EDUARDO LIQUIO TAKAO

UMA ANÁLISE DE MODELOS E SISTEMAS PROBABILÍSTICOS EM RECUPERAÇÃO DE INFORMAÇÃO EM BASES TEXTUAIS.

Florianópolis - SC

2001

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

EDUARDO LIQUIO TAKAO

ANÁLISE COMPARATIVA DOS MODELOS E SISTEMAS PROBABILÍSTICOS EM RECUPERAÇÃO DE INFORMAÇÃO EM BASES TEXTUAIS.

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciências da Computação.

Prof. Dr. Murilo Silva de Camargo

Florianópolis, outubro de 2001

ANÁLISE COMPARATIVA DOS MODELOS E SISTEMAS PROBABILÍSTICOS EM RECUPERAÇÃO DE INFORMAÇÃO EM BASES TEXTUAIS.

EDUARDO LIQUIO TAKAO

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciências da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciências da Computação

_____

Prof. Dr. Eng. Fernando Ostuni Gauthier

(Coordenador do Curso)

_____

Prof. Dr. Eng. Murilo Silva de Camargo

(Orientador e presidente da banca)

Banca Examinadora

_____

Prof. Dr. Pedro Alberto Barbetta

_____

Prof. Dr. Rosvelter João Coelho da Costa

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pela força e capacitação;
À minha esposa Elisa e minhas filhas Larissa e Joyce, pelo amor, carinho e compreensão;
Ao meu orientador Murilo Silva de Camargo, que proporcionou mais esta vitória;
Aos membros da banca, pela disposição e dedicação;
Às Faculdades Maringá, que me acolheu e deu sustento durante todo o trabalho;
Ao Núcleo de Processamento de Dados da Universidade Estadual de Maringá, que me compreendeu e deu condições de trabalho;
À Igreja Nikkey de Evangelização que sempre orou e me amparou nos momentos difíceis;
Ao professor Iran da JRCursos, que conferiu todas as traduções deste trabalho;
Aos Bibliotecários João Fábio e Edilson Damásio, pelo apoio e conferência do trabalho;
A todos que direta ou indiretamente contribuíram para que este estudo fosse possível.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SIGLAS

BSS- Okapi Basic Search System – Sistema Básico de Procura da Okapi.

CIIR - Center for Intelligent Information Retrieval at the University of Massachussets – Centro de Inteligência em Recuperação de Informação da Universidade de Massachussets

CIW - Combined Interative Weight –Peso Interativo Combinado

CW - Combined Weight – Peso Combinado

DARPA - Defense Advanced Research Projects Agency - Agência de Defesa Avançada de Projetos de Pesquisa.

INTERNET - Rede Mundial de Computadores

IR- Information Retrieval - Recuperação de Informação

IRS - Information Retrieval System – Sistema de Recuperação de Informação

LCA - Local Context Analysis – Análise de Contexto Local

LIMSI - Laboratoire d'Informatique pour la Mécanique et les Sciences de Ingénieur – Laboratório de Processamento de Dados e Mecânicos e Ciências da Engenharia.

NIST - National Institute of Standards and Technology - Instituto Nacional de Padrões e Tecnologia

NLP - Natural Language Processing - . Processo de Linguagem Natural

OW - Offer Weight – Oferta de Peso

Q&A - Question and Answer – Perguntas e Respostas

RVS - Retrieval Status Value – Valor do Estado da Recuperação

TREC - Text Retrieval Conference - Conferência de Recuperação de Texto

TRW - Term Relevance Weigh - Peso de Relevância do Termo

WER - Word Error Average – Erro médio de palavras.

RESUMO

A dificuldade de encontrar uma informação específica, é um dos grandes problemas encontrados hoje em dia. A Recuperação de Informação (IR) é uma área da computação que estuda o desenvolvimento de técnicas para permitir o acesso rápido a uma grande quantidade de informações. Estas informações podem ser: texto, vídeo ou áudio. Dentre os modelos clássicos de IR destacam-se três: Booleano, Vetor Espacial e Probabilístico. Neste trabalho estudar-se-ão os modelos clássicos, em especial os probabilísticos alternativos em IR. Os modelos probabilísticos são baseados no Princípio da Classificação da Probabilidade. Muitos modelos probabilísticos estão sendo estudados, mais um dos grandes problemas é trazer somente o conjunto de informações realmente importantes para a necessidade do usuário. Este trabalho descreve os modelos e sistemas probabilísticos em recuperação de informação textual, com o objetivo de analisar suas características, limitações e resultados, a fim de prover melhorias e contribuir para o aperfeiçoamento dos modelos e sistemas propostos.

Palavras-chaves: Recuperação de Informação, IR; Modelos Clássicos em IR; Modelo Probabilístico; Modelo Alternativo Probabilístico; Sistema Probabilístico.

ABSTRACT

The difficulty of finding a specific information is one of the greatest problem found nowadays. Information Retrieval (IR), it is a computer area which studies the techniques development to permit the fast access to a large information quantity. This information may be: text, video or audio. Among the classical models of IR, there are three most important ones: Boolean, Vector Space and Probabilistic. On this paper they will be studied the classical models, in special the alternative probabilistic ones in IR. The probabilistic models are based on Probability Classification Principle. Many probabilistic models are being studied, but one of the greatest problems is bringing only information set that are really important to the user's necessity. This paper describes a textual probabilistic information retrieval model and systems, with objective of comparing their characteristics, limitations and results, in order to conclude the importance of probability use in IR, and provide improvement to the development of model end systems.

Keywords: Information Retrieval, IR; Classical Models of IR; Probabilistic Model; Alternative Probabilistic Models; Probabilistic System.

1 INTRODUÇÃO

1.1 Apresentação

Segundo Crestani (2000), a aproximação da probabilidade com recuperação de informação, foi apresentado primeiramente por Maron e Kuhns. Desde então foram criadas diferentes técnicas, testadas e aplicadas, especialmente por Maron e Cooper, Rijsbergen, Croft e Turtle e por Furh e Robertson.

As literaturas na aproximação probabilística, até mesmo pelos autores mencionados, têm se mostrado denso, muito técnico e de difícil visualização. Os novos modelos probabilísticos desenvolvidos têm tido um firme fundamento e grande utilidade.

Novos modelos e ferramentas estão sendo desenvolvidos, no intuito de minimizar o tempo e aumentar a eficácia do processo de consulta. Com o advento da Rede Mundial de Computadores – (Internet), muitos esforços estão sendo aplicados com o intuito de se criar modelos que se adaptem a esta nova realidade.

Hoje em dia muitas informações estão disponíveis, mas existem alguns aspectos que devem ser melhorados, tais como: segurança, velocidade, autenticidade de informações e principalmente a recuperação de informações, que é o foco deste trabalho.

1.2 O Problema da Pesquisa

Um dos grandes problemas em recuperar informações, é o retorno muito grande de documentos, onde o usuário não tem tempo hábil para filtragem e seleção

dessas informações. Um outro fator importante é o estudo da real necessidade do usuário, pois muitas frases podem conter várias interpretações.

Objetivo geral: Estudar conceitos e modelos de Recuperação de Informação (IR), assim como estudar e analisar alguns sistemas que adotam modelos probabilísticos em IR, visando compreender os avanços e deficiências no campo da IR.

Objetivos específicos:

- Estudar tipos, conceitos e objetivos da IR;
- Analisar a diferença entre dados e informações;
- Estudar modelos de recuperação de informações, fornecendo uma abordagem especial sobre os modelos probabilísticos;
- Estudar as características de alguns sistemas de recuperação de informações;
- Estudar a aplicabilidade de modelos probabilísticos, através de sistemas de recuperação que adotam tais modelos;
- Efetuar uma análise dos modelos e sistemas de IR em bases textuais.

1.3 Justificativa e Relevância

Preservar informações para consultas futuras é um procedimento imprescindível, tanto no passado como para o presente. Cada dia que passa aumenta-se a quantidade de documentos e a necessidade de informações. Um dos fatores agravantes deste processo, é a qualidade e a velocidade da procura destas informações.

Com o advento da Internet, o processo da busca de informação se intensificou, e conseqüentemente a velocidade passou a ser um ponto muito importante. Faz-se necessário o estudo do mecanismo de funcionamento de sistemas de buscas de

informações, para compreender o que está sendo desenvolvido, no intuito de manter a qualidade e a velocidade.

Cada sistema possui características próprias trabalhando com determinado mecanismo de recuperação de informações. Torna-se importante o estudo destes sistemas, para a compreensão de diversas formas de busca de informações, e como cada sistema trata a qualidade e o tempo de busca destas informações. Como cada sistema adota um modelo de recuperação diferente, optou-se neste trabalho, sistemas que trabalham com modelos probabilísticos de recuperação de informação em bases textuais.

1.4 Métodos de Desenvolvimento da Pesquisa

Após a revisão da literatura relativa aos conceitos, paradigmas e problemas sobre IR, foram efetuadas pesquisas em bibliotecas digitais, visando modelos que tratam sobre IR.

Posteriormente foram efetuadas pesquisas em bibliotecas digitais, visando modelos e abordagens probabilísticas em IR.

Para a confirmação da relevância de alguns modelos, foram efetuados contatos com empresas e autores de publicações científicas. Tais contatos, foram importantes para detectar alguns dos principais sistemas que adotam modelos probabilísticos na procura de informações.

Logo após foram efetuadas diversas pesquisas, visando bibliografias e empresas que desenvolvem sistemas probabilísticos em IR, a fim de descrever o seu funcionamento, características, resultados e o tipo do modelo probabilístico adotado.

Por fim, após a descrição dos modelos e sistemas, parte-se para a análise e posteriormente para conclusões e sugestões.

1.5 Limitações do Trabalho

O trabalho proposto estuda modelos de recuperação de informações com ênfase nos modelos probabilísticos. Este trabalho também está limitado ao estudo dos sistemas INQUERY, THISL e OKAPI de recuperação de informação.

1.6 Estrutura do Trabalho

O presente trabalho está organizado em seis capítulos, distribuídos da seguinte forma:

O capítulo 1 apresenta os objetivos da pesquisa, justificativa e relevância do trabalho, metodologia de desenvolvimento, limitações e estrutura do trabalho.

O capítulo 2 apresenta a busca do referencial teórico sobre recuperação de informações, visando estudar paradigmas, objetivos, definições, tipos de recuperação e modelos clássicos em recuperação de informações. Efetuou-se uma abordagem sobre cada um dos modelos clássicos citados.

No capítulo 3, todos os esforços foram direcionados para definições e busca de modelos probabilísticos em recuperação de informação. Efetuou-se uma abordagem mais detalhada sobre cada um dos modelos probabilísticos relatados.

Já no capítulo 4, os esforços foram para detectar e descrever sistemas de IR, que adotam modelos probabilísticos em seu funcionamento.

No capítulo 5, tratou-se da análise dos modelos e sistemas propostos, e no capítulo seguinte, conclusão e considerações em que se demonstra o alcance dos objetivos.

Ao final são colocadas as fontes de referência, com as obras citadas e consultadas.

2 RECUPERAÇÃO DE INFORMAÇÕES

2.1 Introdução

Recuperação de Informação se divide em armazenamento, organização e acesso a itens de informação. A representação e organização das informações, deveriam ser aos interessados, uma tarefa fácil. A questão da necessidade de informação do usuário, não é um problema simples.

Não existe uma relação clara entre a necessidade de informação do usuário e a interface de procura de informações. O usuário primeiramente precisa traduzir sua necessidade, para depois solicitar a consulta através de uma ferramenta. Esta tradução produz um conjunto de palavras-chaves ou lista de termos, a qual faz um resumo da necessidade da informação do usuário. Algumas informações podem ser relevantes ou irrelevantes ao usuário.

2.2 História

Segundo Baeza Yates e Ribeiro Neto (1999), o homem vem organizando suas informações para uma posterior recuperação. Um exemplo clássico dessa organização é o índice de um livro.

Com o aumento de informações, técnicas especializadas foram necessárias para recuperar tais informações, de uma forma segura e rápida. Uma técnica muita conhecida, é o de associar algumas palavras selecionadas à sua informação correspondente.

Os índices sempre foram um ponto fundamental em sistemas de informação. Durante muito tempo foram criados índices manualmente. Muitas empresas ainda adotam índices manuais para recuperar cadastro de clientes ou pacientes.

> *As bibliotecas estavam entre as principais instituições para adotar sistemas de Recuperação de Informação. Na primeira geração, os sistemas adotavam catálogos de cartão, onde permitia procurar obras por nome de autor e título; na segunda geração, permitiam procurar obras por títulos de assunto, através de palavras-chave(keywords) e instalações de questões mais complexas; na terceira geração, interfaces melhoradas, formas eletrônicas e hypertexto foram desenvolvidas para a busca. (BAEZA YATES; RIBEIRO NETO, 1999, p.07).*

Os sistemas de busca de informações em redes informatizadas, continuam adotando índices parecidos como usavam os bibliotecários tempos atrás. Mas houve algumas mudanças fundamentais em virtude da tecnologia e a explosão da Rede Mundial de Computadores - Internet. Primeiro, o custo de acesso a várias fontes de informação ficou mais acessível, o que possibilitou um alcance maior de audiência. Segundo, os avanços na área da comunicação digital proveram maior acesso a informações. Mesmo que um usuário esteja longe, e que necessita de informação, pode ter acesso a ela.

Segundo Baeza Yates e Ribeiro Neto (1999), pela primeira vez na história, muita pessoas têm acesso livre a uma quantidade satisfatória de publicações e informações.

Segundo Weide (2001), por volta dos anos oitenta, foram propostas uma nova técnica em sistemas de recuperação de informações, Processo de Linguagem

Natural (NLP). Os sistemas que adotam esta técnica, compartilham textos dentro do documento com uma seqüência significante de palavras, ao invés das seqüências de caracteres.

Por volta dos anos noventa, quando os recursos da multimídia não pareciam mais algo distante, e com o aceitável caminho cognitivo de representação de informação, os problemas em recuperar informações começam a aumentar. Os usuários gradativamente não procuravam mais textos em documentos, mas também informações contendo sons, imagens ou vídeo. Felizmente, neste período as formas digitais de armazenamento foram aperfeiçoadas.

Com o avanço, os documentos não estavam mais apenas em estantes, mas também em uma forma acessível em computadores. Agora, podendo ter acesso ao conteúdo da informação diretamente em um documento, a tarefa de um sistema de IR aumentou intensamente.

2.3 Paradigmas em Recuperação de Informação (IR)

Segundo Weide (2001), o termo arquivo é uma lembrança do uso estático de uma coleção de documento em um recipiente de informações. Documentos são objetos passivos, esperando ser recuperados pelos seus pesquisadores. O nome desta atividade é Recuperação de Informação. Os papéis podem ser invertidos, tornando os documentos ativos, tentando achar pessoas para os quais eles são atraentes. Isto é especialmente pertinente à Internet, onde informações são oferecidas, e não se pode acompanhar os vestígios de todos os novos desenvolvimentos. Os filtros são introduzidos como um agente ativo de informações. Os documentos são corretamente organizados e orientados, podendo também se tornar elementos ativos.

O gerenciador de arquivos e documentos pode ser chamado de Sistema de Recuperação de Informação (IRS). Em arquivos clássicos, o sistema é completamente dirigido por processadores humanos. O personagem mais importante é o bibliotecário, que é o intermediador entre o cliente, o sistema e os arquivos. Em modernos sistemas, processadores humanos são substituídos por máquinas. A intenção desta substituição é a construção de IRS mais eficientes. Entretanto, ao substituir pessoas por máquinas, inicia-se o problema da construção de interfaces que imitem os seres humanos do modo mais perfeito possível.

Segundo Weide (2001), quando os pesquisadores consultam por um arquivo, pode-se distinguir alguns paradigmas:

- Os pesquisadores possuem apenas algumas informações não especificas sobre a sua necessidade de informação, e tenta transmitir esta necessidade ao IRS. Pode-se chamar esta faze de *formulação*;
- O IRS possui o conhecimento global do conteúdo dos documentos nos arquivos, e tenta investigar de perto este conteúdo a pedido dos pesquisadores. Pode-se chamar esta fase de *combinação*. O conteúdo do documento é descrito em uma caracterização. A formação desta caracterização é chamada de *indexação*. Desta *caracterização* o IRS pode estimar a relevância do documento. Pode-se notar que a estimação desta relevância pode diferir da idéia de relevância do pesquisador;
- Os documentos que são julgados relevantes devem ser apresentados ao pesquisador. Esta fase pode ser chamado de apresentação. Em uma biblioteca convencional, o bibliotecário pode oferecer documentos relevantes efetuando uma seleção. Aproximações mais sofisticadas são

possíveis, porém um IRS automatizado pode oferecer apenas o sumário em ordem para que posteriormente o usuário selecione os documentos relevantes.

Os paradigmas citados acima nos acompanham desde os anos cinqüenta, quando o campo da IR estava emergindo no reino das Ciências Bibliotecárias. Com o aparecimento de padrões como TEX, SGML e HTML, as estruturas dos objetos textuais puderam ser nomeadas. Porém, conhecer somente a base da estrutura não é o suficiente para descobrir a informação contida no objeto de informação. Por isso modernos IRS, usando Inteligência Artificial e Teoria da Probabilidade, estão sendo desenvolvidos para uma melhor desempenho da recuperação.

2.3 Recuperação de Informações e Seus Objetivos

Ultimamente tem crescido muito a necessidade de se recuperar informações, e o primeiro objetivo é a classificação dos documentos úteis no conjunto de informações.

Hoje em dia, segundo Baeza Yates e Ribeiro Neto (1999), recuperação de Informação incorpora os conceitos de modelagem, classificação de documentos, filtragem, interfaces, linguagem, etc.

A visão tendenciosa da rápida disseminação, entre os usuários dos modernos computadores pessoais, de ferramentas de IR para multimídia e aplicações de hypertextos, tem prevalecido por muitos anos.

A Internet está se tornando um depósito universal de conhecimento humano e cultura, no qual tem dado uma grande colaboração de idéias e informações numa escala nunca vista antes.

2.4 Problemas Abordados

Algumas questões são importantes de serem abordadas, por exemplo: apesar da tecnologia e dos avanços na área da IR, pessoas acham uma tarefa árdua recuperar informações que satisfaçam suas necessidades. Técnicas mais precisas são necessárias para que a qualidade de recuperação possa melhorar. Com o aumento da necessidade de informações, fica difícil saber quais as técnicas de recuperação que tratam os índices mais rapidamente e tempo menor de resposta. Um ponto muito importante é saber como estudar o comportamento dos usuários, a fim de saber o seu real interesse em uma busca de informações.

