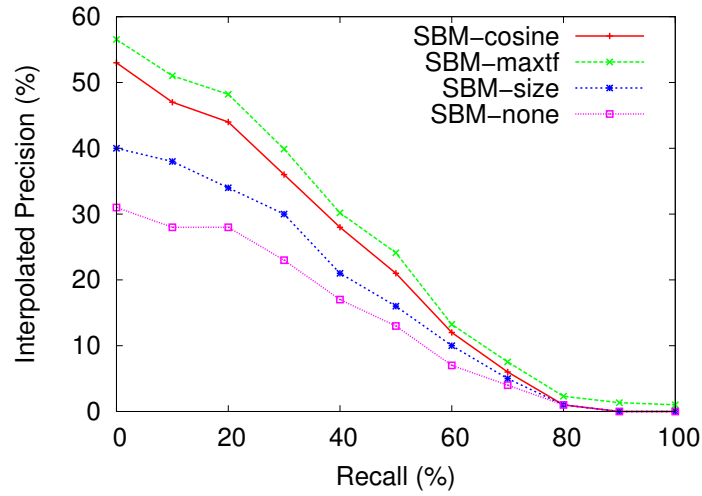


(b) Set-based model (SBM) results.

Figure 5.13: Normalization recall-precision curves for the WBR-99 collection using a training set of 15 queries.



(a) Vector space model (VSM) results.



(b) Set-based model (SBM) results.

Figure 5.14: Normalization recall-precision curves for the WBR-04 collection using a training set of 15 queries.

The experimental results for the vector space model and the set-based model models using Cosine, Maxtf, Size, None, and Termset normalization techniques for processing the 15 queries in our training set are depicted in Figures 5.10, 5.11, 5.12, 5.13, and 5.14. We observe that the effects of the several techniques are analogous in all cases considered. Our results indicate that Maxtf is the method of choice. It is not clear if the Termset normalization technique is better or worse than the other techniques. Thus, we adopt the Maxtf normalization technique in all our experiments and for efficiency reasons, the normalization in our model is based only on 1-termsets. That is, the first order termsets are considered for normalization. This is important because computing the norm of a document using closed termsets might be prohibitively costly.

5.3 Retrieval Effectiveness

In this section we show the retrieval effectiveness for the set-based model and its variants, i.e., the proximity set-based model and the maximal set-based model, for query processing and structuring. Our analysis is based on a comparison to the standard vector space model, to the generalized vector space model, and the probabilistic model using the queries in our test set for each reference collection.

5.3.1 Query Processing

In this section we show experimental results for the evaluation of the set-based model and of the proximity set-based model in terms of retrieval effectiveness for all presented query types (disjunctive, conjunctive and phrase). As mentioned before, our evaluation is based on a comparison to the standard vector space model. For disjunctive queries, we also compare the proposed models effectiveness with the generalized vector space model.

Disjunctive queries

We start our evaluation by verifying the precision-recall curves for each model when applied to each of the test collections. The generalized vector space model could not be evaluated for the TREC-8 and WBR-99 collections, because of the cost of the min-term building phase, which is exponential in the size of the vocabulary, making the computational cost of the associated experiments not feasible.

Figures 5.15, 5.16, 5.17, and 5.18 shows the 11-point average precision figures for the set-based model, the proximity set-based model, the generalized vector space model, and the vector space model algorithms. We observe that the set-based model and the proximity set-based model yield better precision than the vector space model, regardless of the collection

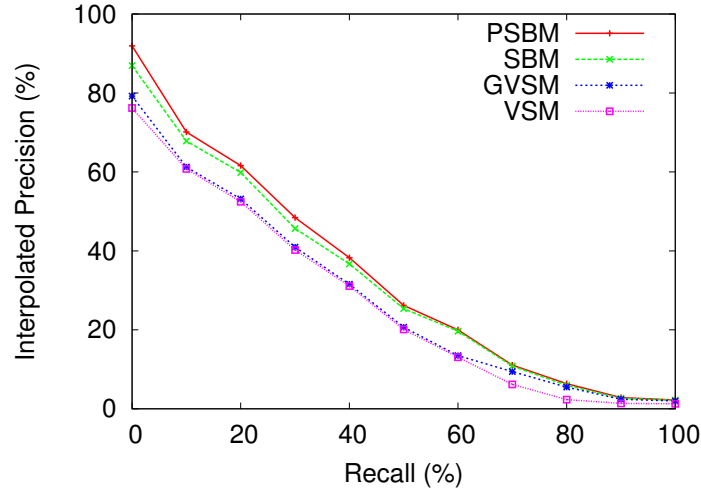


Figure 5.15: Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the CFC test collection, using the test set of sample queries.

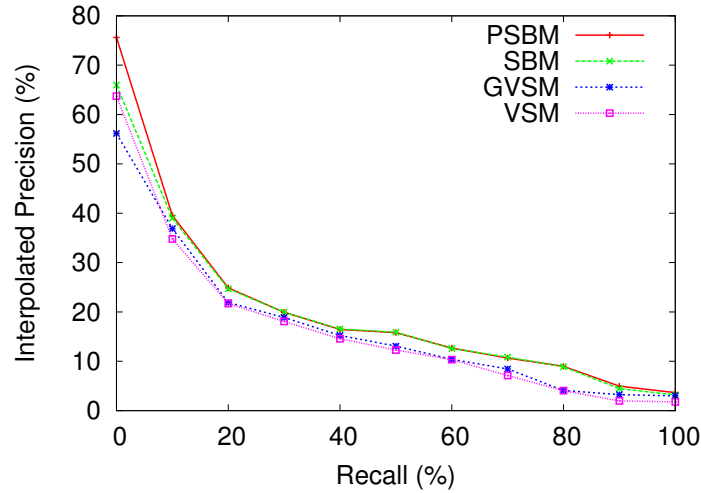


Figure 5.16: Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the CISI test collection, using the test set of sample queries.

and of the recall level. The proximity set-based model ranking yields the largest improvements, which suggests that proximity information can be of value. Further, we observe that the improvements are larger for the TREC-8 collection, because its larger queries allow computing a more representative set of closed termsets. The set-based model and the proximity set-based model also outperforms the generalized vector space model, showing that correlation among query terms can be successfully used to improve retrieval effectiveness.

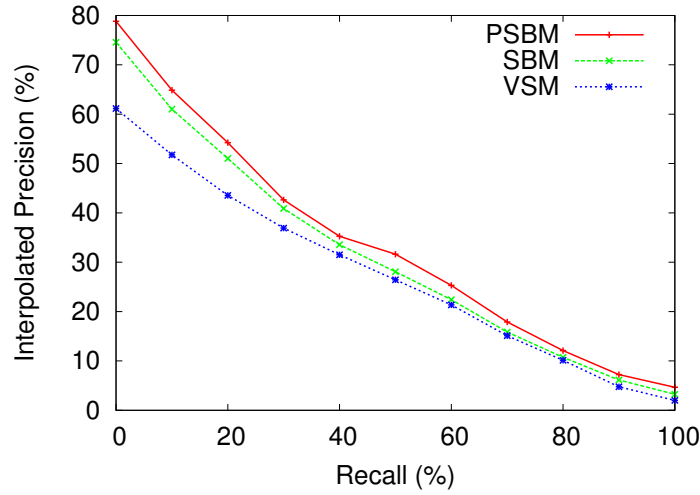


Figure 5.17: Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the TREC-8 test collection, using the test set of sample queries.

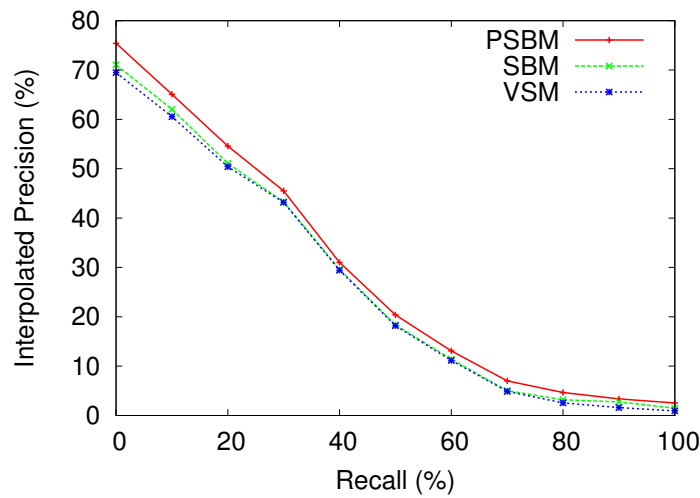


Figure 5.18: Precision-recall curves for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) when disjunctive queries are used, with the WBR-99 test collection, using the test set of sample queries.

Detailed average precision figures are presented in Tables 5.2, 5.3, 5.4 and 5.5. Let us examine first the results for the CFC collection. At the top 10 documents the proximity set-based model provided a very significant gain in average precision of 21.34%, relative to the vector space model, and 16.62% relative to the generalized vector space model. This result clearly shows that termset information can be used to considerably improve retrieval effectiveness. The set-based model, which disposes with information on proximity, also provides a significant gain of 16.74% and 12.20%, respectively.