Segundo Baeza Yates e Ribeiro Neto (1999), o comércio eletrônico é uma tendência principal e mundial. Em uma transação eletrônica, muitas vezes, o comprador necessita enviar ao vendedor o número do seu cartão de crédito. Tais informações poderiam ser interceptadas, ocasionando transtornos para o comprador. Técnicas de criptografia para a codificação e autenticação automática, são recursos usados por várias instituições, para a segurança do comprador.

Existem muitas outras preocupações que precisam ser analisadas, tais como a dos direitos autorais e patentes impossibilitando o crescimento das grandes bibliotecas digitais, e a dos direitos a privacidade, por um lado pode denegrir muitas vezes uma imagem, por outro impossibilita o crescimento de informações na Internet.

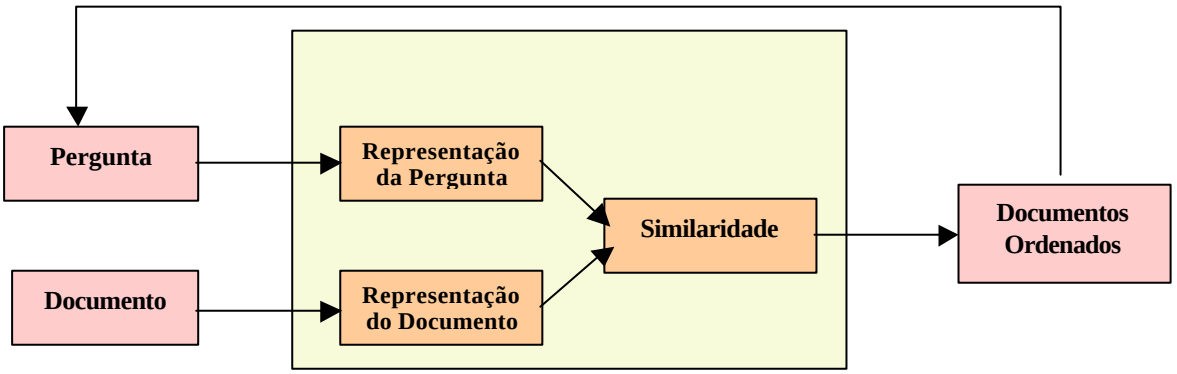
2.5 Definição

Hoje conta-se com um grande número de informações, segundo CRESTANI (1991) elas podem ser textuais, visuais ou auditivas. Como conseqüência os bancos que armazenam tais informações estão se tornando cada vez maiores. A IR trata em

desenvolver recursos para recuperar informações, independente do tamanho do banco de dados. Normalmente um sistema de IR conduz o usuário aos documentos que irão melhor possibilitar a satisfação de suas necessidades.

Um usuário que necessite de uma informação, devido à grandeza do banco de informações, precisa do auxílio de uma ferramenta chamada IRS. Ao contrário de um sistema tradicional de base de dados, o IRS não fornece uma resposta exata, mas um ranking de documentos com informações relevantes, os bons IRS apresentam primeiramente os documentos com maior similaridade com a pergunta, caso estes documentos sejam irrelevantes, o usuário reformula a pergunta para um novo ranking de documentos, conforma a figura abaixo, descrita por (CRESTANI, 1998).

Figura 01: Esquema de um Processo de Recuperação de Informações



Fonte: CRESTANI, F. et al. “Is this document relevant? ... probably”: a survey of probabilistic models in information retrieval. **ACM Computing Surveys**, New York, v.30, no.4, p. 528-552, Dec. 1998.

O ranking é uma coleção ordenada de uma lista de documentos devolvidos que reflete a relevância destes documentos em relação à pergunta do utilizador. Documentos no topo da lista são considerados como mais relevantes. Existem vários problemas referentes a IR, o tamanho do ranking de documentos recuperados, a análise semântica do pedido, compreensão da linguagem natural, palavras com várias interpretações, uso de pronomes e elipses para evitar repetições e o uso de metáforas

expressando outros pensamentos. Os dois problemas principais em IR são, os sinônimos e os polisêmicos. Os sinônimos são conjunto de palavras que detém o mesmo significado; os polisêmicos são palavras ou expressões que possuem vários conceitos ou pensamentos.

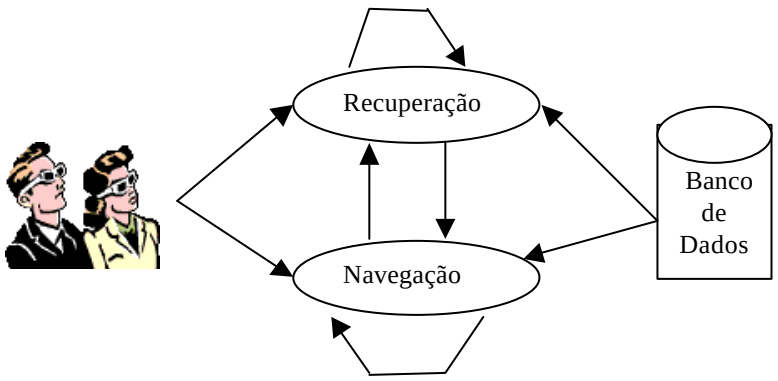
2.6 Conceitos Básicos

A recuperação basicamente requer tratamento tanto do lado do usuário como do lado do documento. Segundo Baeza Yates e Ribeiro Neto (1999), o usuário de um IRS, tem que traduzir sua necessidade de informação em uma linguagem que o sistema possa entender. O sistema relaciona um conjunto de palavras com a informação propriamente dita. O processo de procura do usuário por uma informação útil chama-se IR.

Existem diferenças entre recuperação e navegação. Por exemplo, um usuário poderia estar interessado em uma música, quando o usuário estiver procurando informações sobre músicas, poderia achar documentos que descrevem sobre o MP3. Lendo documentos sobre MP3 poderia achar informações sobre aparelhos eletrônicos que reproduzem músicas gravadas em MP3, e assim por diante. Neste caso, pode-se dizer que o usuário está simplesmente navegando ou folheando os documentos.

Em um processo de recuperação, os objetivos e os propósitos são mais objetivos e definidos, e tendem a mudar durante a interação com o sistema.

Figura 02: Interação do usuário com o sistema de recuperação por tarefas distintas

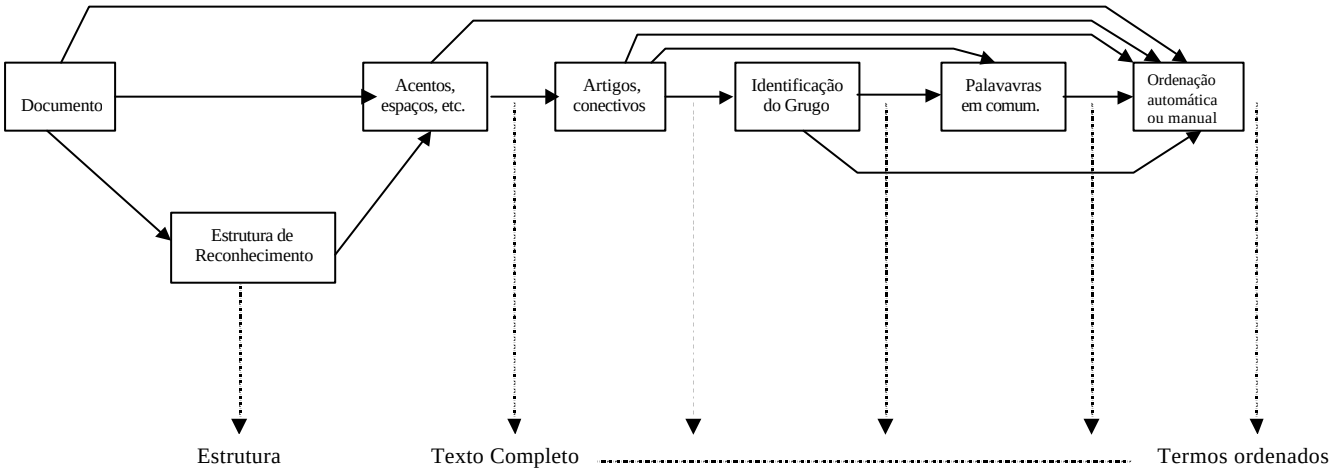


Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Segundo Baeza Yates e Ribeiro Neto (1999), a figura 02 nos mostra que navegação e recuperação ainda são coisas distintas. Bibliotecas digitais modernas e interfaces de rede, podem combinar estas duas tarefas, a fim de prover uma melhor capacidade de recuperação.

Historicamente os documentos são representados por um jogo de condições de índices ou palavras-chave. Estas palavras-chave poderiam ser ligadas diretamente ao texto ou ser especificadas por um assunto. Em geral pode-se dizer que estas palavras-chave apresentam uma visão lógica do documento.

Figura 03:Visão lógica de um documento de texto completo para um conjunto de termos ordenados



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

A passagem de um texto completo para um conjunto de palavras-chave se processa primeiramente pelo reconhecimento da estrutura dos documentos; identificar as palavras através da análise léxica; remoção das palavras com freqüência elevada (stopwords); detecção de termos equivalentes, frases-nome ou grupo de nomes e dos sinônimos; geração da estrutura do índice.Tendo como resultado um conjunto de

palavras que representam o texto, ou seja, um texto completo para uma representação mais específica.

2.7 Recuperação de Dados x Informações

Weide (2001), aborda a diferença entre recuperar dados e recuperar informações, podemos ver no quadro seguir.

QUADRO I : Diferença entre recuperação de dados e recuperação de informação

.Argumentos	Recuperação de Dados	Recuperação de Informações
A representação das informações armazenadas.	Tipos de objetos e fatos são bem definidos.	Informação desestruturada.
Método de responder a uma solicitação de informação.	Direto, através de fatos.	Informação que conterá os prováveis dados requeridos pelo usuário.
A relação entre a questão formulada para o sistema e a satisfação do usuário.	Satisfeito ou não satisfeito (determinístico).	Uma alta probabilidade de satisfação do usuário.
A definição para um sistema eficiente.	Pode o sistema requerer os dados perdidos?	Pode o sistema satisfazer a necessidade de informação do usuário?

Fonte: WEIDE, Th. P. van den. **Information Discovery,** 2001.

Segundo Baeza Yates e Ribeiro Neto (1999), recuperação de dados, num contexto de IR, consiste em determinar quais documentos contém as palavras-chave para a necessidade do usuário. Isto não satisfaz bastante a necessidade de informação do usuário. Um fato importante é que o usuário da IR está interessado em recuperar assuntos no conjunto de dados, que satisfaçam a sua necessidade. A linguagem da recuperação de dados, tem por objetivo recuperar todos os objetos que satisfaçam claramente as condições definidas pelas expressões regulares ou dentro de uma expressão da álgebra relacional. Assim, em um sistema de recuperação de dados, um

simples erro no objeto no meio de milhares de informações recuperadas, significa uma falha total.

Em um IRS, segundo Baeza Yates; Ribeiro Neto (1999), os objetos recuperados podem não ser precisos e pequenos erros podem ser ignorados. A principal diferença é que na IR usualmente é usada uma linguagem de texto natural, e que nem sempre são bem estruturadas e podem ser semanticamente ambíguos.

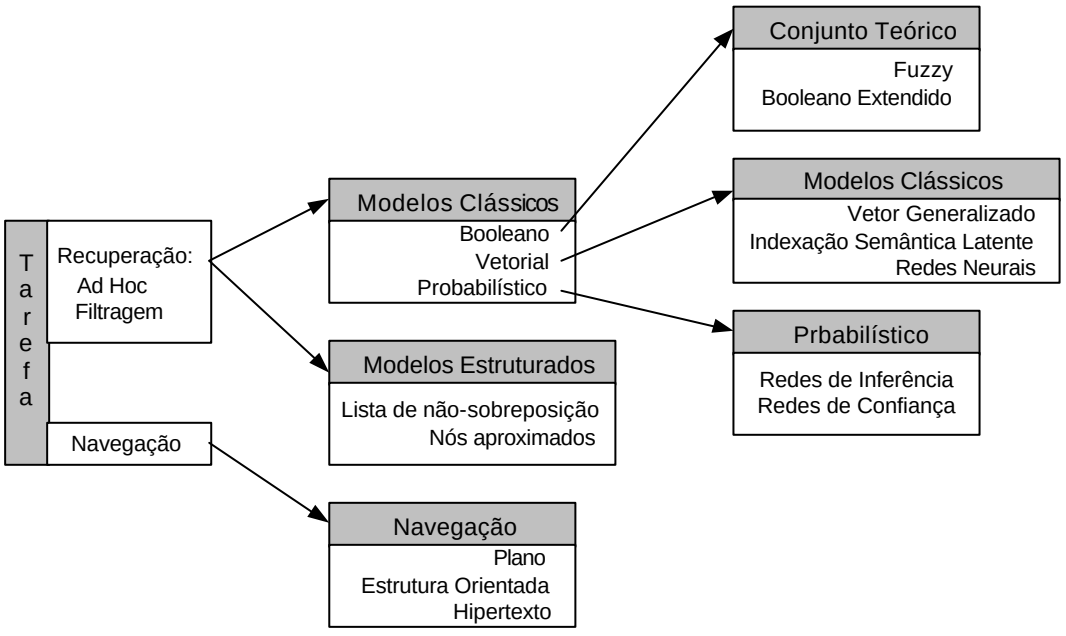
A recuperação de dados provê soluções para o usuário de sistemas de base de dados, mas não soluciona o problema de IR a respeito do assunto. A dificuldade não está apenas em saber a forma de extração destas informações, mas também em saber como o usuário decide a relevância das informações recuperadas. Assim, a noção de relevância está no centro da IR. O primeiro objetivo é recuperar o máximo de documentos relevantes possíveis e o mínimo de não relevantes.

2.8 A Organização dos Modelos em Recuperação de Informação

A característica principal do modelo booleano é que os documentos e *queries (necessidade de informação do usuário)* são representados por um conjunto de termos índices; no modelo vetorial, documentos e *queries* são representados por vetores em um espaço t-dimensional; no modelo probabilístico, o sistema de modelagem dos documentos e *queries* são representações baseadas na teoria probabilística. A diante estudar-se-á com maiores detalhes, os três modelos clássicos em IR: Booleano, Vetor Espacial e o Probabilístico. Ao longo do tempo, muitos paradigmas foram relatados em cada modelos, então foram propostas algumas alternativas que modelavam tais paradigmas, ou seja, muitos conjuntos teóricos algébricos e probabilísticos têm sido propostos.

Existem alguns modelos alternativos que vêm se destacando, logo em seguida estar-se-á mencionando alguns deles. Com respeito ao modelo teórico pode-se citar o modelo Fuzzy e o Booleano Estendido; ao modelo algébrico pode-se citar o Vetor Generalizado, Indexação Semântica Latente e Redes Neurais; ao modelo probabilístico pode-se citar as Redes de Inferência e Redes de Confiança. Existem modelos que além das referências do conteúdo do texto, o modelo faz referências na estrutura, normalmente em textos escritos. Neste caso, tem-se um modelo estruturado. Pode-se classificar os modelos estruturados da seguinte forma: modelo de não-sobreposição de listas e modelos de nós aproximados. Nos modelos de *Browsing* (navegação), destacam-se os modelos: Plano, Estrutura Orientada e Hipertextos. Estará sendo mostrado a seguir uma visão da classificação dos modelos em IR.

Figura 04: A organização dos modelos em recuperação de Informação.



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Figura 05:Modelos mais freqüentes de IR associados com a tarefa do usuário e a visão lógica do documento.

Visão Lógica dos Documentos

Tarefa		Termos Índice	Texto Completo	Texto Completo + Estrutura
	Recuperação	Modelos Clássicos Conjunto Teórico Algébrico Probabilístico	Modelos Clássicos Conjunto Teórico Algébrico Probabilístico	Estruturado
	Navegação	Plano	Plano Hipertexto	Estrutura Orientada Hipertexto

Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

2.9 Recuperação Ad-hoc e Filtragem

Existem dois tipos de recuperação: *ad-hoc* e filtragem. Em um sistema convencional de IR, os documentos na coleção permanecem relativamente estáticos, enquanto que novas *queries* são enviadas ao sistema. Este processo de recuperação, recentemente foi nomeado de *ad-hoc*. Entretanto em um sistema, enquanto que as *queries* permanecem relativamente estáticas, novos documentos estão sendo introduzido no sistema. A este processo de recuperação pode-se chamar de filtragem. Tipicamente a filtragem indica ao usuário, os documentos que possam ser relativamente interessantes para ele.

Existe uma variação da filtragem chamada de *routing*, que é interessante para o usuário, pois o seu objetivo é recuperar um conjunto pequeno de documentos, e que os documentos mais relevantes são colocados no topo da lista. A *routing* apesar de ser interessante, não é muito popular nos sistemas, pois é muito complexo determinar o tamanho mínimo de documentos relevantes.

Estar-se-á sendo relatando mais a diante a recuperação *ad-hoc* e as necessidades de se desenvolver modelos e sistemas que possam compreender ao máximo,a necessidade de recuperação do usuário

2.10 Text Retrieval Conference (TREC)

A Conferência de Recuperação de Texto (TREC), co-patrocinado pelo Instituto Nacional de Padrões e Tecnologia (NIST) e a Agência de Defesa Avançada de Projetos de Pesquisa (DARPA), teve sou início em 1992 como parte do sistema TIPSTER Text Program. Seu propósito era apoiar pesquisa dentro da comunidade de recuperação de informação, dando infra-estrutura necessária para ampla avaliação de metodologia de recuperação de texto. Em particular, a TREC tem as seguintes metas:

- Encorajar pesquisas em recuperação de informação baseados em teste de grandes coleções;
- Aumentar a comunicação entre indústria, academia, e governo, criando um foro aberto para a troca de idéias de pesquisa;
- Acelerar a transferência de tecnologia de laboratórios de pesquisa em produtos comerciais, demonstrando melhorias significativas em metodologias de recuperação em problemas do mundo-real;
- Aumentar a disponibilidade de técnicas de avaliações apropriadas, para o uso das indústrias e academias, inclusive desenvolvimento de técnicas novas de avaliação mais aplicáveis a sistemas atuais.

TREC é supervisionado por um comitê que consiste de representantes do governo, indústrias e academia. Para cada TREC, a NIST, provê um conjunto de teste de documentos e perguntas. Os participantes dos sistemas de recuperação da TREC,

devolvem para a NIST, uma lista de documentos recuperados e ordenados. A NIST agrupa os resultados individualmente para a avaliação dos resultados. O ciclo de TREC termina com um seminário anual, para que os participantes possam compartilhar suas experiências.

O número de participantes tem crescido muito ultimamente. Sessenta e seis grupos que representavam dezesseis países, participaram em TREC-8 (novembro de 1999). A TREC testa coleções de softwares de avaliações que estão disponíveis à comunidade de pesquisa de recuperação, assim organizações podem avaliar seus próprios sistemas de recuperação a qualquer hora. Dobrou-se o número de sistemas de recuperação desde a TREC-1.

A TREC testa coleções realisticamente vários sistemas de recuperação, e a maioria das máquinas comerciais de hoje, inclui tecnologia desenvolvida primeiramente pela TREC. Pode-se considerar hoje, a TREC como um padrão de qualidade de software na área de recuperação de informação.

2.11 Modelos Clássicos

Segundo Baeza Yates e Ribeiro Neto (1999), a caracterização de um modelo de Recuperação de Informação é um quarteto D, Q, F, $R(q_i,d_j)$, onde /D/ é um conjunto composto de representações para os documentos da coleção; /Q/ é um conjunto de representações para o usuário que necessita de uma informação; /F/ é um sistema de modelagem das representações dos documentos, perguntas e seus relacionamentos; $R(q_i,d_j)$ é uma função que ordena a associação entre a necessidade de informação (q_i) pertencente a Q, e a representação do documento (d_j) pertencente a D. Esta função

define uma ordem de importância entre os documentos em relação à pergunta do usuário.

Entre as diversas necessidades da IR, pode-se dizer que uma delas é encontrar modelos que tratem incertezas e ferramentas que possam tratar com mais eficiência o conhecimento impreciso. É uma tarefa muito difícil construir modelos inteligentes, capazes de assimilar a real interpretação de uma solicitação.

Existem hoje em dia muitas pesquisas experimentais em IR. Uma delas é encontrar meios para avaliação do desempenho de um sistema de IR, outra linha tem colocado seus esforços em estudar todas as pessoas envolvidas num sistema de IR.

Sabe-se que um dos maiores bancos de informações é a Internet. Em virtude disso muitos estudos são direcionados em criar modelos e ferramentas que aumentem a eficiência da busca de informações na maior rede de computadores.

Segundo Baeza Yates e Ribeiro Neto (1999), existem vários modelos que caracterizam a Recuperação de Informações, e que serão citados adiante. Em relação aos modelos não probabilísticos, será tratado apenas os Modelos Clássicos de IR.

2.11.1 Modelo booleano

Sabe-se que o modelo booleano, é baseado na álgebra booleana. Pela sua simplicidade e praticidade, este modelo de fácil assimilação, tem sido aplicado em vários sistemas, e em particular sistemas bibliográficos.

Mesmo sendo de simples compreensão e de semântica precisa, este modelo possui limitações, não conseguindo expressar uma necessidade através de expressões. Entretanto, tanto no passado como no presente este modelo recebe o seu destaque, pois muitos sistemas necessitam trabalhar com valores precisos sem pesos intermediários.

Este modelo verifica se os termos ordenados estão ou não no documento, não tendo alternativas intermediárias. Ele prediz se este documento é ou não relevante, ou seja, o retorno é um valor binário {0,1}. Seja a similaridade do documento (dj) em relação à *query* (q), temos:

$$sim(dj,q) = \begin{cases} 1 \rightarrow \text{Se } sim(dj,q) = 1, \text{ o documento (dj) é relevante para a } query \text{ (q);} \\ 0 \rightarrow \text{Se } sim(dj,q) = 0, \text{ o documento (dj) é irrelevant e para a } query \text{ (q);} \end{cases}$$

Um grande problema deste modelo, é a possibilidade de recuperar poucos documentos ou muitos documentos. Entretanto alguns sistemas de IR necessitam trabalhar na lógica Booleana, não necessitando de pesos intermediários.

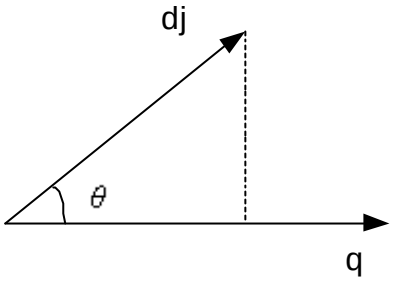
2.11.2 Modelo vetorial

Em IR muitas vezes a prática de um modelo Booleano, que usa pesos binários, não é muito eficiente. Existem vários sistemas IR que trabalham com valores aproximados, necessitando de pesos intermediários em sua estrutura. Muitos modelos de IR adotam o *rank* de documentos. O modelo vetorial é bem aceito devido a sua simplicidade.

O Modelo Vetorial, segundo Baeza Yates e Ribeiro Neto (1999), reconhece que o uso de pesos binários é muito limitado, e propõe uma condição intermediária. Isto é realizado, nomeando pesos não-binários para indexar condições em questões e em documentos. Estes pesos analisam o grau de similaridade entre um documento armazenado em um sistema e a *query* do usuário. Após esta análise, os documentos são ordenados em ordem crescente em grau de similaridade. Um resultado importante, é que o resultado obtido é um conjunto bem mais preciso do que o conjunto obtido pelo modelo Booleano.

Neste modelo, o documento (dj) e a *query* (q) do usuário são representados como um vetor t-dimentional.

Figura 06 :O cosseno do ângulo θ é substituído pelo seno(dj,q)



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Este modelo propõe uma avaliação no grau de similaridade do documento em relação a sua *query*, e nas correlações entre os vetores $\vec{dj}$ e $\vec{q}$, como mostra a figura 06. Esta correlação pode ser quantificada pelo cosseno do ângulo entre os dois vetores, onde o cosseno do ângulo é equivalente à fórmula abaixo:

$$\text{seno}(dj,q) = \frac{\vec{dj} \bullet \vec{q}}{|\vec{dj}| \text{ x } |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \text{ x } w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \text{ x} \sqrt{\sum_{j=1}^{t} w_{i,q}^2}} \qquad (01)$$

onde (dj) são os documentos, (q) são as *queries*, ($w_{i,j}$) é o peso de cada documento e ($w_{i,q}$) é o peso de cada *query*.

As principais vantagens deste modelo são: o esquema de pesagem de temo melhora o desempenho da recuperação; sua estratégia permite recuperação de documentos que aproximam às condições das *queries*, e o grau do cosseno permite ordenar os documentos de acordo com a semelhança com a questão.

2.11.3 Modelo probabilístico

O modelo Probabilístico em IR data por volta dos anos 60, porém definitivamente introduzido por em 1976 por Robertson e Spark Jones, conhecido mais

tarde como Modelo de Independência Binária (BIR). O modelo probabilístico foi talvez o primeiro modelo IR com uma firme fundamentação teórica.

Segundo Baeza Yates e Ribeiro Neto (1999), o modelo probabilístico tenta recuperar informações usando a teoria da probabilidade. Dado uma *query* do usuário, há um conjunto de documentos que contém exatamente os documentos relevantes, e nenhum outro. Dado uma *query*, o retorno seria um conjunto de resposta ideal. Um grande problema, é saber as propriedades significativas dessa resposta ideal. Esta suposição permite gerar uma descrição probabilística preliminar do conjunto de resposta ideal, que é usada para recobrar o primeiro conjunto de documentos.

O princípio da hipótese probabilística, segundo Baeza Yates e Ribeiro Neto (1999), revelam que: dado ao usuário uma *query* (q) e um documento (dj) na coleção, o modelo probabilístico estima a probabilidade do usuário achar o documento (dj) relevante. O modelo determina que esta probabilidade de relevância depende somente da *query* e da representação do documento. Além disso, o modelo determina que existe um sub-conjunto de todos os documentos que o usuário pretende como resultado de sua busca para a *query* (q). O resultado ideal, que é denominado por (R), deve maximizar toda probabilidade relevante para o usuário. Documentos no conjunto (R) são previstos ser relevante para a *query*. Documentos fora deste conjunto são previstos ser não relevantes.

Esta hipótese é bastante preocupante porque não declara explicitamente como computar a relevância das probabilidades. Na verdade, nem mesmo o espaço da amostra que é usado para definir tal probabilidade é dado. Dado uma *query* (q), o modelo probabilístico determina que para cada documento (dj), como uma medida de

similaridade para a *query*, a relação $\frac{P(\text{dj relevante para a (q)})}{P(\text{dj não - relevante para (q)})}$ que calcula a probabilidade do documento (dj) ser relevante para a *query* (q).

O objetivo deste modelo é estimar $P_q(d_k/R)$, a probabilidade do documento d_k, dentro do conjunto de resposta ideal (R), ser relevante para a questão q. Para computar $P_q(d_k/R)$, os IRS's deve de alguma forma, representar e armazenar os documentos. Os IRS's freqüentemente representam o documento por um conjunto de palavras conhecidas, os termos índices. No geral, os termos índices são aquelas palavras do documento que ficam no topo da listas (lista das palavras comuns). Estas palavras são obtidas freqüentemente, removendo-se os prefixos e sufixos.

Inicia então, uma interação com o usuário, com o propósito de melhorar a descrição probabilística do conjunto de resposta ideal. O usuário analisa os documentos recobrados e decide os relevantes ou não. O sistema usa esta informação para refinar a descrição do conjunto de resposta ideal. Repetindo-se muitas vezes este processo, existiria então uma evolução e conseqüentemente uma aproximação do conjunto de resposta ideal. Assim, o usuário deverá ter em mente no princípio, a necessidade, para prever o conjunto de resposta ideal.

2.11.4 Modelo fuzzy

É muitas vezes introduzido a lógica clássica no dia-a-dia de uma forma unidimensional. Com a introdução da lógica Fuzzy, constatado sua eficiência em várias situações, seu conceito tem sido aplicado em vários sistemas, especialmente em sistemas de IR.

Segundo Baeza Yates e Ribeiro Neto (1999), documentos e *queries* são representados pelas descrições de palavras-chaves, que são parcialmente relacionados

aos reais conteúdos semânticos dos respectivos documentos e *queries*. Como resultante, a união dos documentos com os termos da *query* é aproximado ou vago, ou seja, parcial. Isto pode ser aceito, considerando que cada termo da *query* define um conjunto Fuzzy e cada documento possui um grau de similaridade (usualmente menor que 1) neste conjunto. Esta interpretação de processo de recuperação (em termos de conceitos da teoria Fuzzy) é a fundamentação básica dos vários modelos de conjuntos Fuzzy para IR, o qual tem sido proposto durante os anos.

A teoria do conjunto Fuzzy lida com a representação de classes cujos limites não são bem definidos. A idéia principal é associar a função de relacionamento com os elementos das classes. Esta função recebe valores entre o intervalo (0,1) com 0 correspondendo a nenhuma equivalência na classe, e 1 correspondendo para uma completa equivalência junto a classe. Os valores da equivalência entre 0 e 1 indicam os elementos limites da classe.

O sub-conjunto Fuzzy *A*, dentro do universo *U* é caracterizado pela função de associação $\mu_A : U \rightarrow [0,1]$, que leva cada elemento *u* de U a um número $\mu_A(u)$ no intervalo [0,1]. Assim, a equivalência em um conjunto Fuzzy é uma noção intrinsecamente gradativa ao invés de uma noção repentina, como na convencional lógica Booleana.

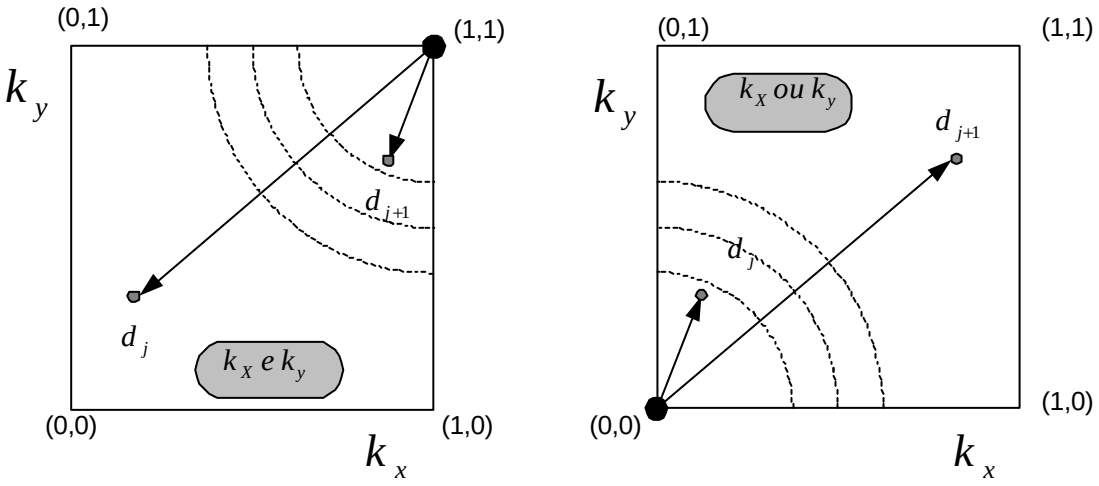
2.11.5 Modelo booleano estendido

O Modelo Booleano nos retorna uma resposta binária (0,1). Isto tem se tornado, muitas vezes, uma limitação para vários sistemas. O Modelo Vetorial trabalha com valores intermediários, ou seja, com aproximações. Uma estratégia foi combinar a facilidade do modelo Booleano com a eficiência do modelo Vetorial.

A recuperação booelana é simples e elegante. Entretanto, desde então não há nenhuma regulamentação para a pesagem dos termos, e não é apresentado nenhum relatório do conjunto de resposta. Como resultante, o tamanho da saída do conjunto de resposta pode ser muito grande ou muito pequeno. Por causa destes problemas, modernos IRS não estão longe de estarem baseados no modelo Booleano. Na realidade, muitos dos novos sistemas têm adotado o modelo Vetor Espacial como elemento central. As razões disto, é devido à facilidade e simplicidade deste modelo na melhora do desempenho da IR. Uma alternativa de aproximação é estender o modelo Booleano com a funcionalidade da combinação parcial da pesagem de termo. Esta estratégia permite combinar as formulações *query* Booleanas com características do modelo vetorial.

O modelo Booleano Estendido, introduzido em 1983, é baseado em uma crítica de uma suposição básica da lógica Booleana. Considerando uma *query* conjuctiva Booleana dada por $q = k_x \wedge k_y$, de acordo com o modelo Booleano, um documento que contenha o termo k_x ou termo k_y , é tão irrelevante quanto um outro documento que não contém nenhum deles.

Figura 07-- :Lógica Booleana Estendida considerando um espaço composto de apenas dois termos.



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Pode-se notar que o documento *dj* é posicionado em um espaço através de pesos $w_{x,j}$ e $w_{y,j}$ com os pares $\lfloor k_x, d_j \rfloor$ e $\lfloor k_y, d_j \rfloor$ respectivamente. Os pesos são normalizados nos valores entre 0 e 1. De acordo com a necessidade de valores aproximados em IR, muitos modelos precisarem ser desenvolvidos. Os resultados obtidos através do modelo Booleano Estendido são altamente positivos, pois permitem trabalhar a eficiência lógica Booleana com pesos intermediários.

2.11.6 Modelo espacial vetorial generalizado

Foi proposta em 1985 uma interpretação na qual os vetores dos termos índices são assumidos linearmente independente mas não são ortogonalmente paralelos. Vale ressaltar que o termo índice é uma palavra chave ou um grupo selecionado de palavras, e existem várias técnicas para a seleção destas palavras. A importância do termo índice é representada por um peso que lhe é associado. Seja (*ki*) o termo índice, (*dj*) o documento e (*Wij*) o peso associado ao par (*ki,dj*). O peso(*Wij*) quantifica a importância do termo índice para descrever o conteúdo do documento.

Tal interpretação nos leva a um modelo Espacial Vetorial Generalizado que passa-se a discutir. No modelo Vetor Espacial Generalizado, dois vetores de termos índices podem ser não ortogonais. Isto significa que os vetores de termos índices não são visto como vetores ortogonais que compõem a base do espaço.

Segundo Baeza Yates e Ribeiro Neto (1999), freqüentemente a independência entre os termos índices é interpretada de uma forma mais restritiva para significar ortogonalidade paralela entre os vetores dos termos índices, isto é, significa que para cada par de vetores de termos índices $\vec{k}_i$ e $\vec{k}_j$ tem-se $\vec{k}_i \bullet \vec{k}_j$.

Os termos índices são compostos de pequenos componentes derivados de pequenas coleções como pode ser vista na definição a seguir.

> *Dado o cojunto {k1, k2, ... , kt} os termos índices em uma coleção, tal como antes permite wij ser associado com o par termo-documento (ki, dj). Se os pesos de wij forem todos binários então todos os possíveis padrões de termo da co-ocorrência podem ser representados por um cojunto de 2t sub-termos, dados por m1=(0,0,...,0), m2=(1,0,...,0), ..., m2t =(1,1,...1) permitindo que gi(mj) retorne ao peso {0,1} dos termos índices ki, nos sub-termos mj (BAEZA YATES; RIBEIRO NETO,1999 p.41).*

No modelo vetorial clássico, o documento *dj* e o usuário da *query q* são expressos através dos vetores $\vec{d_j} = \sum_i w_{i,j} \vec{k}_i$ e $\vec{q} = \sum_i w_{i,q} \vec{k}_i$, respectivamente. No modelo vetorial especial generalizado, estas representações podem ser diretamente traduzidas para os espaço dos vetores $\vec{m}_r$ através da aplicação da equação abaixo:

$$\vec{k}_i = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \vec{m}_r}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c_{i,r}^2}} \quad \text{onde} \quad c_{i,r} = \sum_{dj / gl(\vec{d_j}) = gl(m_r) \text{ para todo } l} w_{i,j} \qquad (02)$$

A equação acima nos fornece uma definição geral do termo índice do vetor $\vec{k}_i$ nos termos dos vetores $\vec{m}_r$. O termo do vetor $\vec{k}_i$ coleciona todos os $\vec{m}_r$ vetores onde o termo índice k_i está na posição 1. Para cada vetor $\vec{m}_r$, a correlação do fator $c_{i,r}$ é definida. Analogamente a correlação entre os fatores adicionados e os pesos $w_{i,j}$ associados com o termo índice k_i em cada documento d_j, de quem a ocorrência padrão

coincide exatamente com o sub-termos m_r. O produto interno $\vec{k}_i \bullet \vec{k}_j$ pode ser usado para quantificar o grau de correlação entre os termos índices k_i e k_j, sendo assim:

$$\vec{k}_i \bullet \vec{k}_j = \sum_{\forall | g_i(m_r)=1 \wedge g_j(m_r)=1} c_{i,r} \text{ x } c_{j,r} \qquad (03)$$

O ranking dos resultados do modelo vetor espacial vetor generalizado associa o peso padrão dos documentos $w_{i,j}$ com a correlação fatorial $c_{i,r}$. Futuramente este modelo poderá trabalhar com maior eficiência em grandes coleções de documentos, porque o número de sub-termos poderá ser proporcional ao número de documentos na coleção.