Let us examine now the results for the CISI collection. At the top 10 documents the proximity set-based model provided a very significant gain in average precision of 21.21%, relative to the vector space model, and 18.60% relative to the generalized vector space model. This result clearly shows that termset information can be used to considerably improve retrieval effectiveness. The set-based model, which disposes with information on proximity, also provides a significant gain of 16.13% and 13.63%, respectively.

Let us examine now the results for the TREC-8 collection. At the top 10 documents the proximity set-based model provided a very significant gain in average precision of 35.29%, relative to the vector space model. This result clearly shows that termset information can be used to considerably improve retrieval effectiveness. The set-based model, which disposes with information on proximity, also provides a significant gain of 28.22%.

Let us examine now the results for the WBR-99 collection. At the top 10 ranked documents, the gains in average precision were of 2.76% for the set-based model and of 10.66% for the proximity set-based model. That is, with very short queries termset information is not very useful because the number of termsets in the user query is too small. Even though, if proximity information is factored in, consistent gains in average precision are observed.

Detailed average precision difference comparison is presented in Table 5.6. This table summarizes two distinct results. The first result is the count of queries where there is a difference between two retrieval techniques. This is expressed in terms of the proportion of queries that differ. The second result is the test for the statistical significance; significant differences are showed in bold.

Retrieval based on the set-based and the proximity set-based models was found to be significantly better than the vector space model for the CFC, the CISI, and the TREC-8 test collections. Regarding the generalized vector space model, our models was also found to be significantly better for the CFC and the CISI test collections. For the WBR-99 test collection, the distinction between set-based model and the standard vector space model is not clear. There is a significant difference between the proximity set-based model and the other evaluated models. However, no significant differences was found when the set-based model is directly compared with the vector space model. That is, with very short queries termset information is not very useful because the number of termsets in the user query is too small, and the majority of the enumerated termsets corresponds to 1-termsets.

CFC - Disjunctive Queries

Level	Precision (%)				Gain (%)		
	<i>VSM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>
At 5 docs.	75.20	81.22	89.04	93.50	8.01	18.40	24.34
At 10 docs.	60.40	62.84	70.51	73.29	4.04	16.74	21.34
At 15 docs.	51.98	52.13	60.54	62.84	0.29	16.47	20.89
At 20 docs.	46.60	47.56	54.17	56.31	2.06	16.24	20.84
At 30 docs.	39.44	40.78	46.09	48.51	3.40	16.86	23.00
At 100 docs.	20.58	22.56	24.16	24.91	9.62	17.40	21.04
At 200 docs.	13.98	15.67	16.55	17.02	12.09	18.38	21.75
At 500 docs.	8.58	10.18	10.45	10.58	18.65	21.79	23.31
At 10000 docs.	6.96	8.35	8.68	8.75	19.97	24.71	25.72
Average	27.37	29.05	33.06	34.45	6.13	20.78	25.86

Table 5.2: CFC document level average figures for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.

CISI - Disjunctive Queries

Level	Precision (%)				Gain (%)		
	<i>VSM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>
At 5 docs.	60.80	55.61	70.05	74.20	-0.09	15.22	22.04
At 10 docs.	40.61	41.50	47.16	49.22	2.19	16.13	21.21
At 15 docs.	32.28	32.64	37.56	39.66	1.12	16.34	22.85
At 20 docs.	26.72	26.91	30.99	32.64	0.71	16.00	22.14
At 30 docs.	23.36	23.87	26.87	28.11	2.18	15.04	20.33
At 100 docs.	14.16	14.25	16.31	17.12	0.64	15.19	20.92
At 200 docs.	9.88	9.90	11.45	11.87	0.20	15.85	20.10
At 500 docs.	5.80	6.08	6.71	7.04	4.83	15.77	21.42
At 10000 docs.	4.24	4.36	4.90	5.15	2.83	15.55	21.40
Average	17.31	17.40	20.18	21.20	0.51	16.58	22.47

Table 5.3: CISI document level average figures for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.

TREC-8 - Disjunctive Queries					
Level	Precision (%)			Gain (%)	
	<i>VSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>SBM</i>	<i>PSBM</i>
At 5 docs.	46.50	61.45	63.87	32.15	37.35
At 10 docs.	43.75	56.10	59.19	28.22	35.29
At 15 docs.	40.23	50.74	52.72	26.12	31.04
At 20 docs.	38.45	47.33	49.82	23.09	29.57
At 30 docs.	35.61	42.41	44.72	19.09	25.58
At 100 docs.	24.41	26.45	29.17	08.35	19.50
At 200 docs.	17.91	18.82	20.99	05.08	17.19
At 500 docs.	10.25	10.26	11.61	00.09	13.26
At 1000 docs.	05.19	05.29	05.76	01.92	10.98
Average	25.44	29.17	31.76	14.66	24.84

Table 5.4: TREC-8 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.

WBR-99 - Disjunctive Queries					
Level	Precision (%)			Gain (%)	
	<i>VSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>SBM</i>	<i>PSBM</i>
At 5 docs.	47.99	50.03	54.00	04.25	12.52
At 10 docs.	45.21	46.46	50.03	02.76	10.66
At 15 docs.	41.15	41.55	45.47	00.97	10.49
At 20 docs.	39.12	39.47	42.52	00.89	08.69
At 30 docs.	35.99	36.14	38.01	00.41	05.61
At 100 docs.	21.79	22.01	23.10	00.55	06.01
At 200 docs.	12.58	12.92	13.22	01.91	05.08
At 500 docs.	08.55	08.79	09.02	01.64	05.49
At 1000 docs.	03.16	03.25	03.35	01.27	06.01
Average	24.85	25.44	37.43	02.37	10.38

Table 5.5: WBR-99 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries.

Disjunctive Queries					
Collection	Statistical Significance				
	<i>SBM/VSM</i>	<i>SBM/GVSM</i>	<i>PSBM/VSM</i>	<i>PSBM/GVSM</i>	<i>PSBM/SBM</i>
CFC	40/21	35/15	50/16	41/17	32/11
CISI	30/18	35/17	43/22	47/14	33/14
TREC-8	58/18	-	79/11	-	64/20
WBR-99	40/15	-	61/10	-	55/26

Table 5.6: Comparison of average precision of the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) with disjunctive queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.

Conjunctive queries

We now discuss the precision-recall results when conjunctive query processing is considered. The minimal frequency threshold is set to 1. The minimal proximity threshold values are set according to the tuning discussed in Section 5.2.2

As we can see in Figures 5.19 and 5.20, the set-based model and the proximity set-based model yield better precision than the vector space model, regardless of the collection and of the recall level. As in the case of disjunctive queries, the proximity set-based model yields the highest results. Also, as before, the improvements are larger in the TREC-8 collection because its larger queries allow computing a more representative set of closed termsets.

Accounting for correlations never degrades the quality of the response sets. In fact, as presented in Tables 5.7 and 5.8, the set-based model yields improvements in average precision of 16.38% and 7.29% for the TREC-8 and the WBR-99 collections, respectively. For the proximity set-based model the improvements are of 29.96% and 11.04% for the TREC-8 and the WBR-99 collections, respectively. At the top 10 ranked documents, the gains in precision are higher. For instance, the gains in average precision were 7.79% (WBR-99) and 27.63% (TREC-8) for the set-based model, and 11.89% (WBR-99) and 30.80% (TREC-8) for the proximity set-based model.

Table 5.9 shows the statistical significance tests for the evaluated models. The proximity set-based model was found to be significantly better than the vector space model and the set-based model for both test collections (TREC-8 and WBR-99). The results found for the set-based model was significantly better than the standard vector space model only for the TREC-8 test collection. As mentioned before, the very short queries found in the WBR-99 collection directly affects the results for the set-based model.

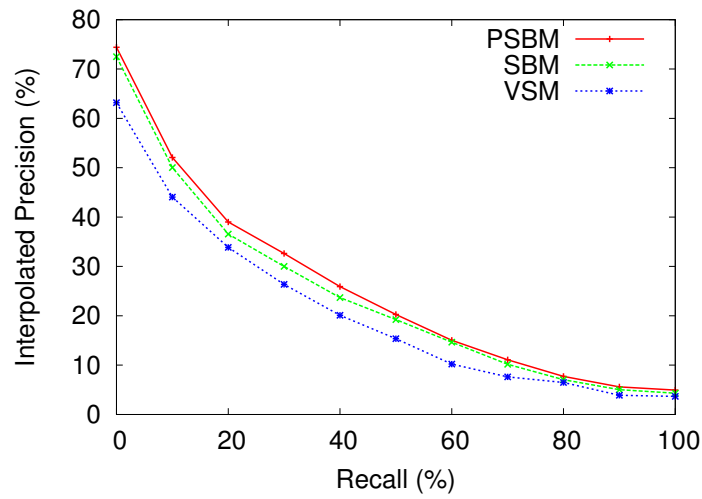


Figure 5.19: Precision-recall curves for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) when conjunctive queries are used, with the TREC-8 test collection, using the test set of sample queries.

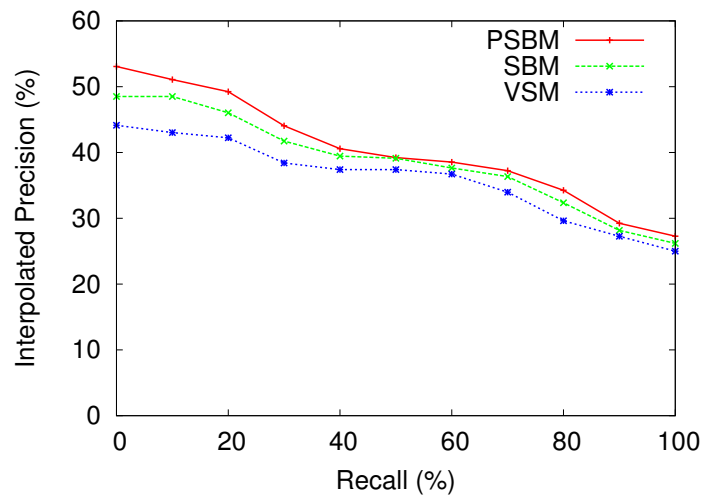


Figure 5.20: Precision-recall curves for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) when conjunctive queries are used, with the WBR-99 test collection, using the test set of sample queries.

TREC-8 - Conjunctive Queries					
Level	Precision (%)			Gain (%)	
	VSM	SBM	PSBM	SBM	PSBM
At 5 docs.	46.71	60.45	62.47	29.42	33.74
At 10 docs.	43.80	55.90	57.29	27.63	30.80
At 15 docs.	39.98	50.13	51.92	25.39	29.86
At 20 docs.	35.02	44.53	45.79	27.16	30.75
At 30 docs.	33.23	41.12	43.16	23.74	29.88
At 100 docs.	22.12	25.95	28.02	17.31	26.67
At 200 docs.	13.45	15.95	16.99	18.59	26.32
At 500 docs.	7.78	9.07	9.81	16.58	26.09
At 1000 docs.	3.56	4.09	4.40	14.89	25.28
Average	19.96	23.23	25.94	16.38	29.96

Table 5.7: TREC-8 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries.

WBR-99 - Conjunctive Queries					
Level	Precision (%)			Gain (%)	
	VSM	SBM	PSBM	SBM	PSBM
At 5 docs.	46.71	50.45	52.25	08.01	11.86
At 10 docs.	43.91	47.33	49.13	07.79	11.89
At 15 docs.	35.01	37.34	38.61	06.66	10.28
At 20 docs.	31.53	33.61	34.99	06.60	10.97
At 30 docs.	29.64	31.44	32.37	06.07	09.21
At 100 docs.	28.76	30.01	30.85	04.35	07.27
At 200 docs.	27.89	29.05	29.39	04.16	05.38
At 500 docs.	22.61	23.22	23.99	02.70	06.10
At 1000 docs.	21.99	22.62	22.97	02.86	04.46
Average	33.60	36.05	37.31	7.29	11.04

Table 5.8: WBR-99 document level average figures for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries.

Conjunctive Queries			
Collection	Statistical Significance		
	<i>SBM/VSM</i>	<i>PSBM/VSM</i>	<i>PSBM/SBM</i>
TREC-8	57/20	80/12	65/24
WBR-99	53/13	62/10	57/28

Table 5.9: Comparison of average precision of the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) with conjunctive queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.

Phrase queries

We now discuss our results when phrase query processing is used. In this case, the proximity set-based model corresponds to the set-based model, since the minimal proximity threshold must be used. The results were obtained by setting the minimal frequency and the minimal proximity thresholds to 1.

As we can see in Figures 5.21 and 5.22, the set-based model yields better precision than the vector space model, regardless of the collection and of the recall level. Tables 5.10 and 5.11 detail the numeric figures. We observe that the set-based model yields improvements in average precision of 17.51% and 8.93% for the TREC-8 and the WBR-99 collections, respectively. At the top 10 ranked documents, the gains in precision are higher. For instance, for the top 10 documents the gains in average precision were 9.92% (WBR-99) and 18.87% (TREC-8) for the set-based model.

Table 5.12 shows the statistical significance tests for the evaluated models. The set-based model was found to be significantly better than the vector space model and the set-based model for both test collections (TREC-8 and WBR-99).

5.3.2 Query Structuring

In this section we report results for our approach for automatically structuring queries, the set-based model using the BM25 weighting scheme, the probabilistic model using the BM25 weighting scheme, and the vector space model for processing conjunctive queries. The BM25 parameters were set with the following values: $k_1 = 1.2$, $k_2 = 0$, $k_3 = 1000$, and $b = 0.75$. Maximum precision is reached for a minimal frequency threshold equal to 10 documents for the TREC-8 collection, and equal to 150 documents for the WBR-04 collection (see Section 5.2.1).

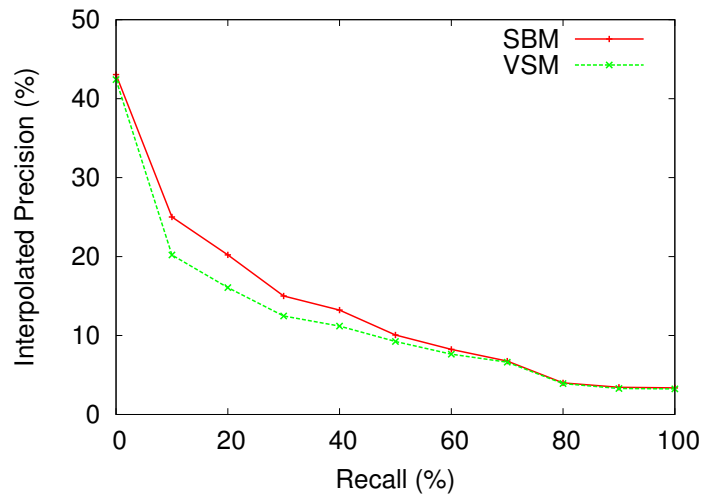


Figure 5.21: Precision-recall curves for the vector space model (VSM) and the set-based model (SBM) when phrase queries are used, with the TREC-8 test collection, using the test set of sample queries

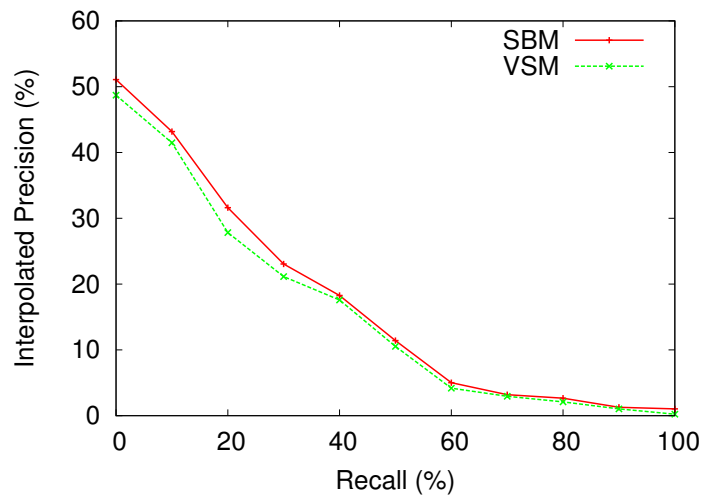


Figure 5.22: Precision-recall curves for the vector space model (VSM) and the set-based model (SBM) when phrase queries are used, with the WBR-99 test collection, using the test set of sample queries

TREC-8 - Phrase Queries			
Level	Precision (%)		Gain (%)
	<i>VSM</i>	<i>SBM</i>	
At 5 docs.	32.33	38.76	19.89
At 10 docs.	26.12	31.05	18.87
At 15 docs.	20.34	24.01	18.04
At 20 docs.	15.25	17.55	15.08
At 30 docs.	12.29	13.89	13.02
At 100 docs.	9.02	9.72	07.76
At 200 docs.	6.46	6.93	07.28
At 500 docs.	2.89	3.04	05.19
At 1000 docs.	1.17	1.21	03.42
Average	11.59	13.62	17.51

Table 5.10: Document level average figures for the vector space model (VSM) and the set-based model (SBM) relative to the TREC-8 test collection, when phrase queries are used.

WBR-99 - Phrase Queries			
Level	Precision (%)		Gain (%)
	<i>VSM</i>	<i>SBM</i>	
At 5 docs.	45.88	50.66	10.42
At 10 docs.	39.13	43.01	09.92
At 15 docs.	28.85	31.58	09.46
At 20 docs.	17.32	18.81	08.60
At 30 docs.	15.69	16.93	07.90
At 100 docs.	13.83	14.80	07.01
At 200 docs.	8.49	9.04	06.48
At 500 docs.	4.63	4.92	06.26
At 1000 docs.	1.99	2.11	06.03
Average	15.11	16.46	08.93

Table 5.11: Document level average figures for the vector space model (VSM) and the set-based model (SBM) relative to the WBR-99 test collection, when phrase queries are used.