2.11.7 Modelo de indexação semântica latente

Segundo Baeza Yates e Ribeiro Neto (1999), resumindo os conteúdos dos documentos e *queries* através de um conjunto de termos índices, podem conduzir a uma pobre performance de IR devido a dois efeitos. Primeiro, muitos documentos não relacionados podem ser incluídos no conjunto de respostas. Segundo, documentos relevantes que não estão indexados por nenhuma das palavras chave da *query* não são recuperados. A principal razão destes dois efeitos é a inerente incerteza associada com o processo de recuperação o qual é baseada em conjunto de palavras-chave.

As idéias em um texto estão mais relacionadas aos conceitos descrito no texto que com os termos indexados usados em sua descrição. Assim, o processo de combinar documentos para uma determinada *query*, pode ser baseada em um conceito de combinação, ao invés vez de termos índices. Isto permitiria a recuperação de documentos até mesmo quando eles não estão indexados pelos termos índices da *query*.

Por exemplo, o documento pode ser recuperado porque compartilha conceitos com outros documentos a qual é relevante para a determinada *query*.

A Indexação Latente Semântica é uma aproximação introduzida em 1988. A idéia principal deste modelo é traçar cada documento e vetor *query* em um espaço dimensional menor que é associado com estes conceitos. Isto é realizado através do mapeamento dos vetores de termos índices no espaço menor. A afirmação é que recuperação dentro de um espaço reduzido pode ser superior a recuperação em um espaço de termos índices. Seja t o número de termos em uma coleção e N o numero total de documentos. Pode-se definir $\vec{M} = (M_{i,j})$ como um termo do documento associado com uma matriz de t linhas e N colunas. Para cada elemento $M_{i,j}$ desta matriz é designada um peso $w_{i,j}$ associado com par dos termos do documento $\lfloor k_i, d_j \rfloor$. Este peso pode ser generalizado usando-se técnicas padrões do modelo clássico de vetor espacial.

Pode-se notar que este modelo nos fornece um conceito interessante em problemas de recuperação de informação baseadas na teoria elementar do valor de decomposição.

2.11.8 Modelo de redes neurais

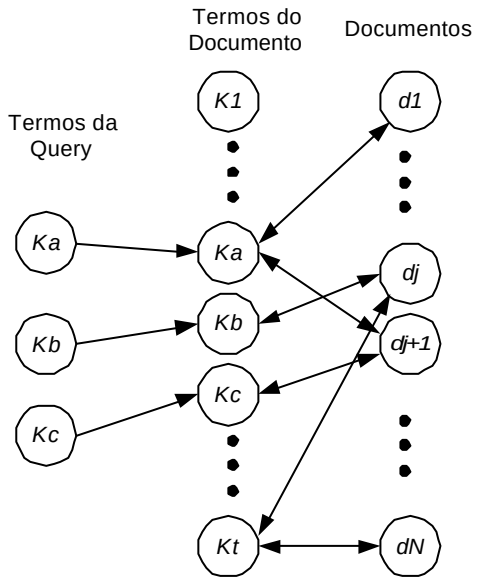
Segundo Baeza Yates e Ribeiro Neto (1999), em um IRS, vetores dos documentos são comparados com vetores da *queries* para o cálculo da posição. Assim, os termos índices nos documentos e *queries* têm que ser combinados e pesados para o cálculo desta posição. Desde então redes neurais são conhecidas como bons padrões de combinação, e é natural considerar o uso deles como um modelo alternativo para a IR.

Agora é bem estabelecido que o nosso célebro é composto por bilhões de neurônios. Cada neurônio pode ser visto como uma unidade de processamento básico o

qual, quando estimulados através de sinais de entrada, podem emitir sinais de saída como uma ação reativa. O sinal emitido através dos neurônios alimenta outros neurônios, o qual pedem novos sinais de saída. Este processo pode se repetir através de várias camadas de neurônios e é usualmente chamada de expansão ativa de processos. Como resultado, a entrada de informação é processada, o qual pode ser conduzida pelo celebro através do controle de resposta de reações físicas.

A rede neural é uma simples representação gráfica da rede de interconexões de neurônios no cérebro humano. Os nós deste gráfico são as unidades de processo enquanto que as extremidades desempenham a função de uma conexão sináptica (região de encontro de duas células nervosas). Simular o fato que a força de uma conexão sináptica em um cérebro humano muda a todo tempo, um peso é nomeado a cada extremidade em nossa rede de neurônios. A cada instante, os estados dos nós são definidos através dos níveis de ativação. Dependendo deste nível de ativação, o nó A pode enviar um sinal ao seu vizinho, nó B. A força deste sinal até o nó B, depende do peso associado às extremidades entre o nó A e B.

Figura 08: Modelo de redes neurais para IR.



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

A figura anterior relata três níveis distintos: uma para os termos da *query*, outra para os termos do documento e a terceira para os documentos. Existe uma similaridade da topologia deste modelo com a topologia dos modelos inferenciais e redes de confiança. Os nós dos termos da *query* iniciam o processo enviando sinais para os nós dos termos do documento. Em seguida, os nós dos termos do documento podem gerar sinais para os nós do documento. Nas redes neurais, um fato que pode-se notar, é que os próprios documentos podem gerar novos sinais que são direcionados aos nós dos termos do documento, agindo assim de uma forma bidirecional. Recebendo os estímulos dos nós do documento, os nós dos termos do documento podem então gerar novos sinais direcionados aos nós do documento.

Um dos grandes problemas encontrados neste modelo, é sua confiabilidade após sucessivas propagações. Não existe uma conclusão real sobre a eficiência deste modelo ser de grande performance em grandes e complexas coleções. Pode-se analisar que este modelo não foi testado o suficiente a ponto de termos um parâmetro significativo. Mas este modelo nos mostra como uma alternativa para alguns paradigmas em IR.

.

3 ABORDAGEM PROBABILÍSTICA

3.1 Definição

Há muito tempo vem se desenvolvendo a teoria probabilística em IR. Hoje existem vários Sistemas de IR baseados em modelos probabilísticos ou semi-probabilístico.

Segundo Crestani et al (1998), em IR, modelagem probabilística é o uso de modelos que ordena documentos em ordem descresente de importância, de acordo com a necessidade de informação do usuário.

Um principal obstáculo para os modelos de IR probabilísticos ou semi-probabilísticos, é encontrar métodos relevantes que sejam teoricamente e computacionalmente eficientes. Muitos métodos têm se desenvolvido para a melhoria da Recuperação de Informação. Uma delas é o uso de técnicas estatísticas e análise de regressão logística.

O grau de heurística merece uma atenção importante em IR. Uma solução seria o uso da regressão logística ao invés da regressão padrão. O modelo padrão de regressão considera a variável dependente como contínua, e o modelo de regressão logística é uma adaptação em situações em que a variável dependente é do tipo [0-1].

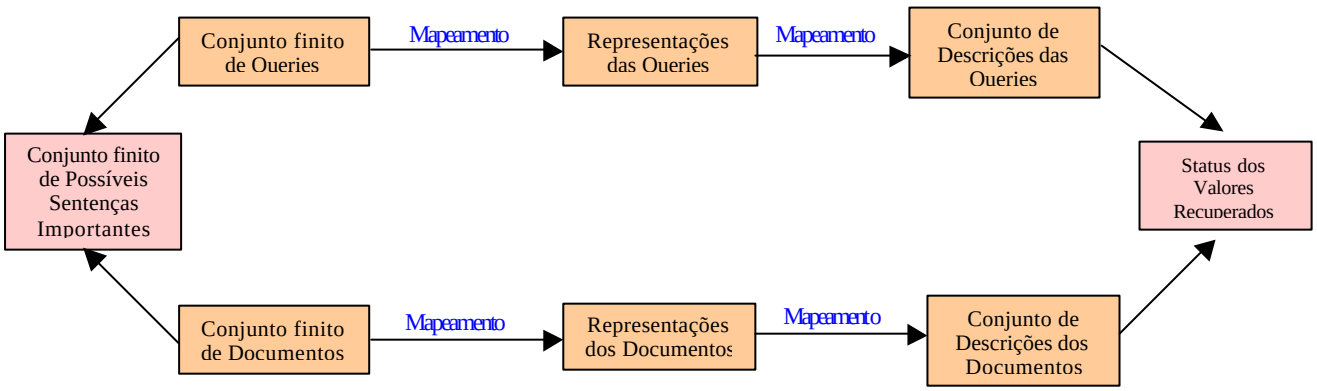
Existem muitas correntes de pesquisas no mundo, uma delas têm o objetivo do desenvolvimento de modelos baseados na lógica não-clássica e na teoria da probabilidade, ou seja a integração da lógica com a linguagem natural de processamento.

Geralmente modelos probabilísticos têm como espaço amostral o cojunto I x D, onde I representa o conjunto de *queries* (necessidade de informação) e D a união de

documentos. Uma *query* pode ser definida como uma expressão de uma necessidade de informação. Vale ressaltar que uma *query* é um acontecimento único ou seja, se uma *query* for feita por dois usuários diferentes, são consideradas diferentes. Geralmente em IR relaciona-se documentos como um objeto textual, mas na realidade pode ser texto, imagem ou som.

Analisa-se a seguir o modelo conceitual de Fuhr, que oferece uma base conceitual a todos os modelos probabilísticos.

Figura 09 - Modelo Conceitual de Furh em Recuperação Probabilística



Fonte: CRESTANI, F. et al. "Is this document relevant? ... probably": a survey of probabilistic models in information retrieval. **ACM Computing Surveys**, New York, v.30, no.4, p. 528-552, Dec. 1998.

O conjunto das Possíveis Sentenças Importantes (PSI), pode ser considerado um caso binário PSI=(R , R), ou seja, a ligação de um documento(R) com uma *query* (R).

O mapeamento das *queries* é importante, por exemplo, o sucesso da busca de um livro em uma biblioteca depende diretamente da qualidade das suas representações, quer estejam por autor, título ou por resumo. Em IR outro mapeamento torna-se necessário, o mapeamento entre as representações e descrições. Na seguinte *query*, "*Eu estou procurando apostilas do Excel*", ***apostila*** e ***Excel*** seriam provavelmente duas descrições importantes.

De acordo com o princípio de ordenamento de probabilidade, a IR assume a tarefa de ordenar os documentos de acordo com uma probabilidade estimada de ser importante P(R/qk, dj), isto é, probabilidade(P) da relevância(R) dado o relacionamento entre um documento (dj) e a necessidade de informação do usuário(qk). Vale lembrar que relevância é um relacionamento que pode ou não pode existir, entre um documento e um usuário do IRS, que procura alguma informação. Se o usuário se interessar pelo documento, pode-se dizer que houve uma conexão.

Segundo Crestani et al (1998), a necessidade de apresentar a relevância como uma probabilidade, deve se ao fato de que relevância é uma função de um grande número de variáveis concernente ao documento, ao usuário e às necessidades de informações, e portanto, pode ser considerado como uma variável aleatória. É impossível prever rigorosamente se a conexão de relevância se manterá entre um documento e uma necessidade de informação do usuário. A probabilidade entra com o papel de aproximar esta previsão.

Segundo o princípio da probabilidade de ordenamento, pode-se considerar uma recuperação ideal quando os documentos estão ordenados, de acordo com suas probabilidades, em ordem de relevância para uma *query*.

3.2 Modelos

Vale ressaltar duas categorias principais em IR Probabilístico: modelos probabilísticos e modelos inferenciais. Os modelos probabilísticos, são baseados na evidência sobre quais documentos são relevantes para uma *query*. Os modelos inferenciais aplicam conceitos e técnicas originadas de áreas tais como lógica e Inteligência Artificial.

Segundo Crestani et al *(1998) e* Baeza Yates e Ribeiro Neto (1999), existem muitos modelos probabilísticos que podem ser colocados em destaque, e que passa-se-á a descrever. Existem vários outros modelos sendo desenvolvidos em IR, entretanto pode-se concluir que os Modelos Probabilísticos são importantes e satisfatórios em processar informações incertas e imprecisas que é característica dos usuários da IR.

3.2.1 Modelo probabilístico generalizado (Okapi system)

Passar-se-á a mostrar o modelo formal usado pelo sistema Okapi de Recuperação de Informação, segundo (JONES et al., 1998)

Seja algum documento (D) e queries Q, logo ter-se-á dois eventos:

1. L, se D é considerado, ou seja, é relevante para Q;
2. $\bar{L}$, se D não é considerado, ou seja, não é relevante para Q.

Pode-se calcular a probabilidade de P(L/D), ou seja, a probabilidade de que um documento é considerado, qualquer que seja a descrição que ele tenha. Mas para permitir a pós-expansão de D aos atributos de D, aplica-se o Teorema de Bayes e expressa-se P(L/D) nos termos de P(D/L) ou seja, a probabilidade da identificação no conjunto de documentos, dado o fato do documento ser considerado. Esta inversão da probabilidade condicional, é graças ao Teorema de Bayes

$$P(L/D) = \frac{P(D/L)P(L)}{P(D)} \qquad (04)$$

Além disso, desde que o uso desta fórmula tenha requerido uma expansão prolongada da P(D) além do que se deseja, pode-se usar simplesmente usar os *log-odds* (derivados da probabilidade para uma transformação pré-requerida, isto satisfaz os princípios da probabilidade).

$$\log \frac{P(L/D)}{P(\bar{L}/D)} = \log \frac{P(D/L)P(L)}{P(D/\bar{L})P(\bar{L})} = \log \frac{P(D/L)}{P(D/\bar{L})} + \log \frac{P(L)}{P(\bar{L})} \quad (05)$$

Introduz-se agora a idéia do resultado da busca, MS (Matching Score), como uma função de descrições, especificamente MS(D) como o resultado para um documento individual. Assim, começa-se aqui com o caso mais primitivo, e defini-se

$$\text{MS-PRIM(D)} = \log \frac{P(D/L)}{P(D/\bar{L})} \quad (06)$$

MS-PRIM é a função de todo documento descrito em D; planeja-se ampliar isto depois em uma função de atributos de D

$$\text{MS-PRIM(D)} = \log \frac{P(D/L)}{P(D/\bar{L})} - \log \frac{P(L)}{P(\bar{L})} \quad (07)$$

Desde que o último termo é o mesmo para todos os documentos, a posição dos documentos na ordem MS-PRIM é o posicionamento na ordem P(L/D). Assim, dado uma estimativa para MS-PRIM para cada documento, pode-se usá-lo para posição do documento na ordem apropriada.

Este modelo adotado pelo sistema Okapi, tem dado bons resultados, pois segundo este princípio o sistema tem tido bom desempenho na área de recuperação de informações. Os resultados deste modelo adotados pelo sistema Okapi serão descritos adiante.

3.2.2 Modelo de recuperação de independência binária (BIR)

Nesta seção, segundo Crestani et al (1998), continua sendo necessária para avaliação, o conjunto de probabilidades P(d/R) e P(dj/$\bar{R}$), isto é, P($\vec{x}$/R) e P($\vec{x}/\bar{R}$), onde P($\vec{x}$/R) e P($\vec{x}/\bar{R}$) é a probabilidade da existência da a representação $\vec{x}$ no

documento de vetor binário (R, $\bar{R}$), podendo esteser relevantes ou não relevantes para a *query*.

Para simplificar o processo de estimação, os componentes do vetor $\vec{x}$ são estocasticamente independente quando condicionalmente dependente de R ou $\bar{R}$. Isto é, a junção da distribuição de probabilidade com os termos do documento dj é determinado pelo produto da distribuição de probabilidade:

$$P(dj/R) = P(\vec{x}/R) = \prod_{i=1}^{n} P(xi/R),$$
$$P(dj/\bar{R}) = P(\vec{x}/\bar{R}) = \prod_{i=1}^{n} P(xi/\bar{R}), \qquad (08)$$

A teoria da independência binária está baseada no primeiro modelo proposto por Robertson e Sparck Jones, o Modelo de Recuperação de Independência Binária. A suposição sempre foi reconhecida como irreal. Não obstante, foi apontada por Cooper, a suposição de estar por baixo do Modelo de BIR, não o torna uma independência binária, mas uma fraca suposição de união de dependência.

$$\frac{P(\vec{x}/R)}{P(\vec{x}/\bar{R})} = \prod_{i=1}^{n} \frac{P(xi/R)}{P(Xi/\bar{R})} \qquad (09)$$

Isto declara que a relação entre a probabilidade e $\vec{x}$ ocorre em documentos relevantes e irrelevantes, e é igual ao produto das relações correspondentes das únicas condições.

Considerando as estratégias já vistas, uma nova possibilidade de obter uma estratégia definitiva usando transformação logarítmica obtendo assim uma função de decisão linear:

$$g(dj) = \log \frac{P(dj / R)}{P(dj / \bar{R})} > \log \ddot{e}.. \qquad (10)$$

Fazendo substituições na fórmula acima chega-se na seguinte condição:

$$g(dj) = \sum_{j=1}^{n} cjxj + C, onde \ \text{cj} = \log \frac{\text{pj(1 - qj)}}{\text{qj(1 - pj)}} \ e \ \text{C} = \sum_{\text{j=1}}^{\text{n}} \log \frac{1 - pj}{1 - qj}. \qquad (11)$$

Esta fórmula nos concede o Valor do Estado da Recuperação (RSV), valor da recuperação do documento dj para a procura considerada. O valor de l pode ser usado para determinar o ponto de parada da lista dos documentos, embora o RSV é geralmente usado para medir a coleção inteira de documentos. Em um sistema IR de verdade, a ordenação dos documentos é probabilisticamente de acordo com a relevância dos documentos e em relação à solicitação.