Phrase Queries	
Collection	Statistical Significance
	<i>SBM/VSM</i>
TREC-8	60/28
WBR-99	55/15

Table 5.12: Comparison of average precision of the vector space model (VSM) and the set-based model (SBM) with phrase queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.

Figures 5.23 and 5.24 show the 11-point average precision for the evaluated ranking methods. We observe that maximal set-based model and the set-based model yield better precision than the vector space and the probabilistic models, regardless of the collection and of the recall level. Our approach yields the largest improvements, what shows that structured queries outperform bag-of-words queries because they capture some of the relational structure normally expressed in natural language texts. Further, we observe that the improvements are larger for the TREC-8 collection, because its larger queries allow computing a more representative set of maximal termsets.

Detailed average precision figures are presented in Tables 5.13 and 5.14. Let us examine first the results for the TREC-8 collection. At the top 10 documents the set-based model provided a nice gain of 27.51% relative to the vector model, and 16.59% relative to the probabilistic model, while the maximal set-based model boosted this gain to 31.93% and to 20.64%, respectively. The maximal set-based model outperforms the set-based model because it takes into account just the co-occurrence patterns that represent meaningfully “entities” found in the document collections (maximal termsets). Those “entities” may correspond to linguistic constructs (e.g., noun phrases) or other valid relationship captured by statistical constructs.

For the WBR-04 test collection at the 10 top documents, while the set-based model yields a gain of 28.03% relative to the vector model, and 10.25% relative to the probabilistic model, our approach leads to a gain of 40.56% and of 21.05%, respectively. That is, our query structuring mechanism is also useful in the context of the Web. This might be important because complex queries tend to lead to more specific Web pages that are not well-known by the general public. This means that the number of links to these pages tends to be small, making ranking based on link analysis less effective. In this scenario, the gains provided by our approach might represent the difference between a good and a bad answer set.

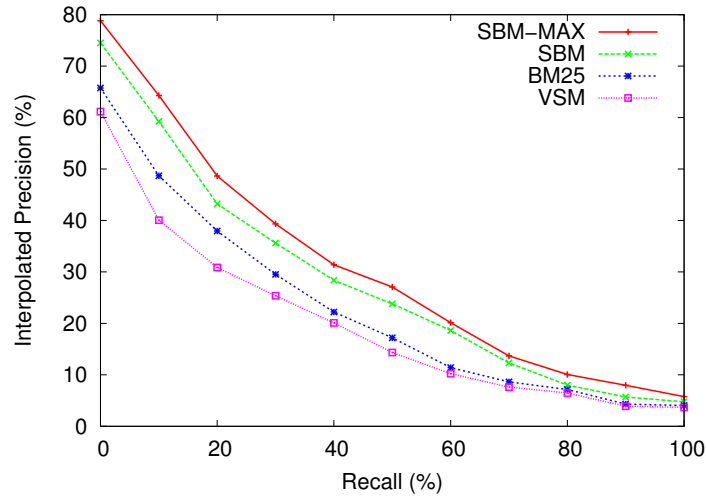


Figure 5.23: Precision-recall curves for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structures queries are used, with the TREC-8 test collection, using the test set of sample queries.

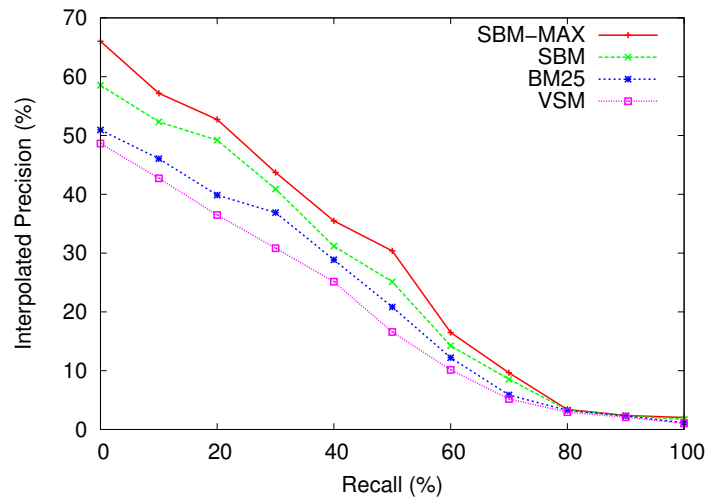


Figure 5.24: Precision-recall curves for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structures queries are used, with the WBR-04 test collection, using the test set of sample queries.

TREC-8 - Structured Queries							
Level	Precision (%)				Gain (%)		
	VSM	BM25	SBM	SBM-MAX	VSM	BM25	SBM
At 5 docs.	46.71	50.77	60.63	63.23	35.36	24.53	4.28
At 10 docs.	43.80	47.90	55.85	57.79	31.93	20.63	3.47
At 15 docs.	39.98	43.91	50.02	53.09	32.79	20.90	6.13
At 20 docs.	35.02	38.40	45.52	46.43	32.58	20.92	2.00
At 30 docs.	33.23	35.90	43.11	43.93	32.20	22.36	1.90
At 100 docs.	22.12	24.11	27.99	29.29	32.43	21.50	4.65
At 200 docs.	13.45	14.75	16.38	18.09	34.53	22.65	10.47
At 500 docs.	7.78	8.53	9.68	10.40	33.62	21.87	7.39
At 1000 docs.	3.56	3.90	4.21	4.80	34.87	23.26	14.05
Average	19.96	21.95	25.44	26.89	34.71	22.50	5.69

Table 5.13: TREC-8 document level average figures for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structured queries are used.

WBR-04 - Structured Queries							
Level	Precision (%)				Gain (%)		
	VSM	BM25	SBM	SBM-MAX	VSM	BM25	SBM
At 5 docs.	54.77	63.11	70.71	78.08	42.55	23.72	10.41
At 10 docs.	43.23	50.20	55.35	60.77	40.56	21.04	9.77
At 15 docs.	35.55	41.36	46.19	49.98	40.58	20.84	8.20
At 20 docs.	28.83	33.44	37.11	40.62	40.90	21.46	9.46
At 30 docs.	23.35	26.86	30.18	32.88	40.83	22.42	8.98
At 100 docs.	18.24	21.01	23.71	25.74	41.13	22.52	8.59
At 200 docs.	13.67	15.84	17.68	19.29	41.12	21.81	9.09
At 500 docs.	9.99	11.57	13.00	14.02	40.42	21.29	7.89
At 1000 docs.	7.69	8.89	9.99	10.69	39.10	20.38	6.98
Average	20.18	23.56	26.13	28.54	41.42	21.13	9.22

Table 5.14: WBR-04 document level average figures for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) when structured queries are used.

Structured Queries			
Collection	Statistical Significance		
	<i>SBM-MAX/VSM</i>	<i>SBM-MAX/BM25</i>	<i>SBM-MAX/SBM</i>
TREC-8	83/12	75/17	62/20
WBR-04	70/21	66/23	60/25

Table 5.15: Comparison of average precision of the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) with structured queries. Each entry has two numbers X and Y (that is, X/Y). X is the percentage of queries where a technique A is better than a technique B. Y is the percentage of queries where a technique A is worse than a technique B. The numbers in bold represent the significant results using the “Wilcoxon’s signed rank test” with a 95% confidence level.

Table 5.15 shows the statistical significance tests for the evaluated models. The maximal set-based model was found to be significantly better than the vector space model, the probabilistic model, and the set-based model for both test collections (TREC-8 and WBR-04) with a 95% confidence level.

5.4 Computational Performance

In this section we show computational performance for the set-based model and its variants, when query response times are considered. This is important because one major limitation of existing models that account for term correlations is their computational cost. Several of these models cannot be applied to large or even mid-size collections, since their costs increase exponentially with the vocabulary size.

5.4.1 Query Processing

In this section we show experimental results for the evaluation of the set-based model and of the proximity set-based model in terms of computational performance for all presented query types (disjunctive, conjunctive and phrase). Our evaluation is based on a comparison to the standard vector space model. For disjunctive queries, we also compare the proposed models with the generalized vector space model.

Disjunctive queries

We determined the average number of closed termsets and the average inverted list sizes for the set-based model and the proximity set-based model for disjunctive query processing. The results, presented in Table 5.16, show that the average case scenario is much better

Collection	# Closed Termsets		Avg. Inverted List Size		
	<i>SBM</i>	<i>PSBM</i>	<i>VSM</i>	<i>SBM</i>	<i>PSBM</i>
CFC	32.58	26.72	145.0	55.48	38.90
CISI	908.59	665.09	90.99	25.04	16.91
TREC-8	581.99	432.55	20,234	6,151	4,189
WBR-99	5.31	4.89	304,101	90,639	66,023

Table 5.16: Average number of closed termsets and inverted list sizes for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM).

Collection	Average Response Time (s)				Increase (%)		
	<i>VSM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>GVSM</i>	<i>SBM</i>	<i>PSBM</i>
CFC	0.0045	0.0118	0.0063	0.0179	162.2	40.0	297.7
CISI	0.0101	0.0455	0.0160	0.0525	350.5	58.4	419.8
TREC-8	0.2314	-	0.3174	0.9691	-	37.16	318.79
WBR-99	0.1179	-	0.1406	0.6265	-	19.25	431.38

Table 5.17: Average response times and response time increases for the vector space model (VSM), the generalized vector space model (GVSM), the set-based model (SBM), and the proximity set-based model (PSBM) for disjunctive query processing.

than the worst case one. For the TREC-8 collection, the average number of closed termsets per query is 581.89 for the set-based model, and 432.55 for the proximity set-based model. Notice that the number of closed termsets is smaller than the worst case $2^{\lceil q \rceil} = 2048$, where $q = 10.80$.

Table 5.17 displays the response time for disjunctive query processing. We also calculated the increase in response time for the set-based and the proximity set-based models, when compared to the vector space model and the generalized vector space model. The generalized vector space model could not be evaluated for the TREC-8 and WBR-99 collections, because of the cost of the min-term building phase, which is exponential in the size of the vocabulary, making the computational cost of the associated experiments not feasible. We observe that the set-based model computes in times 40.00%, 58.40%, 19.25%, and 37.16% larger than the vector space model for the CFC, the CISI, the WBR-99, and the TREC-8 collections, respectively. These results confirm our complexity analysis, indicating that the set-based model is comparable to vector space model in terms of computational costs. The increases in processing time for the proximity set-based model are much larger, 297.7% for the CFC, 419.8% for the CISI, 318.79% for the TREC-8, and 431.38% for the WBR-99. As expected, the increase for the generalized vector space model is much greater, ranging from 162.2% to 350.5% for the CFC and CISI collections respectively.

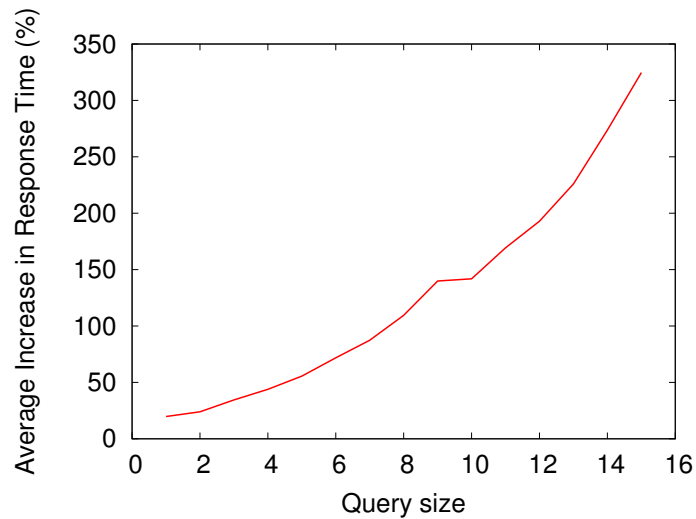


Figure 5.25: Impact of query size on average response time in the WBR-99 for the set-based model (SBM).

We identify two main reasons for the relatively small increase in execution time for the set-based model. First, there is a small number of query related termsets in the reference collections. As a consequence, the inverted lists associated tend to be small (as presented in Table 5.16) and are usually manipulated in main memory in our implementation. Second, we employ pruning techniques that discard irrelevant termsets early in the computation. The main reason for the increase in execution time for the proximity set-based model is the size of the positional index. For instance, while the TREC-8 inverted list file has approximately 300 megabytes, its positional inverted list file has approximately 800 megabytes.

We also evaluated the average increase in the response time of the set-based model as a function of the number of terms in the query. Figure 5.25 summarizes the results of our experiments using all the 100,000 queries of the WBR-99 collection. The execution time of the set-based model is directly affected by the number of terms in the query for a given threshold. Increases in the number of terms results in increases in the overall execution time. In this case, the execution time is dominated by operations over the inverted lists of the termsets obtained. Both the number of termsets and the size of their inverted lists increase with the query size. Figure 5.26 shows the query size distribution for all the 100,000 queries of the WBR-99 collection. The number of queries with more than 8 terms is very small. Variations on the popularity and the correlation between the query terms can explain the small increase for the execution time for the queries containing 10 terms, when compared with the queries with 9 terms.

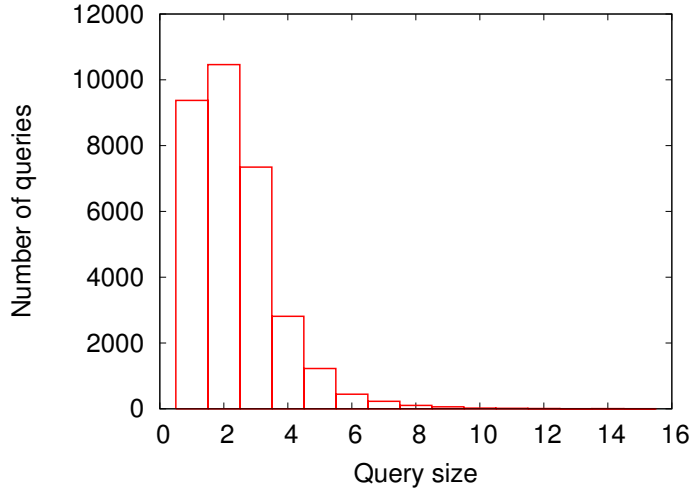


Figure 5.26: Query size distribution for the WBR-99.

Collection	Average Response Time (s)			Increase (%)	
	<i>VSM</i>	<i>SBM</i>	<i>PSBM</i>	<i>SBM</i>	<i>PSBM</i>
TREC-8	0.0478	0.0637	0.1973	33.26	312.76
WBR-99	0.0414	0.0501	0.1187	21.01	186.71

Table 5.18: Average response times and response time increases for the vector space model (VSM), the set-based model (SBM), and the proximity set-based model (PSBM) for conjunctive query processing.

Conjunctive queries

Table 5.18 displays the response time and the increase in response time for the set-based model and the proximity set-based model, when compared to vector space model for conjunctive query processing. All 100,000 queries submitted to the *TodoBR* search engine, excluding the unique term queries, were evaluated for the WBR-99 collection. We observe that the set-based model computes in times 21.01% and 33.26% larger than the vector space model for the WBR-99 and the TREC-8 collections, respectively. The increases in processing time for the proximity set-based model are much larger, 312.76% for TREC-8 and 186.71% for the WBR-99.

Phrase queries

Finally, the response time and the increase in response time for all models and collections considered for phrase query processing is presented in Table 5.19. Only the phrase queries contained in the 100,000 queries submitted to the *TodoBR* search engine were evaluated for the WBR-99 collection. We observe that the set-based model computes in times 18.05% and 22.46% larger than the vector space model for the WBR-99 and the TREC-8 collections, respectively.

Collection	Avg. Response Time (s)		Increase (%)
	VSM	SBM	
TREC-8	0.1073	0.1314	22.46
WBR-99	0.1185	0.1399	18.05

Table 5.19: Average response times and response time increases for the vector space model (VSM) and the set-based model (SBM) for phrase query processing.

Collection	Average Response Time (s)				Increase (%)		
	VSM	BM25	SBM	SBM-MAX	VSM	BM25	SBM
TREC-8	0.6022	0.6141	0.8589	0.7285			
WBR-04	0.5531	0.5609	0.6319	0.6081			

Table 5.20: Average response times and response time increases for the vector space model (VSM), the probabilistic model (BM25), the set-based model (SBM), and the maximal set-based model (SBM-MAX) with the TREC-8 and the WBR-04 test collections.

The execution time of both models, the set-based model and the vector space model, are dominated by operations over the positional inverted lists of the query terms. The small increase in execution time for the set-based model corresponds to the closed termset enumeration phase.

5.4.2 Query Structuring

In this section we compare our approach for automatically structuring queries, the maximal set-based model, to the set-based model using the BM25 weighting scheme, to the probabilistic model using the BM25 weighting scheme, and to the vector space model, when query response times are considered.

Table 5.20 displays the response time for the evaluated approaches. We observe that the set-based model takes execution times 14.24% and 42.62% larger than the vector space model, 12.65% and 39.86% larger than the probabilistic model for the WBR-04 and the TREC-8 collections, respectively. The execution time increase for the maximal set-based model ranges from 9.94 to 20.97% relative to the vector space model, 8.41 to 18.62% relative to the probabilistic model for the WBR-04 and the TREC-8 collections, respectively. The results show that the maximal set-based model outperforms the set-based model considering both retrieval effectiveness and execution time.

The execution times of the set-based model and the maximal set-based model are proportional to the number of terms in the query for a given minimal frequency threshold. The number of termsets depends also on the query size and the minimal frequency threshold employed. Table 5.21 shows the average number of enumerated termsets for the set-based

Collection	Average Number of Termsets	
	<i>SBM</i>	<i>SBM-MAX</i>
TREC-8	285.90	5.54
WBR-04	124.11	1.92

Table 5.21: Average number of termsets for the set-based model (SBM) and the maximal set-based model (SBM-MAX) with the TREC-8 and the WBR-04 reference collections.

model and the maximal set-based model. The operations over the inverted lists of termsets dominate the execution time for these models. The execution times of our approach are lower when compared to the set-based model due to the smaller number of termsets it uses.