Então, desde que o valor de C seja constante em uma procura específica, necessita-se de considerar apenas o valor de cj. Este valor é chamado de Peso de Relevância do Termo (TRW), e discrimina os documentos relevantes dos irrelevantes. Como pode-se ver, o peso dos termos relevantes no modelo BIR, pode contribuir independentemente para a relevância de um documento.

Para aplicação do modelo BIR, é necessário a estimativa do parâmetro *pj* e *qj* para cada termo usando na procura, e pode ser retrospectivo ou previsivo. Na primeira é usada uma coleção testes onde os índices de relevância são conhecidos; no segundo com uma coleção normal os parâmetros são estimados através do *"feedback"* do usuário.

Pode-se exemplificar o modelo segundo FUHR (1992), conforme a tabela a seguir, assumindo que a query (q) tenha apenas dois termos $q^T = (t_1, t_2)$, onde $T = \{t_1, ..., t_n\}$ é o conjunto de termos na coleção. Toma-se também que $d_i = 20$ é o

número de documentos junto com a distribuição dos termos dentro dos documentos, e que o vetor $\vec{x} = (x_1, x_2)$ recebe os seguintes valores binários em ordem decrescente de significância: (1,1), (1,0), (0,1), (0,0).

Tabela 01: Exemplo do modelo BIR de Recuperação.

di	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
x_2	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
$r(q,d_i)$	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*	*R*

Fonte: FUHR, N. Probabilistic models in information retrieval. **The Computer Journal**, v.35, no.3, p.243-255, 1992.

Weide (2001) nos mostra em seus experimentos, um exemplo da performance do BIR. Em seu exemplo, foi tomados um arquivo com 100 documentos, numerados de 1,...,100. Foram escolhidos os números: 0,...,b-1 (com b=94) termos de indexação. São escolhidos os 10 primeiros termos índices associados aos documentos.

Efetuados a distribuição de relevância do documento, num conjunto de n=100 documentos, foram considerados relevantes apenas m=13. A tabela abaixo demonstra através de um (*) a posição dos documentos relevantes.

Tabela 02: Resultado da recuperação após a distribuição.

5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|
.....	.....	.....	***..	.....	.....	... *.	.....	.....	.....	.. *..	.....	... **	**** .	.....	.....	.....	... *.	.....	... *.

Fonte: WEIDE, Th. P. van den. **Information Discovery,** 2001.

A tabela acima nos mostra que a posição dos documentos relevante é ruim ou seja, os documentos mais relevantes não estão no topo da lista. A precisão do retorno neste caso é 0,4350, o que demonstra que é ligeiramente pior do que a precisão média de 0,5.

Para que isto não ocorra o IRS inicia tomando aleatoriamente uma amostra de 15 documentos, e solicita ao pesquisador marcar quais documentos são relevantes dentro da amostra. Na nova distribuição, segundo a tabela abaixo, nos revela que a precisão aumentou de 0,4350 para 0,7905, após 15 solicitações do pesquisador.

Tabela 03: Resultado da recuperação após a primeira amostra randômica.

5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|	....+	....|
*****	. *...	... *.	.....	.... *	.. *..	.....	.....	.. *..	.. *..	.....	.....	*....	.....	.....	.....	.....	.. *..	.....	.....
s.ss.	s....	.. s..	.....	... s.	.....	s....	s....	s....	s....	.....	...s.	.....	.....	....s	s....	... s.	s....	.....	.....

Fonte: WEIDE, Th. P. van den. **Information Discovery,** 2001.

Após 10 etapas sucessivas pode-se analisar os resultados conforme tabela abaixo:

Tabela 04: Resultado da recuperação após a última amostra randômica.

Etapas	*Precisão*
0	0,4350
1	0,7905
2	0,8090
3	0,8462
4	0,8515
5	0,8515
6	0,8515
7	0,8515
8	0,8515
9	0,8515

Fonte: WEIDE, Th. P. van den. **Information Discovery,** 2001.

Nota-se que a partir da quarta etapa, a precisão se manteve constante. Ao passar das etapas o número de documentos com a relevância desconhecida decresce, tornando desta forma cada etapa menos efetiva, estabilizando em uma precisão de 0,815. Desta forma os documentos mais relevantes foram colocados no topo da lista com uma precisão aceitável.

3.2.3 Modelo de indexação da independência binária (BII)

Este modelo, segundo Crestani et al (1998), é uma variação do modelo BIR. Onde o modelo BIR considera uma única *query* a respeito da coleção inteira de documentos, o modelo BII considera um documento em relação a várias *queries*. O peso

do posicionamento de um termo é avaliado como uma estimativa da relevância da probabilidade daquele documento, em relação às questões usadas naquele termo.

O enfoque do BII está na representação da procura, a qual assumi ser o vetor binário $\vec{z}$. Este modelo busca uma estimativa de probabilidade P(R/$\vec{z}$,dj) que o documento dj poderá ser julgado relevante para a representação da busca de $\vec{z}$. Nesta seção usare-se $\vec{x}$ para denotar a representação do documento. Pode-se notar uma similaridade deste modelo como o BIR, a diferença está na aplicação do Teorema de Bayes :

$$P(R/\vec{z},\vec{x}) = \frac{P(R/\vec{x}).P(\vec{z}/R,\vec{x})}{P(\vec{z}/\vec{x})} \qquad (12)$$

P(R/$\vec{x}$) é a probabilidades que os documentos representados por $\vec{x}$ podem ser julgados relevantes para uma *query* arbitrária. $P(\vec{z}/R,\vec{x})$ é a probabilidade do documento ser relevante para a *query* com representação $\vec{z}$. Como $\vec{x}$ e $\vec{z}$ são assumidamente independentes, P($\vec{z}$/$\vec{x}$) reduz a probabilidade de que a *query* $\vec{z}$ seja submetido para o sistema P($\vec{z}$).

Existem vantagens e desvantagem deste modelo. Uma das desvantagens é fornecer pesos para as *queries* que não acontecem no documento. Mas o BII pode ter ótimos resultados usados com enciclopédias ou relações termo-a-termo.

Segundo Fuhr (1992), o BII não é muito aplicado, pois na maioria dos casos não haverá informação relevante o suficiente para a estimação da probabilidade P(R/$\vec{z}$,dj) para os pares de termos–documento específicos. Para superar esta

dificuldade, pode-se assumir que o documento consiste de componentes independentes (sentenças ou palavras) para qual a indexação dos pesos possam se relacionar, mas algumas avaliações experimentais nos têm mostrado apenas resultados de aproximação moderados para esta aproximação. Observando que o modelo BIR resguarda uma simples *query* para vários documentos, e o modelo BII observa um documento em relação a um número de *query* submetida ao sistema. Uma conseqüência do modelo BII pode fornecer o mesmo *ranking* para duas *queries* formuladas com o mesmo conjunto de termos. Mas a maior vantagem deste modelo, é que a representação do documento não necessita ser especificado, pois o modelo calcula a probabilidade da relevância das representações usando o Teorema de Bayes.

3.2.4 Modelo de indexação Darmstadt (DIA)

Segundo Crestani et al (1998), a idéia básica do DIA é o uso de uma indexação de pesos para o aprendizado em longo prazo. O DIA pode ser visto como uma tentativa para desenvolver uma estimativa de termos indexados, baseados no desenvolvimento de termos e na aprendizagem do experimento ou seja, na heurísica.

O DIA procura estimar P(R/xi,qk), de um modelo de julgamento relevante dos query-documento ou "termo-documento. Esta abordagem quando usada para indexar, associa um conjunto de atributos selecionados heuristicamente, cada par de termo-documento, em lugar de estimar a probabilidade associada com um termo indexado diretamente.

O uso destes atributos reduz a quantidade dos treinamentos dos dados requeridos, e permite que a aprendizagem seja uma coleção mais específica. Entretanto,

o grau da estimação para qual as estimativas resultantes são termos-específicos, depende dos atributos particulares usados.

A performance da indexação do DIA está divido em dois passos: o passo da descrição, e o passo da decisão. No primeiro passo, os pares dos termo-documento (xi, $\vec{x}$) são formados. Estas descrições relevantes (xi, $\vec{x}$) , inclui um conjunto de atributos considerados importantes para a tarefa de nomear pesos para os termos com respeito aos documentos; e no passo da decisão, é nomeado previamente, um índice de pesos probabilísticos nos dados. Esta idéia que pode ser calculado por P(R/s(xi, $\vec{x}$)) e não por P(R/xi, $\vec{x}$).

Segundo Fuhr (1992), o DIA tem tido sucesso em termos de estratégia de aproximação. No DIA, a indexação das tarefas é subdivida em descrições e decisões de procedimentos. Na descrição, as descrições relevantes dos pares do termo-documento (t_i, d_m)são formadas, onde as descrições relevante $x(t_i, d_m)$ contém os valores de atributos do termo t_i, o documento d_m e seus relacionamentos. A maior vantagem desta aproximação de indexação é a flexibilidade da representação dos documentos, sendo um fator importante quando complexos métodos de análise de textos são usados.

O resultado dos experimentos nas coleções de testes padrões, indica que a aproximação do DIA é freqüentemente superior a outros métodos de indexação, devido a eficiência da representação dos documentos.

3.2.5 Recuperação com modelo de indexação probabilístico (RPI)

Este modelo, segundo Crestani et. al. (1998), nos leva a diferentes aproximações em relação a outros modelos. Este modelo assume o uso de pesos não

apenas nos termos dos índices em relação aos documentos, mas também pesos dos termos das *queries*.

Denota-se por W_{mi} o peso dos termos índices *xi*, com respeito ao documento $\vec{x}_m$ e por V_{ki} o peso da *query* no termo *zi=xi* com respeito à *query* $\vec{z}_k$, então pode-se avaliar o produto, e usar a seguinte função de recuperação:

$$r(\vec{x}_m, \vec{z}_k) = \sum_{(xm=zk)} w_{mi} v_{ki} \qquad (13)$$

RPI foi desenvolvido especialmente para combinar pesos de indexação probabilística com pesos baseados nos pesos dos termos das *queries*, por exemplo a relevância do *feedback*. Sua vantagem principal é a sua adequada portabilidade em diferentes esquemas de indexação probabilística.

Fuhr (1993), trabalhou com modelos de relações probabilísticos para a integração da IR com banco de dados, no seu trabalho foi apresentada a Álgebra Relacional Probabilística (ARP), que combina a álgebra relacional padrão com recuperação probabilística. Neste caminho, foi alcançada uma linguagem poderosa de *query* para IR, que também facilita a integração das aplicações entre IR e banco de dados, pois estes contêm também textos. Em seus experimentos foi constatado que a ARP só permitem combinações em condições Booleanas. Isto significa que as recuperações probabilísticas, conforme o modelo BIR, não podem ser executadas com estes operadores. Desde que se esteja trabalhando com pesos indexados em documentos em nossa álgebra, houve uma integração da generalização correspondente do modelo BIR com o modelo de indexação probabilística RPI. Então nota-se que o modelo RPI pode servir para padronizar interface da *query*, e para integrar banco de dados e IRS.

3.2.6 Modelo de inferência probabilística

Segundo Crestani et. al. (1998), com a teoria da visão epistemológica probabilística, as probabilidades são definidas como a base da relação semântica entre documentos e *queries*. As probabilidades são interpretadas como graus de confiança.

A idéia geral deste modelo, começa com a definição e concepção de espaço, o qual pode ser interpretado como espaços de conhecimento para cada documento.

Termo Índice e usuários das *queries*, são representadas como proposições. Por exemplo, a proposição (d) é o conhecimento contido no documento; a proposição (q) é a necessidade do pedido de informação; e a proposição (d $\cap$ q) é a porção do conhecimento comum tanto para (d) como para (q).

Segundo Wong (1995), neste modelo assumi-se que existe um espaço conceitual ideal *U*, chamado de universo de discussão ou domínio de referência. Elementos neste espaço, considerados ser conceitos elementares à proposição, é um sub-conjunto de *U*. A função de probabilidade *P* é definida no conceito espacial *U*, logo na função espistemológica probabilística, *P* é definido como um conceito espacial. Por exemplo, P(d) é o grau para qual o conceito espacial é englobado pelo conhecimento contido no documento e P(d $\cap$ q) é o grau para qual o conceito espacial é englobado pelo conhecimento comum do documento e a *query*.

Wong (1995) relata em seus artigos, a importância da visão epistemológica probabilística, particularmente ao nível conceitual de modelagem de recuperação probabilística com inferência probabilística, enquanto que uma visão aleatória é útil na implementação de níveis, para a estimar os cálculos das probabilidades exigidas.

Wong (1995) conclui principalmente, que muitos modelos probabilísticos podem ser compreendidos dentro do sistema de inferência probabilística. Dependendo

do conhecimento da estrutura dos espaços e da interpretação da relevância probabilística, documentos e esquemas das representações das *queries* poderão ser aceitos. As diferenças entre os vários modelos, podem ser traçadas entre as diferentes considerações feitas no modelo de inferência probabilística.

3.2.7 Modelo de regressão logística organizada (SLR)

Segundo Crestani et. al. (1998), a organização do modelo logístico de regressão (SLR), é uma tentativa de superar alguns problemas usando métodos de regressão padrão para calcular probabilidades e relevâncias em IR.

A teoria da regressão padrão é baseada na suposição que os valores amostrais, para as variáveis dependentes, é de uma quantidade contínua de possíveis magnitudes. Em IR, as variáveis dependentes são usualmente dicotômicas, ou seja, o documento é relevante ou irrelevante. A regressão padrão é inapropriada para estes casos. A ferramenta mais apropriada, para tratar dicotomias de variáves dependentes, é a regressão logística e o método estatístico.

O método proposto representa uma tentativa de tratar as combinações de pistas compostas, em pelo menos dois níveis: um nível o qual a previsão estatística é estimada separadamente para cada pista composta, e um outro nível a qual estas separações estatísticas são combinações para obter uma estimativa da relevância probabilística para a *query* e documentos apresentados.Como isto procede em fases, o método é chamado de Regressão Logística Organizada (SLR).

A análise dos processos dos dois estágios do SLR pode ser apresentada através dos seguintes passos: passo 1, uma concepção de hipótese estatística mais simplificada é usada para interromper a união das complexas expressões probabilística,

transformando-as em expressões mais simples para as pistas compostas; passo 2, a análise de regressão logística em uma amostra de aprendizado é usada para obter uma equação para estimar os valores das equações posteriores; passo 3, uma segunda análise de regressão logística, baseada na mesma amostra de aprendizagem é usada para obter uma previsão de outras regras para combinar as pistas compostas e correção de qualquer preconceito sistemático introduzido pela suposição de simplificação.

O método SLR pode ser perfeitamente usado em quaisquer das direções. Supondo por exemplo que a correção padrão uniforme para o termo de dependência apresentado neste estudo, através das equações, é profundamente inadequado, e umas correções adicionais aplicados em um grau individuais de interdependência são necessárias em particular nestes casos. Para obter-se pelo menos algumas melhoras ao longo deste trabalho, tudo que é necessário é uma forma de medir a média da quantidade dos parâmetros dos termos de interdependência, que são apresentados entre os termos envolvidos na análise de uma *query* apresentada. Quaisquer umas destas medidas da associação estatística pode servir para este propósito, aplicando a regra da variável independente adicional na equação de regressão do segundo passo. É claro que uma elaboração desta natureza pode ter um significante custo em termos de complexidade computacional adicional.

3.2.8 Modelo de indexação de n-Poisson

Este modelo segundo Crestani et al (1998), declara que a indexação probabilística é uma extensão de n dimensões do modelo 2-poisson, onde o modelo 2-Poisson é o resultado de uma procura de distribuição estatística de temos potenciais, em duas classes de documentos.. Se o número de ocorrências de um termo dentro do

documento é indeterminado, dependendo se o documento é relevante, e o número de ocorrências daquele termo pode ser modelado usando o conhecimento distribuído, então é possível decidir se o termo deve ser nomeado a um documento determinando e para qual das duas distribuições o termo pertence.

Pode-se estender a idéia do caso n-dimensional. Supõe-se que exista n classe de documentos a qual o termos xi apareça com diferentes freqüências, de acordo com a extensão da cobertura do tópico relatado para aquele termo específico. A distribuição dos termos dentro de cada classe é governada por uma única distribuição de Poisson.

Sugere-se um estudo da forte evidência da distribuição de n-Poisson ser usado como a base de uma modelagem estatística precisa, em uma grande coleção de documentos.

Segundo Margulis (1992), a freqüência da ocorrência dos termos referenciais do texto dentro de um documento textual completo em uma coleção de documentos, pode ser descrito pela soma de distribuições de Poisson. Cada resultado desta soma é uma distribuição de Poisson independente, que descreve a freqüência da ocorrência dos termos referenciais do texto dentro do subconjunto de documentos, pertencente ao mesmo nível de pesquisa dos tópicos relacionados àqueles pontos referenciais. A probabilidade da escolha aleatória do documento D conter k ocorrências de um certo ponto referencial textual h é dado por:

$$P(dfreq(D,h)=k)=\sum_i \eth_i \frac{\dot{e}_i^k}{k!} e^{-\ddot{e}} \qquad (14)$$

onde i representa a classe da pesquisa dos tópicos relacionados aos pontos referenciais do texto; $ë_i$ é a extensão média da cobertura dos tópicos relacionados aos pontos referenciais do texto, dentro da classe C_i; $ð_i$ é a probabilidade do documento pertencer à classe $C_{i;}$; e $\sum_i ð_i = 1$.