5.5 Bibliography Revision

Evaluation of retrieval system effectiveness has been an integral part of the field since its beginning, but can be difficult to do well. Tague catalogs dozens of decisions that are required to design and execute a valid, efficient, and reliable retrieval test (Tague-Sutcliffe, 1981; Tague-Sutcliffe and Blustein, 1992). A common way of simplifying the experimental process is to perform laboratory tests using test collections, a tradition started by the Cranfield tests Cleverdon et al. (1968).

At least two questions remain when constraining retrieval experimentation to laboratory tests using test collections: (i) how to build and validate good test collections, and (ii) what measure(s) should be used to assess the effectiveness of retrieval output.

The first question was addressed by Sparck and van Rijsbergen (1976) who listed a set of criteria that an ideal test collection would meet. The test collections created through the TREC conferences have been validated by demonstrating the stability of relative retrieval scores despite incomplete relevance judgments (Zobel, 1998) and different opinions as to what constitutes a relevant document Voorhees (1998). The works by Zobel (1998) and Cormack et al. (1998) proposed several methods for efficiently building large test collections. Buckley and Voorhees (2000) presents a method for quantifying how the number of requests, the evaluation measure, and the notion of difference used in an information retrieval experiment affect the confidence that can be placed in the conclusions drawn from the experiment.

The second question has received enormous attention in the literature. van Rijsbergen (1979) contains a good summary, while Keen (1992) gives a detailed account on how to present retrieval results. Different evaluation measures have different properties with respect to how closely correlated they are with user satisfaction criteria, how easy they are to interpret, how meaningful aggregates such as average values are, and how much power they have to discriminate among retrieval results.

5.6 Summary

In this chapter we described the four aggregate metrics for measuring retrieval effectiveness used to evaluate the set-based model, and the measure used to establish whether the difference gains obtained by our model is statistically significant. A detailed description of the five reference collections (CFC, CISI, TREC-8, WBR-99, and WBR-04) used was also presented. We also showed a detailed bibliographic discussion providing the most relevant works on dealing, building, and evaluating reference collections.

We evaluate the impact of frequency and proximity thresholds in retrieval effectiveness, and also evaluate several normalization techniques for the set-based model. The discussed fine tuned thresholds were used during retrieval effectiveness and computational performance evaluation for the two types of applications of our model.

We show a complete evaluation of our proposed information retrieval models using several test collections for query processing and query structuring. Regarding retrieval effectiveness, we show that our models are superior for all the collections and query types considered when compared to the standard vector space model. We also show that our models outperforms both, the generalized vector space model and the probabilistic relevance model using the BM25 term weighting scheme.

We also evaluated and validated the set-based model and its variants through computational performance, when query response times are considered. We show that the computational costs allow the use of our set-based model with general large collections. We also show that the number of generated termsets in the average case scenario is much better than the worst case one.

Chapter 6

Conclusions and Future Work

In this chapter, we present a brief summary of the achievements of this work. In Section 6.1, a final analysis of the results is presented and some conclusions are drawn. Following, in Section 6.2, some future work is suggested to complement this work and solve problems left open.

6.1 Thesis Summary

We presented the set-based model, an information retrieval model that uses term correlations to compute term weights. We showed that our model allows significant improvement in retrieval effectiveness, as high as 30%, while keeping extra computational costs small, when compared to the standard vector space model.

To summarize our conclusions, the models proposed and the experiments performed allow us to provide answers to the research questions that motivated this work, as follows:

- Determination of index term weights is derived directly from association rules theory, which naturally considers representative patterns of term co-occurrence (Chapter 3).
- In the set-based model term correlations were not restricted to adjacent, or sentence bounded words, all valid/genuine correlations between the query terms are taken into consideration (Chapter 3).
- Experimental results (Chapter 5) show significant and consistent improvements in average precision curves in comparison to the vector space model and to the generalized vector space model. These improvements in average precision do not always occur in the generalized vector space model because exhaustive application of term co-occurrences to all documents in the collection might eventually hurt the overall effectiveness and performance.

- The set-based model algorithm is practical and efficient for queries containing up to 30 terms, with processing times comparable to the times to process the standard vector space model (Section 5.4).
- Our approach for automatically query structuring (Section 4.6.2) may be used for general document collections, and does not require a syntactic knowledge base, a linguistic processing of the user queries, or a limit in the number of terms in term correlations. It treats the various conjunctive components differently, depending on their support in the document collection - a critical step not addressed by conjunctive normal form (CNF) based approaches.

The set-based model is the first information retrieval model that exploits term correlations and term proximity effectively and provides significant gains in terms of precision, regardless of the size of the collection, of the size of the vocabulary, and query processing type. All known approaches that account for correlation among index terms were initially designed for processing only disjunctive queries. The set-based model provides a simple, effective, efficient, and parameterized way to process disjunctive, conjunctive, phrase, and automatically structured queries.

Although the exploration of the correlation among index terms in the set-based model has showed to be quite effective to provide relevant gains in terms of retrieval effectiveness, it is our believe that precision can be improved even further. This can be accomplished combining other evidences in the ranking calculation. For instance, Web ranking techniques such as Page ranking (Brin and Page, 1998) and the anchor texts could be used together with our content-based ranking algorithm to produce better answers to the users. In the following section we preset suggestions to improve the quality of the results already obtained.

6.2 Future Work

This section shows a list of suggestions for the continuation of the work presented. The list addresses open questions left and present new ideas to extend our work.

Model Refinement

Computation of termsets might be restricted by proximity information. This is useful because proximate termsets carry more semantic information than standard termsets. For future work, the behavior of our query structuring mechanism could be investigated when proximity information of query terms in the documents of the collection is taken into account.

Proximity information, a simple constraint in our model, works as a pruning strategy that limits termsets to those formed by proximate terms. However, it is important to investigate whether the proximity information could be added to the ranking computation in order to classify a termset contribution based on the proximity of its terms.

The set-based model can be easily extended to deal with different kinds of constraints, as minimal proximity. In contrast with most related work accounting for correlations among index terms, the changes to our termsets framework are minimal to accomplish this, since the new constraints can be modeled by the concept of constraint based association rules mining (Srikant et al., 1997).

The closed and maximal termsets allow one to automatically discard term correlations that do not aggregate any additional information of value. Manually built thesaurus, can be used to identify many relationships that can not be automatically extracted from a document collection. Synonymy relationships are such example. The termsets enumeration algorithm can be expanded to use a thesaurus in order to found termsets that express more precise term correlations. This can be accomplished by the use of generalized association rules mining (Srikant and Agrawal, 1995).

Model Performance

Although the set-based model algorithm is practical and efficient (with processing times comparable to the times to process the standard vector space model) for large document collections, the cost to evaluate the cosine measure is potentially high, requiring the reading and processing of whole inverted lists for each enumerated termset. This task may have a high cost once some termsets might occur in a large number of documents in the collection. One of the most effective techniques to compute an approximation of the cosine measure without modification in the final ranking is presented in Anh et al. (2001). It uses thresholds for early recognition of which documents are likely to be highly ranked to reduce costs. The set-based model ranking algorithm can be adapted to use this early termination technique, and several other optimizations. For instance: (i) better inverted lists intersection algorithms, which may reflect how its entries are sorted and stored, (ii) offline enumeration of the termsets, or (iii) a cache for the most frequent enumerated termsets.

Compression has several benefits for information retrieval systems. The space required for storage of text and index is reduced, and less time is required for both index processing and text retrieval. In the set-based model, the inverted lists for the single order termsets are compressed. However, the inverted lists for the termsets generated during query processing, which are stored in main memory, are not compressed. Another interesting problem is to evaluate the behavior of the set-based model in terms of its performance when several

compression techniques are applied, for both single and high order termsets. Compressing techniques can be used to decrease the memory requirement for the termsets enumeration algorithm, allowing the use of our model with queries with more than 30 terms.

Model Formalization

The concept of information is too broad to be captured completely by a single definition. However, the *entropy* has many properties that agree with the intuitive notion of what a measure of information could be. This notion can be extended to define *mutual information*, which is a measure of the amount of information one entity contains about another. It is the reduction in the uncertainty of an entity due to the knowledge of the other. Those concepts, originated from Information Theory, could be used to formalize the vector space defined by the termsets in the set-based model. It is also important to present a theoretical foundation of the association rules framework and this could be done using the mutual information measure, which can be derived from support and confidence.

The set oriented model corresponds to a theoretical framework for representing concepts or co-occurrence patterns that can be used to retrieve any subset of documents of the collection. The set oriented model can also be used to formalize the set-based model, since it clearly defines the bounds for all correlation based approaches.

Automatically Tuning Model Parameters

Finding the ideal values for the set-based model parameters, such as minimal frequency and minimal proximity, is an important problem. Although we studied the effects of such parameters on retrieval effectiveness and computational efficiency, the study of methods to automatically determine them could be investigated. Genetic algorithms could be a way to perform this task.