Margulis (1992), em seus experimentos usou uma Sun 4/490 SPARC server para estimar o parâmetro *nP* para cada palavra e termos em suas coleções, e então testou os parâmetros calculados usando teste $÷^2$ (qui-quadrado) do ajuste. A performance da estimação do algoritmo usado depende dos valores iniciais estimados $(ë_i \; e \; ð_i)$. Para estimar os valores iniciais destes parâmetros randômicos foram usadas aproximações usadas por KARP (1990) para determinar se interrompe o processo de estimação ou fazer outra tentativa com um conjunto diferente de valores iniciais. Se a estimação ou a boa qualidade no teste de ajuste falhar para um conjunto inicial de valores, outro conjunto é gerado, a estimação no teste de ajuste é repetida em até 10 vezes. Se após 10 tentativas não for achada nenhuma estimativa a boa qualidade do ajuste, o número de componentes de Poisson é aumentado e o processo é repetido. Segundo os testes efetuados por MARGULIS (1992), a complexidade de tempo do algoritmo é no pior caso, assintoticamente (que se aproxima, mas nunca chega a ele) proporcional ao produto do número de documentos e ao número de iterações máximas estimadas pelo algoritmo. Foi também constatado que acima de 70% da ocorrência das palavras e termos, a distribuição é realmente efetuada de acordo com *n-Poisson* - Distribuição Múltipla de Poisson. Então, MARGULIS (1992) mostrou a forte evidência da distribuição de Possion ser uma base precisa e modelo probabilístico útil para grandes coleções de documentos.

3.2.9 Lógica não-clássica para IR

Segundo Crestani et al (1998), em 1986 van Rijsbergen propôs um paradigma para a Recuperação de Informação Probabilística no qual IR foi considerado um processo de suposição incerta. O paradigma é baseado na suposição das *query* dos documentos poderem ser considerados como uma fórmula lógica para responder uma *query*, ou seja, um sistema IR pode prover uma *query* para os documentos. Isto significa que um documento é relevante para a *query* se esta for relevante para a *query*, em outras palavras, se a lógica da fórmula ($d \rightarrow q$) pode ser provado que é segura.

A introdução de incerteza vem para considerar que uma coleção de documentos não pode ser considerada consistente a um conjunto completo de declarações. De fato, documentos na coleção podem contradizer um ao outro em qualquer lógica particular, e nem todo o conhecimento é avaliado. Para isto, foi mostrado que a lógica clássica comumente usada, não é adequada para representar *queries* e documentos, porque a incerteza está intrinsecamente presente em IR. Então, van Rijsvergen propôs um princípio da incerteza lógica que nos relata que: dadas duas sentenças (x e y), a medida da incerteza de ($y \rightarrow x$) relacionado a um determinado conjunto de dados, é determinado pela extensão mínima o qual tem que se somar informação para o conjunto de dados, para estabelecer a verdadeira relação entre ($y \rightarrow x$).

O princípio não diz nada como “incerteza” e “mínimo” pode ser quantificado. Entretanto, foi sugerida uma teoria de aproximação de informação. Esta teoria pode ser vista como uma generalização do método Bayesiano, e pode ser usada em alguns modelos de IR, será relatado a seguir.

3.2.10 Redes bayesianas

As redes Bayesianas, segundo Baeza Yates e Ribeiro Neto (1999), são gráficos acíclicos ordenados, a qual as ligações (nós) representam variáveis aleatórias, o arco retrata o relacionamento entre estas variáveis, e as forças dessa influência são expressas por probabilidade condicional. Os parentes das ligações (*child node*) são aqueles julgados ser de origem direta . Este relacionamento está representado por uma comunicação (*link*) direcionada para um outro nó-*parente, os child nodes*. As rotas da rede são "nós-parentes" exteriores.

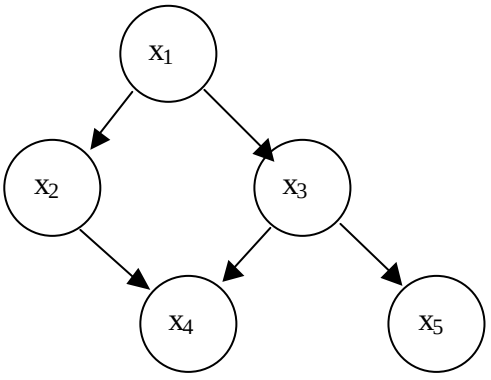
Seja xi um nó da rede Bayesiana, G e T_{xi} um nó-parente (*child-node*) de xi. A influência exercida de T_{xi} sobre xi, pode ser especificada por algum conjunto de funções $F_i(xi, T_{xi})$ que satisfaz:

$$\sum_{\forall xi} F_i(xi, T_{xi}) = 1 \qquad 0 \le F_i(xi, T_{xi}) \le 1 \qquad (15)$$

onde, xi se refere ao estado da variável aleatória associada ao nó xi.

Esta especificação está completa e consistente porque o produto $\prod_{\forall i} F_i(xi, T_{xi})$ constitui a união probabilística distribuída por todos os nós em G.

Figura 10 – Exemplo de uma rede Bayesiana



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Segundo a figura acima, as redes Bayesianas formam uma união de distribuição de probabilidade, $P(x_1,x_2,x_3,x_4,x_5)$. Neste caso a dependência declarada na rede, está em permitir uma expressão natural da união de distribuição probabilística em termos de condições de probabilidade local, como segue:

$$P(x_1,x_2,x_3,x_4,x_5)=P(x_1)\ P(x_2|x_1)\ P(x_3|x_1)\ P(x_4|x_2,x_3)\ P(x_5|x_3) \qquad (16)$$

A probabilidade $P(x_1)$ é chamado de probabilidade superior ou anterior para uma rede e pode ser usado para modelos de conhecimento prévio sobre semânticas e suas aplicações.

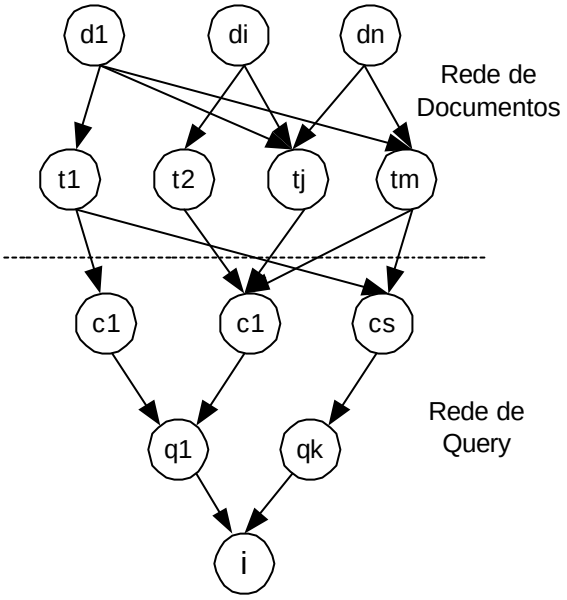
Existem dois modelos de IR baseados em Redes Bayesianas. O primeiro modelo é chamado de Redes de Inferência e provê a teoria básica do mecanismo de um IRS. Este sucesso tem atraído a atenção dos usuários das Redes Bayesianas em IR. O segundo modelo é chamado de Redes de Confiança, é uma generalização do primeiro modelo. Estes dois modelos serão discutidos a seguir.

3.2.11 Modelo de rede de inferência

Segundo Crestani et al (1998), quando IR foi considerado um processo de inferência incerta, os cálculos da relevância da probabilidade, e a noção geral de relevância, tornaram-se mais complexos. Relevância é relacionada ao processo inferencial, por qual pode-se encontrar e avaliar a relação entre o documento e a *query*.

O formalismo probabilístico que descrever a relação de inferência com a incerteza, é proveniente das Redes de Inferência Bayesianas.

Figura 11 – Redes de Inferência para IR



Fonte: CRESTANI, F. et al. “Is this document relevant? ... probably”: a survey of probabilistic models in information retrieval. **ACM Computing Surveys**, New York, v.30, no.4, p. 528-552, Dec. 1998.

Na figura acima pode-se notar que os *nós* representam entidades de IR tais como: documentos, termos índices, conceitos, *queries* e identificação do documento. Pode-se escolher o número e o tipo de *nós* que se deseja trabalhar, e de acordo com a complexidade das representações das coleções de documentos, e da necessidade de informação. Arcos representam a dependência probabilística das entidades. Eles representam uma probabilidade condicional, isto é, a probabilidade de uma entidade ser verdadeira determina que seus parentes também são.

As redes de inferência são usualmente divididas em dois componentes de redes: as redes de documentos e redes de *queries*. As redes de documentos, o qual representa a coleção de documentos, é construída uma vez através de determinada

coleção e esta estrutura não pode ser modificada. As redes das *queries* são construídas para cada necessidade de informação, e pode ser modificada e estendida a cada sessão pelo usuário, de um modo interativo e dinâmico. As redes das *queries* são anexadas à rede estática de documentos para processar uma *query*.

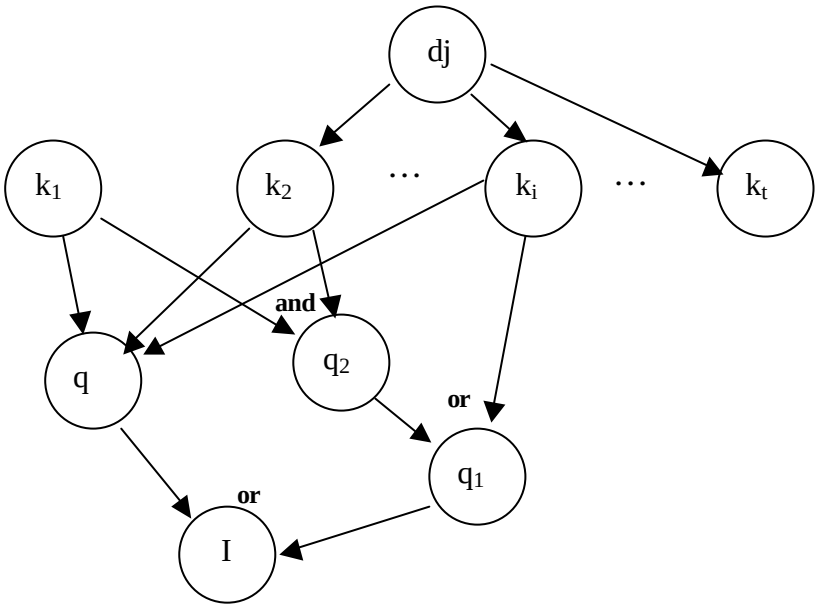
Uma característica particular deste modelo, é garantir que a exploração da representação de múltiplos documentos e *queries*, possa ser usada dentro do contexto de uma coleção particular de documento. Além disso, dado uma única necessidade de informação, é possível combinar resultados de múltiplas *queries* e estratégias de procura múltipla.

Segundo Baeza Yates e Ribeiro Neto (1999), as duas escolas mais tradicionais em pensamento de probabilidade são baseadas em visões frequentivista e visões epistemológicas . A frequetivista visa interpretar a probabilidade como uma noção estatística relacionada às leis do acaso. A epistemológica visa interpretar a probabilidade como um grau de convicção cuja especificação poderia ser destituída de experiências estatísticas. Esta segunda visão é muito importante, porque freqüentemente refere-se à probabilidade em nossa vida diária sem uma clara definição de uma experiência estatística.

O modelo de rede de inferência, trata a visão epistemológica como um problema de IR. Associa variáveis aleatórias com termos índices, os documentos e as perguntas dos usuários. As variáveis aleatórias associadas ao documento dj, representam o evento de observação dos documentos, isto é, o modelo assume os documentos que estão sendo observados, e procura por documentos relevantes. Observando o documento dj, pode-se afirmar a associação das variáveis aleatórias com os termos índices. Assim, o documento é a causa do aumento nas variáveis com os termos índices. Os termos

índices e as variáveis dos documentos são representados por nós na rede. Gradativamente os nós dos documentos são direcionados para os nós termos indicados, para observar se os rendimentos de documento melhoraram em seus nós.

Figura 12 – Exemplo de uma redes de inferência



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

Na figura acima, o documento dj possui k_2, k_i e k_t termos índices. Ele é modelado e direcionado do nó dj para os nós k_2, k_i e k_t. O pedido de restauração da *query* (q), é composto dos Índices Termo k_1, k_2 e k_i. Ele é modelado e direcionado dos nós k_1, k_2 e k_i para o nó q. Observando a figura, nota-se que existe três nós extras, q_1, q_2 e I. Os nós q_1, q_2 são usados para modelar uma função (alternativa) Booleana q_1 para o pedido q , onde $q_1 = ((k_1 \wedge k_2) \vee k_i)$. Quando tal informação está disponível, existe a necessidade de I ser apoiado por q e q_1.

Turtle e Croft (1991), relataram que as representações de rede têm sido usadas em IR desde 1960. IR é um processo de inferência que pode estimar a

probabilidade da necessidade do usuário expressas em uma ou mais *queries*, retornando os documentos relevantes.

Pela sua eficiência, este modelo foi adotado pelo sistema INQUERY de IR. Os teste, resultados e eficiência serão vistos no decorrer deste estudo.

3.2.13 Modelo de redes de confiança (Belief)

Segundo Baeza Yates e Ribeiro Neto (1999), o modelo de rede confiança foi introduzido em 1996, baseado em interpretação epistemológica de probabilidade.

Considere-se $K=\{k_1, \ldots, k_t\}$ o universo da discussão que define o espaço de amostra para o modelo acima. Seja $(u) \subset K$ um sub-sistema de K. Para cada sub-sistema u está associado o vetor $\vec{k}$ tal que $g_i(\vec{k})=1 \leftrightarrow k_i \in (u)$.

Para cada termo de índice k_i está associado uma variável aleatória binária que também é chamado de k_i. Esta variável $k_{i,}$ está fixado com valor 1 indicando que o índice k_i é um membro do conceito de representação de k_i.

O documento $d_{j,}$ na coleção, está representado como um conceito composto de termos o qual são usados para o índice de d_j. Analogamente, a *query* (*q*) do usuário está representada como um conceito composto de termos que são usados para o índice (*q)*.

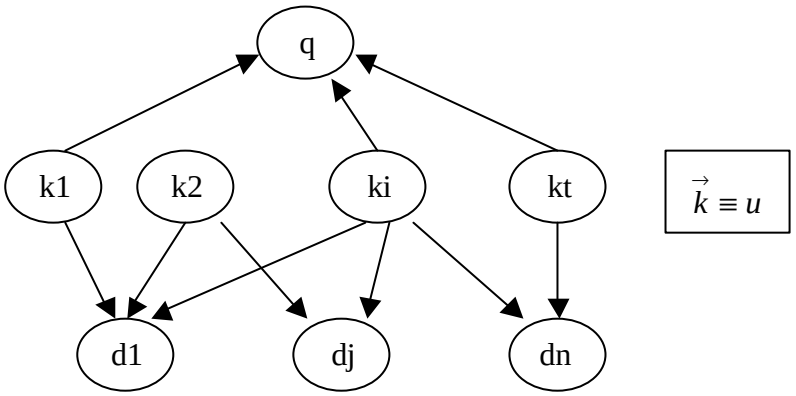
Neste tipo de rede, o pedido de restauração de informação *query* (q) está modelado com um nó da rede que está associado a uma variável aleatória binária, que é também chamado de (*q*).

Esta variável está fixada com valor 1(um), não importando quando q completamente cobre o conceito do espaço *K*. Assim, quando avalia-se *P(q*), computa-se o grau de convergência do espaço *K* para (*q*).

O documento *dj* é modelado como um nó da rede, associado a uma variável aleatória binária que também é chamado de *dj*. Esta variável é 1 (um) e indica que *dj* abrange o conceito do espaço *K*. Quando se avalia *P(dj)*, computa-se o grau de convergência do espaço *K* para o *dj*.

O usuário da *query* e os documentos contidos na coleção são modelados em sub-sistemas nos termos índices. Cada um desses sub-sistemas está interpretado como um conceito incluso no conceito do espaço *K*. Além disso, o usuário da *query* e documentos são modelados identicamente.

Figura 13 – Modelo básico de redes de confiança



Fonte: BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

A figura acima nos mostra o modelo de redes de confiança. Neste tipo de rede, a *query* (q) é modelada como uma variável aleatória binária, a qual está apontada para os nós dos temos índices, que é composto por conceitos da *query*. Os documentos são tratados da mesma forma para os usuários da *query*, ambos são conceitos do espaço *K*.

Assim, ao contrário dos modelos de redes de inferência, o documento no nó está apontado para os índices dos termos que compõe o documento. A posição de um

documento *dj* relativo a uma determinada *query* (*q*) está representado como um conceito que combina relacionamento e o grau de convergência ao conceito *dj* pelo conceito (*q)*.

Ribeiro e Muntz (1996), efetuou experimentos com três estratégias distintas de recuperação de informação: modelo vetorial clássico, modelo vetorial usando *queries* expandidas com índices relacionados (modelo também chamado de Vector-exp), e modelo de redes de confiança usando *queries* expandidas com índices relacionados, que são determinados pela correlação de fatores (modelo também chamado de redes de comunicações – *Network*). O modelo de redes de confiança mostrou algumas vantagens: este modelo é fundado em um espaço claramente definido tornando-o intuitivo; este modelo é derivado de considerações probabilísticas em cima deste espaço de amostra, simplificando a sua compreensão; este modelo pode ser visto como uma alternativa para modelos de redes de inferência proposta por (TURTLE E CROFT 1991).

No ponto de vista teórico, o modelo de redes de confiança é mais comum, por que pode reproduzir qualquer estratégia de ranking, enquanto que o inverso não é verdade.

4 SISTEMAS PROBABILÍSTICOS DE RECUPERAÇÃO DE INFORMAÇÃO EM BASE TEXTUAL

Existem alguns sistemas probabilísticos em IR, mas muitos deles foram elaborados localmente para as necessidades somente de algumas empresas. Neste trabalho foram pesquisados sistemas de IR avaliados e testados pela TREC, a fim fornecer um comparativo somente de sistemas aprovados e qualificados por um órgão competente.

Após pesquisas efetuadas em empresas filiadas a TREC, passa-se a discutir três sistemas que usam modelos probabilísticos para IR: o sistema INQUERY, THISL e o OKAPI. Estes sistemas foram implantados e melhorados ao decorrer dos anos, em virtude dos testes e análises efetuados pela TREC. Apesar do sistema THILS de IR usar alguns componentes da OKAPI, ele foi analisado neste trabalho por possuir a característica de solicitação de informações através do comando de voz.

Hoje existem alguns sistemas que foram elaborados baseados nos três sistemas probabilísticos de IR relacionados neste trabalho, estes não foram mencionados por usarem os mesmos algoritmos, e terem desempenho similar.

O sistema INQUERY foi desenvolvido pelo Departamento de Ciências da Computação da Universidade de Massachusetts nos Estados Unidos. Este sistema baseia-se em um modelo de recuperação probabilística chamado de Redes de Inferência, usando arquitetura de redes Bayesianas.

O sistema THILS foi desenvolvido pelo Departamento de Ciências da Computação da Universidade de Sheffield no Reino Unido. Este sistema baseia-se em um modelo probabilístico padrão usando lista de parada com 132 palavras, e uma lista adicional com 78 palavras quando a query e processada.

O sistema OKAPI foi desenvolvido primeiramente pelo Centro Politécnico de Londres, agora Universidade de Westminster, e posteriormente pela Microsoft Research,

Cambridge (UK). Okapi é simples e robusto, baseado no modelo probabilístico generalizado com facilidades para uma busca completa, mas também de grande abrangência para operações Booleanas determinísticas e operações semi-Booleanas.

4.1 Sistema de Recuperação INQUERY

4.1.1 Introdução

Com os bancos de dados textuais ficando cada vez maiores e heterogêneos, a pesquisa da IR dependerá cada vez mais do desenvolvimento da força, eficiência e flexibilidade dos mecanismos de recuperação. O sistema chamado INQUERY, é baseado em modelos de recuperação probabilísticos e oferece suporte para sofisticadas indexações e complexos sistemas de procura. INQUERY tem sido usado sucesso com banco de dados com aproximadamente 400.000 documentos.

O crescente interesse em técnicas sofisticadas de IR, tem conduzido acessos a um grande número de dados, tornando-se disponível para pesquisa. O tamanho destes bancos de dados, em ambas as situações, tanto no tamanho do número de documentos contido neles, quanto no tamanho dos documentos que ali foram digitados, que tipicamente são "*full text*", tem apresentado significante desafio para os pesquisadores da IR, onde são usados para experimentos, dois ou três mil resumos de documentos.

Para desenvolver a pesquisa com diferentes tipos de representações de texto, modelos de recuperação, técnicas de aprendizagem e interfaces, uma nova geração de mecanismos de recuperação eficientes, flexíveis e potentes, precisam ser elaborados. Tem-se desenvolvido tal sistema nos últimos dois anos, no laboratório de Recuperação de Informação da Universidade de Massachusetts.

O Sistema INQUERY é baseado em um modelo de recuperação probabilística chamado de Redes de Inferência, usando arquitetura de redes Bayesianas. Este modelo é potente, de forma que ele pode representar muitas semelhanças, e as combinam em um sistema simples. Ele também oferece uma habilidade para especificar as necessidades de um IR complexo e as compara com um documento.

Segundo Allan et. al. (2000), todas as aproximações permanecem como nos anos anteriores, isto é, a versão atual do INQUERY usa a mesma pesagem de CALLAN et al (1992). Por este motivo enfoca-se, segundo CALLAN et al (1992), o projeto e a implementação do sistema INQUERY, que tem designado pesquisas para grandes bancos de dados. Apresentar-se-á uma amostra do projeto INQUERY, seguido de detalhes e descrições dos seus sistemas mais importantes. Ao longo desta descrição estar-se-ão experiências, uma delas recente com 1 Gigabyte de dados que contém cerca de 400.000 documentos, variando de pequenos resumos a relatórios de 150 páginas. Relatar-se-á algumas inovações e resultados do INQUERY, proposto por ALLAN (1999,2000) na TREC-8 e TREC9

As redes de Inferência Bayesianas são modelos probabilísticos de evidente raciocínio, que tem sido usado amplamente nos últimos anos. Uma Rede de Inferência Bayesianas ou também chamado de *Bayes net*, é um Gráfico Acíclico Direcionado, no qual os *nós* representam variáveis (proposições) sequenciais e os arcos representam dependências. Um valor de um nó é uma função dos valores dos nós na que depende. Nós esquerdos normalmente representam (proposições) valores seqüenciais, e podem ser determinados através de observações. Outros nós normalmente representam (proposições) seqüenciais cujos valores devem ser determinados pela inferência.

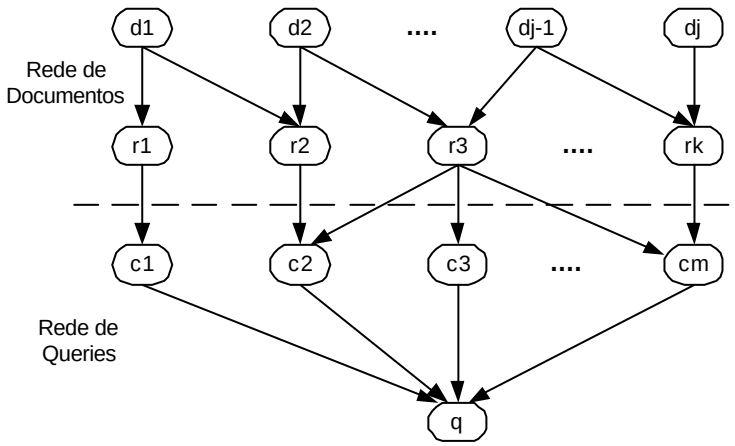
A notável característica das redes de Bayes, é que dependências não são absolutamente necessárias. Certeza ou probabilidade pode ser representada através de pesos em arcos.

INQUERY é baseado em um tipo de Redes Bayesianas chamado de Redes de Recuperação Inferencial de documentos ou Redes de Inferência. As Redes de Inferência consistem em dois tipos de redes: a primeira para documentos e a segunda para as *queries*. Nós (nodes) nas Redes de Inferência, podem ser verdadeiro ou falso. Valores determinados para os arcos variam de 0 a 1, e são interpretadas como valores absolutos.

4.1.2 A rede de documentos

Uma rede de documentos pode representar um conjunto de documentos com diferentes técnicas de representações, variando de níveis de abstrações. A figura abaixo mostra uma rede simples de documentos com dois níveis de abstração: o documento de texto de nível *d*, e a representação com conteúdo de nível *r*. Níveis adicionais de abstração são possíveis, por exemplo representação de áudio ou vídeo, mas não são comuns no INQUERY.

Figura 14:Rede de inferência simples de recuperação de documentos



Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

Um nó de um documento d_i representa a proposição se satisfaça a procura do usuário. Os nós de documento são designados com valores verdadeiro(*true*). O valor em um arco entre o documento texto d_i, e o nó de representação r_k, é a probabilidade condicional $P(r_k/d_i)$. Previamente, a probabilidade do documento recebe valor 1.

O conteúdo de um nó de representação r_k , apresenta a proposição de que um conceito foi observado. O nó pode ser verdadeiro(*true*) ou falso(*false*). O valor em um arco entre o conteúdo da representação do nó r_k e o conteúdo do nó da *query* c_j , é a certeza da proposição.

INQUERY usa vários tipos de representação de nós. O mais simples corresponde a uma única palavra do documento texto, enquanto que conceitos mais complexos incluem números, datas e nomes de companhia.

4.1.3 A rede "query"

A rede *query* representa a necessidade por uma informação. Os nós das *queries* representam a proposição de que uma necessidade de informação seja satisfeita. Os nós das *queries* são sempre verdadeiros. Os nós de conceito representam a proposição de que um conceito é observado em um documento. Os nós de conceitos podem ser verdadeiros ou falsos. As redes *query* são anexadas à rede de documentos por arcos, entre os nós de conceito e nós de representação de conteúdo.

O mapeamento nem sempre é um-a-um, porque os nós de conceito podem não definir explicitamente conceitos representados na rede de documentos. Por exemplo, o termo INQUERY pode ser usado para definir um conceito que não é representado explicitamente na rede de documentos.

A habilidade para especificar conceitos de *query* ao longo do tempo, é uma das características que distingue a inteligente recuperação de informação em base de dados.

4.1.4 A matriz de comunicação

As redes de inferência de IR, como as redes de Bayes das quais elas foram originadas, permitem especificar funções arbitrariamente complexas, para analisar a certeza em uma proposição, dada verdadeira as orientações em seus respectivos nós paternos.

Estas funções são às vezes chamadas de *Link Matrices*. Se a certeza para cada combinação de evidência fosse especificada diretamente, um *Link Matrix* para um nó com n paternos seria de tamanho $(2,2)^n$. Este problema pode ser evitado restringindo os meios nos quais a evidência foi combinada. INQUERY usa um pequeno conjunto de operadores descritos posteriormente, no qual até mesmo expressões escondidas e não contidas no texto principal do documento podem ser encontradas.

4.1.5 Visão geral do projeto

O INQUERY usa para calcular a relevância do termo (*t*) em relação ao documento (*d*) a seguinte fórmula abaixo:

$$w_{t,d} = 0,4 + 0,6 * \frac{tf_{t,d}}{tf_{t,d} + 0,5 + 1,5\frac{length(d)}{avg\ len}} * \frac{\log\frac{N + 0,5}{n_t}}{\log N + 1} \qquad (17)$$

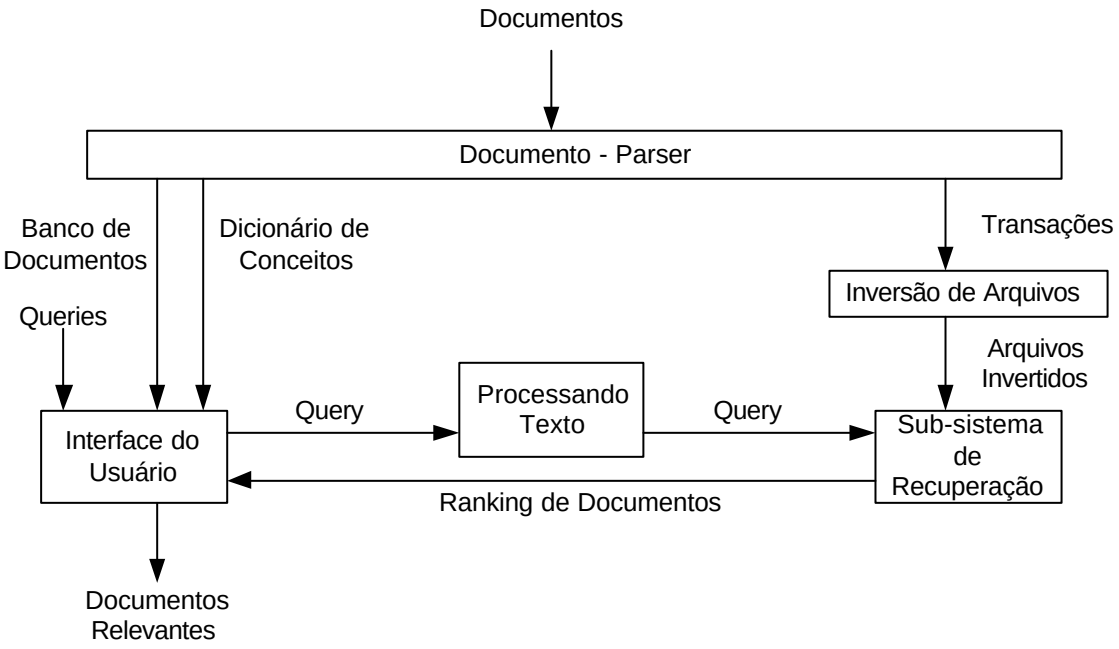
Onde n_t é o número de documentos contendo o termo (*t*), N é o número de documentos na coleção, *"avg len"* é o comprimento médio (das palavras) dos

documentos na coleção, *length(d)* é o comprimento (das palavras) do documento (*d*), $tf_{t,d}$ é o número de vezes que o termo (*t*) ocorre no documento (*d*). O componente "*tf*" usualmente se refere à função "Okapi tf" de Stephen Robterson no sistema de IR Okapi.

As principais tarefas executadas pelo sistema INQUERY são, a criação de redes de documentos, criação de redes de *query* e o uso de redes para recuperação de documentos. A rede de documentos é criada automaticamente através do mapeamento de documentos sobre o conteúdo dos nós de representação e armazenamento dos nós em um arquivo invertido para uma recuperação eficiente. As redes de *query* são especificadas para o usuário através de uma interface.

A recuperação de um documento é executada usando recursos de inferência para propagar os valores certos através da rede de inferência, que estão listados superiormente. A seguir, pode-se ver os componentes principais do sistema INQUERY, e como a informação flui entre eles.

Figura 15: A arquitetura do Sistema de Recuperação de Informação INQUERY.



Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

4.1.6 O sub-sistema corretor

A primeira tarefa da construção de rede de documentos, é o mapeamento de cada documento para o conjunto de representação dos nós. Este processo de mapeamento se refere à correção (*parsing*) dos documentos, e consistem em cinco etapas: análise léxica, análise sintática, identificação de conceitos, armazenamento de dicionários e geração de transações. É importante que cada uma dessas etapas seja eficiente, porque a construção de uma rede de documentos é uma das partes mais demoradas na construção de uma rede de inferência.

O sistema de análise do INQUERY possui elevado nível de conceito, e requer 19,8 horas de processamento em um SPARserver 490 da SUN com 128 Mbytes de memória para analisar 1 Gbyte de coleção de documentos. A seguir ver-se-á cada componente desta implementação.

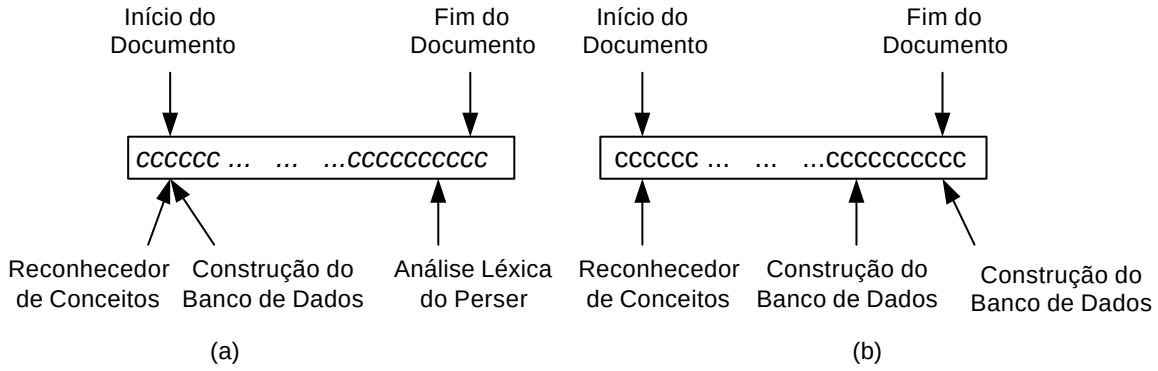
4.1.7 Análise léxica e sintática

Existem três finalidades distintas para o uso da analise léxica no INQUERY. O analisador léxico do "parser" (*parser´s lexical analyzer)* fornece sinais léxicos (geralmente palavras ou marcadores de campos) para a análise sintática. A análise léxica consiste no processo de converter uma seqüência de caracteres (o texto dos documentos) numa seqüência de palavras (as possíveis palavras candidatas a serem termos índices). Em uma análise léxica os números por si não são bons termos, por serem vagos e sem contexto, e a conversão de todo o texto para maiúsculas ou minúsculas é importante. O gerador de banco de dados (*database builder*) armazena os documentos texto em um banco de dados para o uso das interfaces dos usuários. O analisador de conceito (*concept analyzer*) analisa o alto nível de conceitos, por exemplo

datas e nomes que ocorrem no texto. As atividades destas análises léxicas são livremente coordenadas por um gerente de análise léxica.

Uma razão de se ter muitas análises léxicas, é que o INQUERY atualmente possui um programa de análise textual para seis formatos diferentes de documentos. A carga de suporte de muitos formatos de documentos é minimizada mantendo-se o provedor de dados, e ignorando os formatos dos documentos através dos analisadores de conceitos. O gerenciador de análises léxicas reforça este conceito controlando o acesso do fluxo de entrada. O gerenciador lê grandes trechos de texto dentro do buffer interno, do qual os analisadores léxicos fazem a sua leitura. Quando o novo documento é localizado, é fornecido ao corretor o acesso exclusivo do documento, como mostra a seguir.

Figura 16: Análise Léxica e Construção do Banco de Dados.



Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

O analisador parser é responsável pela conversão em formato canônico, todos os marcadores de campo encontrados no documento. Quando o analisador parser chega ao fim do documento, é fornecido ao outro analisador, o acesso ao documento, como mostra a figura anterior.

O analisador parser possui duas importantes obrigações além da conversão de documentos para a forma canônica, ou seja, em uma forma padrão para o sistema. Ele é responsável pela provisão de *tokens* (sinais), usualmente palavras, números ou marcadores de campos para a análise sintática. Ele é responsável também pela conversão de palavras para minúsculas, descartando uso de palavras de interrupção (a, and, the), e palavras de conclusão (-ed, -ing) antes de notificar o gerenciador de transação sobre a ocorrência de cada palavra.

O principal uso da análise sintática no INQUERY, é assegurar que um documento está em um formato esperado, e prover a recuperação de erros.

4.1.8 Conceitos de reconhecimento

INQUERY é atualmente capaz do reconhecimento e transformação dentro do formato canônico, quatro tipos de conceitos: números, datas, nome de pessoas e nomes de companhias. INQUERY também contém o conceito de reconhecimento, que é capaz de identificar e gravar a localização das sentenças e limites dos parágrafos. Conceitos de reconhecimento tende a ser complexo, assim é necessário implementá-los e deixá-los tão eficiente quanto possíveis. Todos os conceitos de reconhecimento do INQUERY estão atualmente em estado limitado, criado pelo autômato LEX – *Lexical Analyzer Generator*. Principalmente, é possível a combinação do reconhecimento dentro de um estado finito de automação, entretanto LEX não pode criar autômato de tamanho exigido.

O INQUERY usa uma seqüência aritmética para evitar erros no número de reconhecimento. O mapa de reconhecimento traça diferentes expressões de um conceito (e.g. 1 milhão, ou 1000000, ou 1.000.000) dentro de um formato canônico.

O reconhecimento do nome da companhia é similar também, mas menos sofisticada. Olhando para uma seqüência de palavras em maiúsculas, que é um identificador legal que geralmente acompanha nomes de companhia (Co, Inc, Ltd, ou SpA). Se o nome da companhia acontece uma vez com um identificador legal, o reconhecimento pode identificar todos as outras ocorrências dos nomes em um documento. Esta performance ocorreu com freqüência, e forneceu bons resultados em nossos testes.