Creation of New Models

Finally, there are several other applications of the set-based model and its termsets framework. Another interesting problem is to investigate how to apply the termsets framework to the probabilistic and statistical language models, providing a new way to explore the correlation among query terms.

Source Code

All the source code and associated documentation produced for this thesis is freely available for research use only from the web site <http://www.dcc.ufmg.br/gerindo/>.

Bibliography

- Agrawal, R., Aggarwal, C., and Prasad, V. (2000). Depth first generation of long patterns. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 108–118, Boston, MA, USA.
- Agrawal, R., Imielinski, T., and A. Swami (1993a). Database mining: A performance perspective. In *IEEE Transactions on Knowledge and Data Engineering*, volume 5(6), pages 914–925, San Jose, CA.
- Agrawal, R., Imielinski, T., and Swami, A. (1993b). Mining association rules between sets of items in large databases. In Buneman, P. and Jajodia, S., editors, *Proceedings of the ACM SIGMOD International Conference Management of Data*, pages 207–216, Washington, D.C. ACM Press.
- Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. (1996). Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328, San Jose, CA. AAAI/MIT Press.
- Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *The 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile. Morgan Kaufmann Publishers.
- Alsaffar, A. H., Deogun, J. S., Raghavan, V. V., and Sever, H. (2000). Enhancing concept-based retrieval based on minimal term sets. *Journal of Intelligent Information Systems*, 14(2–3):155–173.
- Anh, V. N., Kretser, O., and Moffat, A. (2001). Vector-space ranking with effective early termination. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42, New Orleans, Louisiana, USA. ACM Press.
- Aumann, Y. and Lindell, Y. (1999). A statistical theory for quantitative association rules. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 261–270, San Diego, CA.

- Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley-Longman, Wokingham, UK, 1st edition.
- Bayardo, R. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 85–93.
- Bayardo, R. and Agrawal, R. (1999). Mining the most interesting rules. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154, San Diego, CA.
- Bayardo, R., Agrawal, R., and Gunopulos, D. (1999). Constraint-based rule mining in large, dense databases. In *Proceedings of the 15th International Conference on Data Engineering*, pages 188–197, Sydney, Australia.
- Berger, A. and Lafferty, J. (1999). Information retrieval as statistical translation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, Berkeley, California, USA. ACM Press.
- Billhardt, H., Borrajo, D., and Maojo, V. (2002). A context vector model for information retrieval. *Journal of the American Society for Information Science and Technology*, 53(3):236–249.
- Bollmann-Sdorra, P. and Raghavan, V. V. (1998). On the necessity of term dependence in a query space for weighted retrieval. *Journal of the American Society for Information Science*, 49(13):1161–1168.
- Bookstein, A. (1988). Set oriented retrieval. In *The 11th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 583–596, Grenoble, France.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117.
- Brown, P. F., Cocke, J., Pietra, S. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine learning translation. *Journal of Computational Linguistics*, 16(2):79–85.
- Buckley, C. and Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, Athens, Greece.
- Buell, D. (1981). A general model of query processing in information retrieval systems. *Information Processing & Management*, 17:249–262.

- Burdick, D., Calimlim, M., and Gehrke, J. (2001). MAFIA: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering*, pages 443–452, Washington, DC. IEEE Computer Society.
- Cao, G., Nie, J., and Bai, J. (2005). Integrating word relationships into language models. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–305, Salvador, Bahia, Brazil. ACM Press.
- Cao, J., Nie, J., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 170–177, Sheffield, South Yorkshire, UK. ACM Press.
- Cleverdon, C. W., Mills, J., and Keen, E. M. (1968). Factors determining the performance of indexing systems. In *Two volumes*, Cranfield, England.
- Cormack, G. V., Palmer, C. R., and Clarke, C. L. A. (1998). Efficient construction of large test collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 282–289, Melbourne, Australia. ACM Press.
- Croft, W. B., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45, Chicago, Illinois, USA.
- Das-Gupta, P. (1987). Boolean interpretation of conjunctions for document retrieval. In *Journal of the American Society for Information Science*, volume 38, pages 349–368.
- Elias, P. (1975). Universal code word sets and representations of the integers. In *IEEE Transactions on Information Theory*, volume 21, pages 194–203.
- Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–43, Menlo Park, CA. AAAI Press.
- Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996b). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM – Data Mining and Knowledge Discovery in Databases*, 29(11):27–34.
- Feldman, R. and Dagan, I. (1995). Kdt - knowledge discovery in texts. In *First International Conference on Knowledge Discovery and Data Mining*, pages 112–117, Montreal, Canada.

- Feldman, R. and Hirsh, H. (1997). Exploiting background information in knowledge discovery from text. *Journal of Intelligent Information Systems*, 9(1):83–97.
- Gouda, K. and Zaki, M. J. (2001). Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163–170.
- Gunopulos, D., Mannila, H., and Saluja, S. (1997). Discovering all the most specific sentences by randomized algorithms. In *Proceedings of the 1997 International Conference on Database Theory*, pages 215–229.
- Harper, D. J. and Rijsbergen, C. J. V. (1978). An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation*, 34:189–216.
- Hawking, D., Craswell, N., and Thistlewaite, P. B. (1998). Overview of TREC-7 very large collection track. In Voorhees, E. M. and Harman, D. K., editors, *The Seventh Text REtrieval Conference (TREC-7)*, pages 91–104, Gaithersburg, Maryland, USA. Department of Commerce, National Institute of Standards and Technology.
- Hawking, D., Craswell, N., Thistlewaite, P. B., and Harman, D. (1999). Results and challenges in web search evaluation. *Computer Networks*, 31(11–16):1321–1330. Also in Proceedings of the 8th International World Wide Web Conference.
- Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *European Conference of Digital Libraries*, pages 569–584.
- Holsheimer, M., Kersten, M., Mannila, H., and Toivonen, H. (1995). A perspective on databases and data mining. In *First International Conference on Knowledge Discovery and Data Mining*, pages 150–155, Montreal, Canada.
- Houtsma, M. A. W. and Swami, A. N. (1995). Set-oriented mining for association rules in relational databases. In *ICDE '95: Proceedings of the Eleventh International Conference on Data Engineering*, pages 25–33, Washington, DC, USA. IEEE Computer Society.
- Jelinek, J. (1998). *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.
- Kaszkeil, M. and Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185, Philadelphia, Philadelphia, USA. ACM Press.
- Kaszkeil, M., Zobel, J., and Sacks-Davis, R. (1999). Efficient passage ranking for document databases. *ACM Transactions on Information Systems (TOIS)*, 17(4):406–439.

- Keen, E. M. (1992). Presenting results of experimental retrieval comparisons. *Information Processing & Management*, 28(4):491–502.
- Kim, M., Alsaffar, A. H., Deogun, J. S., and Raghavan, V. V. (2000). On modeling of concept based retrieval in generalized vector spaces. In *Proceedings International Symposium on Methods of Intelligent Systems*, pages 453–462, Charlotte, N.C., USA. Springer-Verlag.
- Lafferty, J. and Zhai, C. (2001). Document language models, query models and risk minimization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, New Orleans, Louisiana, USA. ACM Press.
- Lavrenko, V. and Croft, W. B. (2001). Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, New Orleans, Louisiana, USA. ACM Press.
- Lin, D. I. and Kedem, Z. M. (1998). Pincer-search: A new algorithm for discovering the maximum frequent set. In *Proceedings of the 1998 International Conference on Extending Database Technology*, pages 105–119.
- Liu, B., Hsu, W., and Ma, Y. (1999). Pruning and summarizing the discovered associations. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125–134, San Diego, CA.
- Maron, M. and Kuhns, J. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of ACM*, 7:216–244.
- Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999). A hidden markov model information retrieval system. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221, Berkeley, California, USA. ACM Press.
- Miller, R. and Yang, Y. (1997). Association rules over interval data. In *Proceedings of the ACM SIGMOD International Conference Management of Data*, volume 26(2), pages 452–461, Tucson, Arizona.
- Mitra, M., Buckley, C., Singhal, A., and Cardie, C. (1997). An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97, 5th International Conference Recherche d'Information Assistee par Ordinateur*, pages 200–214, Montreal, CA.
- Nallapati, R. and Allan, J. (2002). Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the 11th international conference on Information and knowledge management*, pages 383–390, McLean, Virginia, USA. ACM Press.

- Narita, M. and Ogawa, Y. (2000). The use of phrases from query texts in information retrieval. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–320, Athens, Greece.
- Paice, C. D. (1984). Soft evaluation of boolean search queries in information retrieval systems. *Information Technology*, 3(1):33–41.
- Park, J., Chen, M., and Yu, P. (1995). An effective hash based algorithm for mining associative rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 175–186, San Jose, CA.
- Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Lecture Notes In Computer Science archive Proceeding of the 7th International Conference on Database Theory, Lecture Notes In Computer Science (LNCS)*, pages 398–416. Springer-Verlag.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference*. Morgan Kaufmann Publishers, 2nd edition.
- Pei, J., Han, J., and Mao, R. (2000). Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, Arlington.
- Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, Melbourne, Australia. ACM Press.
- Pôssas, B., Meira Jr., W., Carvalho, M., and Resende, R. (2000). Using quantitative information for efficient association rule generation. *ACM Sigmod Record*, 29(4):19–25.
- Pôssas, B., Ziviani, N., and Meira Jr., W. (2002a). Enhancing the set-based model using proximity information. In *The 9th International Symposium on String Processing and Information Retrieval*, Lecture Notes in Computer Science, pages 104–116, Lisbon, Portugal. Springer-Verlag.
- Pôssas, B., Ziviani, N., Meira Jr., W., and Ribeiro-Neto, B. (2002b). Modeling co-occurrence patterns and proximity among terms in information retrieval systems. In *The First Seminar on Advanced Research in Electronic Business*, pages 123–131, Rio de Janeiro, Brazil.
- Pôssas, B., Ziviani, N., Meira Jr., W., and Ribeiro-Neto, B. (2002c). Set-based model: A new approach for information retrieval. In *The 25th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, Tampere, Finland. ACM Press.

- Pôssas, B., Ziviani, N., Meira Jr., W., and Ribeiro-Neto, B. (2005a). Maximal termsets as a query structuring mechanism. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-05)*, Bremen, Germany.
- Pôssas, B., Ziviani, N., Meira Jr., W., and Ribeiro-Neto, B. (2005b). Maximal termsets as a query structuring mechanism. Technical Report TR012/2005, Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil.
- Pôssas, B., Ziviani, N., Ribeiro-Neto, B., and Meira Jr., W. (2004). Processing conjunctive and phrase queries with the set-based model. In *The 11th International Symposium on String Processing and Information Retrieval*, Lecture Notes in Computer Science, pages 171–183, Padova, Italy. Springer-Verlag.
- Pôssas, B., Ziviani, N., Ribeiro-Neto, B., and Meira Jr., W. (2005c). Set-based vector model: An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems*, 23(4). To appear.
- Raghavan, V. V. and Yu, C. T. (1979). Experiments on the determination of the relationships between terms. *ACM Transactions on Databases Systems*, 4(2):240–260.
- Ribeiro-Neto, B. and Muntz, R. (1996). A belief network model for IR. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, Zurich, Switzerland.
- Rijsbergen, C. J. V. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:106–119.
- Robertson, S. and Jones, K. S. (1976). Relevance weighting of search terms. *Journal of American Society for Information Science*, 27:129–146.
- Robertson, S., Maron, M. E., and Cooper, W. S. (1982). Probability of relevance: a unification of two competing models for document retrieval. *Information Technology: Research and Development*, 1:1–21.
- Robertson, S. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland. Springer-Verlag.
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. In Voorhees, E. M. and Harman, D. K., editors, *The Third Text REtrieval Conference (TREC-3)*, pages 109–126, Gaithersburg, Maryland, USA. Department of Commerce, National Institute of Standards and Technology.

- Salton, G. (1971). *The SMART retrieval system – Experiments in automatic document processing*. Prentice Hall Inc., Englewood Cliffs, NJ.
- Salton, G. (1992). The state of retrieval system evaluation. *Information Processing & Management*, 28(4):441–449.
- Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic retrieval. *Information Processing & Management*, 24(5):513–523.
- Salton, G., Buckley, C., and Yu, C. T. (1982). An evaluation of term dependencies models in information retrieval. In *The 5th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 151-173, Berlin, Germany. ACM Press.
- Salton, G., Fox, E. A., and Wu, H. (1983). Extended boolean information retrieval. In *Communications of the ACM*, volume 26, pages 1022–1036.
- Salton, G. and Lesk, M. E. (1968). Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36.
- Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1st edition.
- Salton, G. and Yang, C. S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351-372.
- Savasere, A., Omiecinski, E., and Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. In *The 21st International Conference on Very Large Data Bases*, pages 432-444, Zurich, Switzerland.
- Shaw, W. M., Wood, J. B., Wood, R. E., and Tibbo, H. R. (1991). The cystic fibrosis database: Content and research opportunities. In *Library and Information Science Research*, volume 13), pages 347–366.
- Silva, A., Veloso, E., Golgher, P., Ribeiro-Neto, B., Laender, A., and Ziviani, N. (1999). CobWeb - a crawler for the brazilian web. In *Proceedings of the 6th String Processing and Information Retrieval Symposium*, pages 184–191, Cancun, Mexico. IEEE Computer Society.
- Small, H. (1981). The relationship of information science to the social sciences: A co-citation analysis. *Information Processing & Management*, 17(1):39–50.
- Smeaton, A. F. and van Rijsbergen, C. J. (1988). Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In *The 11th ACM-SIGIR*

- Conference on Research and Development in Information Retrieval*, pages 31–51, Grenoble, France.
- Smith, M. E. (1990). *Aspects of the P-Norm Model of Information Retrieval: Syntactic Query Generation, Efficiency and Theoretical Properties*. PhD thesis, Computer Science Department, Cornell University.
- Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the 8th international conference on Information and knowledge management*, pages 316–321, Kansas City, Missouri, United States. ACM Press.
- Sparck, J. K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11-21.
- Sparck, J. K. and van Rijsbergen, C. J. (1976). Information retrieval test collections. *Journal of Documentation*, 32(1):59–75.
- Srihari, R., Niu, C., and Li, W. (1999). Use of maximum entropy in back-off modeling for a named entity tagger. In *Proceedings of the HKK Conference*, pages 159–164.
- Srikant, R. and Agrawal, R. (1995). Mining generalized association rules. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pages 407–419, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *Proceedings of the ACM SIGMOD International Conference Management of Data*, pages 1-12, Montreal, Canada.
- Srikant, R., Vu, Q., and Agrawal, R. (1997). Mining association rules with item constraints. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, KDD*, pages 67–73. AAAI Press.
- Srikanth, M. and Srihari, R. (2002). Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–426, Tampere, Finland. ACM Press.
- Srikanth, M. and Srihari, R. (2003). Exploiting syntactic structure of queries in a language modeling approach to ir. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 476–483. ACM Press.
- Tague-Sutcliffe, J. (1981). The pragmatics of information retrieval experimentation. *Information Retrieval Experiment*, pages 59–102.

- Tague-Sutcliffe, J. and Blustein, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management*, 28(4):467–490.
- Turtle, H. and Croft, W. B. (1990). Inference networks for document retrieval. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, Brussels, Belgium. ACM Press.
- Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222.
- van Rijsbergen, C. J. (1979). *Information Retrieval*. ButterWorths, London, UK, 2nd edition.
- Veloso, A. A., Meira Jr., W., de Carvalho, M. B., Pôssas, B., and Zaki, M. J. (2002). Mining frequent itemsets in evolving databases. In *Second SIAM International Conference on DATA MINING*, Arlington, VA.
- Voorhees, E. and Harman, D. (1999). Overview of the eighth text retrieval conference (trec 8). In Voorhees, E. M. and Harman, D. K., editors, *The Eighth Text REtrieval Conference (TREC-8)*, pages 1–23, Gaithersburg, Maryland, USA. Department of Commerce, National Institute of Standards and Technology.
- Voorhees, E. M. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 315–323, Melbourne, Australia. ACM Press.
- Webb, G. (1995). Opus: An efficient admissible algorithm for unordered search. In *Journal of Artificial Intelligence Research*, volume 3, pages 431–465.
- Witten, I. H., Moffat, A., and Bell, T. C. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, San Francisco, 2nd edition.
- Wong, S. K. M., Ziarko, W., Raghavan, V. V., and Wong, P. C. N. (1987). On modeling of information retrieval concepts in vector spaces. *The ACM Transactions on Databases Systems*, 12(2):299–321.
- Wong, S. K. M., Ziarko, W., and Wong, P. C. N. (1985). Generalized vector space model in information retrieval. In *The 8th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, New York, USA. ACM Press.
- Yu, C. T. and Salton, G. (1976). Precision weighting – an effective automatic indexing method. *Journal of the ACM*, 23(1):76–88.

- Zaki, M. and Hsiao, C. (2002). Charm: An efficient algorithm for closed association rule mining. In *2nd SIAM International Conference on Data Mining*, Arlington.
- Zaki, M. and Ogihara, M. (1999). Theoretical foundations of association rules. Technical report, The University of Rochester Computer Science Department, Rochester, NY.
- Zaki, M., Parthasarathy, S., Ogihara, M., and Li, W. (1997). New algorithms for fast discovery of association rules. In *Third International Conference on Knowledge Discovery and Data Mining*, pages 283-286, Newport Beach, CA.
- Zaki, M. J. (2000). Generating non-redundant association rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34-43, Boston, MA, USA. ACM Press.
- Zhang, Z., Lu, Y., and Zhang, B. (1997). An effective partitioning-combining algorithm for discovering quantitative association rules. In *First Pacific Asia Conference on Knowledge Discovery and Datamining*, pages 241-251, Singapore.
- Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, Melbourne, Australia. ACM Press.
- Zobel, J., Moffat, A., Wilkinson, R., and Sacks-Davis, R. (1995). Efficient retrieval of partial documents. *Information Processing & Management*, 31(3):361–377.