O reconhecimento do nome da pessoa usa estratégia similar ao reconhecimento do nome da companhia, exceto aos que ocupam títulos de ocupação ou títulos honoríficos. Esta estratégia ocorreu em uma forma ruim em nossos experimentos. Estar-se-ão providenciando a troca do atual algoritmo com um que atua mais perfeitamente em grandes banco de dados de nomes de pessoas.

O atual reconhecedor de limites de parágrafos e sentenças, é capaz apenas de reconhecer limites que estão explicitamente determinados através de um marcador de campo. A recuperação destes limites é guardada em um arquivo, de modo que possa ser usado em um projeto para recuperação de parágrafos e sentenças de grandes documentos sem marcadores de campo.

Em princípio, não há limite para o número e complexidade para o conceito de reconhecimento que possa ser adicionado ao INQUERY. Por exemplo, investiga-se o uso de uniões estocásticas para a identificação automática de frases. A principal conseqüência deste conceito adicional é a sobrecarga que eles acrescentam ao processo parser. O atual conjunto reduz a velocidade do analisador parser em aproximadamente 25%.

4.1.9 Conceitos de armazenamento

Os analisadores léxicos são designados para trabalhar eficientemente com uma seqüência de caracteres, mas o restante do INQUERY não. Quando uma decisão é feita para selecionar um documento através de uma palavras ou conceito de nível maior, a seqüência de caracteres é trocada pelos termos de entrada selecionadas em um dicionário. Uma referência para uma entrada numérica usa menos espaços e pode ser manipulado mais eficientemente do que uma seqüência de caracteres. Se a palavra já existe, o número da entrada existente é retornado, se não uma nova entrada é criada.

INQUERY originalmente armazena seus dicionários em uma estrutura de dados em uma árvore balanceada (*B-tree)*. Porém a análise da performance mostrou que o dicionário pode ser um gargalo. A atual versão do INQUERY armazena seus dicionários em numerosas e pequenas tabelas. Esta única mudança reduziu o tempo necessário para um parser, de uma coleção de 338 Mbyte de documentos de 16,8 horas de CPU para 8,2 horas de CPU.

4.1.10 Geração de transação

A cada momento um termo é identificado no meio do analisador gramatical léxico ou no identificador de conceitos. Seu local é informado para o gerente de transação, repassando apenas um tópico por vez. Quando o fim do documento é atingido, o gerente de transação grava no disco o conjunto das transações indexadas, que registra cada termo de freqüência e a localização de sua ocorrência naquele documento.

As transações são normalmente armazenadas em arquivos de texto usando um método de codificação adequado. Experimentos com grandes coleções, têm

produzido mais transações que ajustes em um de nossos discos. O gerenciador de transação lida com este problema através da criação de um novo arquivo de transação a cada momento que o analisador gramatical do INQUERY é invocado. (Uma invocação do parser pode analisar muitos documentos). A periódica criação de novos arquivos de transação nos possibilita propagar através de vários discos.

4.1.11 Inversão de arquivos

Cada transação representa uma união entre um nó de documento e um nó de representação de documento. A rede completa de documentos é representada por um conjunto de arquivos de transações produzidas durante a análise gramatical. A tarefa dirigida após da análise gramatical, é organizar a rede de uma forma que a evidência possa ser propagada através dele rapidamente e eficazmente.

O valor do nó interno da rede é uma função de valores determinados pelos seus paternos. INQUERY evita instantaneamente que a rede de documentos termine durante a recuperação, usando a inferência recursiva para determinar os valores de um nó. A velocidade da inferência recursiva depende da velocidade da informação sobre o nó e como suas ligações (*links*) podem ser obtidas. INQUERY fornece rápido acesso a esta informação armazenado-a como arquivo invertido na estrutura de dados na árvore balanceada (*B-Tree)*.

O arquivo invertido é construído mais eficientemente se a transação para o termos forem processadas juntas. Então as transações de arquivos são armazenadas antes que os arquivos invertidos sejam construídos. O processo de classificação envolve muitos passos, ambos por eficiência. As transações podem ser armazenadas em múltiplos arquivos os quais não estão todos ajustados em um só disco.

Começa-se a usar o programa de classificação no UNIX para classificar cada arquivos de transação por termos e identificador de documentos. Se todas as transações se ajustam em um simples disco, separa-se os arquivos de transações classificados. Por outro lado particiona-se os arquivos de transação classificados em termos e separa-se todas as partições que cobriam as mesmas abrangências dos termos.

A classificação é uma das muitas tarefas que consomem mais tempo na construção de uma rede de documentos. Nos teste com 1Gbyte de coleção de documentos, classificação, particionamento e a união da classificação de 1,3 Gbytes de transação, requereu 13,6 horas de CPU numa Sun SPARC-Server 490.

Depois que as transações são classificadas, os arquivos invertidos podem ser construídos em *O(n)* tempo. As chaves para os arquivos invertidos são termos *ids*. O registro nos arquivos invertido armazena a freqüência da coleção dos termos, o número de documentos na qual o termo ocorre, e a transação na qual o termos ocorre. O arquivo invertido é armazenado em um formato binário, que o torna menor que as transações dos arquivos do qual é montado. Os 1,3 Gbytes de transações mencionadas acima foram convertidos em 880 Mbyte de arquivos invertidos, em 2,5 horas de CPU.

4.1.12 O sub-sistema de recuperação

O sub-sistema de recuperação converte os textos da *query* em redes de *query*, e então avalia a rede de *query* e o contexto da rede de documento previamente construída.

4.1.13 Construindo uma rede query

Queries podem ser feitos para o INQUERY usando também linguagem natural ou uma linguagem de *query* estruturado. *Queries* de linguagem natural são convertidos em uma linguagem estruturada de *query* aplicando o operador `#sum` para os termos da *query*. Na figura a seguir, tem-se a descrição #sum e outros operadores na linguagem query do INQUERY.

Figura 17: Os operadores na linguagem Query do INQUERY

Operação	Ação
#and	AND - termo de junção no escopo do operador.
#or	OR - termos condicional no escopo do operador.
#not	NEGATE - termo de negação no escopo do operador.
#sum	Média da confiança dos argumentos.
#wsum	Soma dos pesos da confiança dos argumentos, escalado pela soma dos pesos.
#max	É a confiança máxima nos argumentos.
#n	É o número máximo de separações adjacentes em argumentos. Por exemplo: #3(AB) adapta-se com "A B", "A *c* B" e "A c c B".
#phrase	É o valor da função de confiança retornado pelos operadores #n e #sum.
#syn	O argumento dos termos podem ser considerados sinônimos.

Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

Os operadores *query* permitem ao usuário prover informação estruturada na *query*, incluindo frases e necessidades aproximadas. Textos da *query* são convertidos em minúsculas, possivelmente enquadrando *stopwords* ou *stemming* para conversão das palavras em forma canônica, e as compara no conceito do dicionário, antes de ser convertido em uma rede *query*.

As *stopwords* podem ser chamadas também de *remoção de palavras de ligação*. Este procedimento consistem na remoção de palavras com freqüência elevada, ou seja, palavras com baixo valor discriminativo para fins de pesquisa. Palavras que

ocorrem em mais de 80% dos documentos, na maioria dos casos, são considerados pouco úteis. Geralmente são artigos, preposições, etc., (poucas vezes verbos, advérbios e adjetivos) Por exemplo “se”, “a”, “que”, “desde”, “te”.

Os *stemming* podem ser chamados também de radicalização ou lemmatizar. Este é um processo de fuzão, que consistem em igualar o termo da pergunta ou do documento, por exemplo, “florestas”a “floresta”, “ligar” a “ligação” a “ligante” a “ligado”; remoção de sufixos, por exemplo na frase “condenação das matas da amazônias” pode ficar reduzido a “conden matas amazon”; deteção de radicais equivalentes, por exemplo, Act eAcç (Acto e Acção).

Nós da rede *query*, correspondem aos termos de *query* e aos operadores de linguagem estruturada. As informações contidas nestes nós variam, dependendo do seu tipo. A junção da rede *query* para a rede de documentos pré-existente acontece nos nós de termo da *query*.

4.1.14 Mecanismo de recuperação

O mecanismo de recuperação INQUERY, aceita o nó principal de uma rede *query* e o avalia retornando um nó simples contendo uma lista confiável. Esta lista confiável é uma estrutura contendo os documentos e suas “sugestões” correspondentes ou probabilidades da união de informações necessárias como definidas pela *query*. O mecanismo de recuperação executa seu trabalho através das listas de proximidades instantâneas, ou seja, através de uma busca imediata dentro dos nós de termos da *query*, convertendo tais listas em listas confiáveis de acordo com aquelas requisitadas através da estrutura da rede de *query*, usando métodos específicos. Esta lista pode ser escolhida para prover uma lista de documentos pré-selecionados para o usuário averiguar.

A rede de inferência é avaliada através de chamadas recursivas das rotinas de avaliação principal, que resgata um dos muitos possíveis nós de rotinas de avaliação específica. As rotinas representam a forma canônica de avaliação e simplificação de "*link matrices*" a cada nó.

A estrutura básica do qual todos nós de documento de computações reais, são derivados das listas de proximidade e listas de certeza. Uma lista de proximidade contém informações estatísticas e aproximações através de documentos de bases específicas. A lista de certeza é uma lista de documentos associados a valores de certeza de um determinado nó, como também a falta de certeza e pesos usados quando houver a combinação de listas de certeza de nós diferentes. As listas de certeza conterão a probabilidade acumulativa de uma relevância de documentos para os valores da *query* apresentados de seus paternos. Listas de certeza podem ser computados através de listas de aproximadas, mas a derivação reversa não é possível. Esta limitação obriga algumas restrições na *query*. A *query* não deve gerar a lista de proximidade, tipo resultante do nó pelo qual é atuado por uma espera rotineira de um tipo de lista de certeza. Listas de proximidade são transformados dentro dos valores de certeza usando a informação na lista, e uso combinado de funções de pesagem ou conquistas.

São calculadas as contagens dos nós de certeza como uma combinação do termo de freqüência *(tf)* e o peso da freqüência do documento inverso *(idf*). Os valores são normalizados permanecendo entre 0 e 1, e são futuramente modificados por um termo de freqüência *(tf*) e valores padrões de falta de certeza dos quais o usuário pode definir no programa solicitado.

Cálculo de uma certeza para um determinado nó, depende do tipo do nó, do número e certeza em seus paternos. As combinações probabilísticas são alcançadas através da lista de certezas unidas e negadas.

4.1.15 Análise de contexto local (LCA)

Em 1996, o Centro de Inteligência em Recuperação de Informação da Universidade de Massachussets (CIIR) apresentou uma técnica de expansão da *query*, que trabalhou um método mais confiante do que os métodos anteriores de busca de pseudorelevância, tal técnica se chama Análise de Contexto Local (LCA), que localiza os termos de expansão nas passagens mais requeridas. Para caracterizar a expansão ela utiliza frases também, e determina os pesos às características, a fim de elevar o valor esperado destas características, que regularmente acontece perto dos termos da *query*.

O LCA possui vários parâmetros que influenciam em seus resultados. O primeiro é a escolha do banco de dados do LCA: as passagens mais requeridas são extraídas da coleção. Este banco de dados poderia ser a própria coleção de teste, mas geralmente há uma outra coleção, talvez maior que se espera aumentar a expansão de termos. Os outros dois parâmetros do LCA são: o número de passagens usadas para a expansão, e o número de características das expansões adicionado a *query*. As características do LCA foram introduzidas na construção da *query*, que permitiu um peso médio das características.

4.1.16 Resultado do INQUERY

Alguns resultados foram obtidos pelos experimentos de CALLAN et al (1992), apesar dos experimentos datarem por volta de 1992, o mecanismo de

recuperação e pesagem dos termos permanecem o mesmo até hoje. Por este motivo relatare-se alguns resultados promovidos por CALLAN et at (1992) e alguns resultados promovidos pelo INQUERY na TREC-8 e TREC-9.

O INQUERY foi desenvolvido para funcionar no Sistema Operacional UNIX, estação de trabalho desenvolvido pela Digital Equipment Corporation e SUN Microsystems. A quantidade de memória e espaço necessário em disco, depende do tamanho da coleção de documentos. Para uma coleção de N bytes, INQUERY necessita de aproximadamente 5N bytes de espaço livre em disco para construir a sua base de dados. Uma vez que a base de dados é construída, o INQUERY necessita de aproximadamente, 1,5N bytes de espaço livre em disco para armazenar a base de dados de documentos (N bytes para o texto principal, 0,5N bytes para os índices).

As quantidades de memórias necessárias são mais difíceis de prever, porque eles dependem diretamente da característica da coleção de documentos, e da complexibilidade das *queries*. Para uma estação de trabalho UNIX, uma estimativa superficial é que o INQUERY requer certa de N/15 bytes de memória virtual. Uma quantidade razoável de memória virtual seria de N/60 bytes. Desta forma, em uma coleção de 2 gigabytes, seria necessário no mínimo de aproximadamente 135 Mb de memória virtual e 32 Mb de memória física. Embora a necessidade de memória e espaço em disco do INQUERY não seja tão imprevisível, quando comparado com muitos sistemas de recuperação de informação, ele é bem previsível.

Tem ocorrido estudos e experimento usando uma memória interna para indexação de documentos que, significantemente tem reduzido o espaço em disco necessário durante a criação dos índices. As vantagens desta aproximação, são sua simplicidade para o usuário, e a redução do espaço em disco de 5N bytes para 1,9N

bytes. A desvantagem do uso permanente de disco é o aumento de 1,5N bytes para 1,9N bytes.

O sistema INQUERY constrói coleções de documentos automaticamente de aproximadamente 40-50 megabytes por hora de CPU em um SUN SPARCserver com 128 Mb. A velocidade varia de acordo com o tamanho da coleção de documentos, porque o processo de classificação leva um tempo proporcional à (*n log n)*.

No mesmo sistema UNIX, recuperação de documentos leva em média de aproximadamente 1 segundo de CPU por termo de *query* em 1 Gb de coleção de documentos. O tempo varia largamente, dependendo da freqüência do termo na coleção e o tipo da linguagem dos operadores *query*.

Tempo de processo de uma *query* típica é de 3 a 60 segundos em 1 Gbyte de coleção de documentos. Tempo de processamento varia de acordo com a complexibilidade da query e o número de termos na sua coleção. Termos com alta freqüência na coleção são provavelmente adicionados ao tempo de processo, e ao comprimento das listas de proximidades associadas. Performance da recuperação é bem melhor do que as recuperações booleanas e probabilísticas convencionais.

Alguns procedimentos de *query ad- hoc* estão sendo desenvolvidos, mas em caráter secundário, e alguns resultados na TREC-8 (1999) foram obtidos. Os resultados ainda são experimentais, pode-se verificar alguns dos resultados na tabela abaixo.

Tabela 05 : Resultados do INQUERY em tarefas ad-hoc

Processo	Tipo da query	Precisão
INQ601	Apenas título	0,2325
INQ602	Apenas descrições	0,2492
INQ603	Título e descrições	0,2659
INQ604	Títulos, descrições e narrativas	0,2809

Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

Na TREC-8 para uma *Small Web Track* (rede de baixa transmissão) de 2 gigabytes, os resultados foram razoáveis. Dos 50 tópicos, 42 foram melhores do que a média, e apenas 7 tiveram baixo desempenho. Três tiveram resultados máximos para a sua respectiva *query*. Ao processar o INQ620, foram usadas alguns conjuntos de *query* do *ad-hoc* INQ603, isto é, títulos e descrições do tópico com o mesmo processo de *query*. A média de precisão na busca foi de 0,3327, precisão em 10 documentos foi de 0,5040 e em 20 documentos 0,4143. Mostrando assim um resultado satisfatório para buscas *ad-hoc*.

Em uma *Large Web Track* (rede de grande transmissão), no processamento do INQ650, a média da precisão modificou-se para 0,3927. A precisão na recuperação de 10 documentos foi de 0,4690 e de 20 documentos 0,5000.

Em nossos testes de Perguntas e Respostas (Q&A), foi utilizado o melhor operador de passagem do INQUERY, para localizar as respostas para as *queries* nos *5* melhores documentos recuperados. O processo adotado foi de contar as palavras ao invés dos *bytes*. Foi determinado que a melhor passagem para 50 bytes era de 10 palavras e para 250 bytes 50 palavras. Processa-se então a resposta, e coloca-se em um tamanho correto. Se uma palavra fosse interrompida pelos limites de 50 a 250 bytes, não a incluímos em uma resposta parcial.

O CIIR submeteu quatro processos, dois com limite de 50 bytes e 2 com limite de 250 bytes. A diferença entre estes processos é que em alguns casos foram usadas as expansões LCA para as *queries*, e em outros não. Os resultados dos experimentos mostraram que a expansão LCA melhorou o desempenho. Pode-se ver alguns resultados, conforme tabela a seguir.

Tabela 06: Resultados do INQUERY com expansão LCA

Processo	Índice Alcançado no INQUERY (LCA)	Índice Geral Médio	Diferença %
INQ634 – 50 bytes, com expansão	0,18737	0,11970	+57,1%
INQ635 – 250 bytes, com expansão	0,37828	0,28081	+34,7%
INQ638 – 50 bytes, sem expansão	0,12556	0,12485	+0,5%
INQ639 – 250 bytes, sem expansão	0,33283	0,28081	+18,5%

Fonte: CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência.

Na TREC-9 foi determinado que na configuração do LCA, fosse feito experimentos tentando de 10 a 50 passagens, e foi selecionado de 10 a 50 conceitos. Os resultados mostraram que em 50 passagens e 50 conceitos de expansão, teve-se uma melhora no desempenho. A média de precisão foi de aproximadamente 4% maior do que usado na TREC-8, que era de 30 passagens e 50 termos de expansão.

4.2 Sistema de Recuperação THISL

4.2.1 Introdução

THISL é um Sistema de Recuperação de Informação por voz da Inglaterra e da Rede de Notícia Norte Americano (Nort American Broadcast News). O sistema é baseado no ABBOT, identificador de voz de grande vocabulário usando uma rede repetitiva acústica, e um sistema de recuperação de texto THISLIR. O sistema da Rede de Notícia Britânica é efetuado em tempo real, e existe uma integração com um sistema de recuperação de documento por voz.

O Sistema THISL de recuperação por voz de documentos é baseado no Sistema ABBOT de reconhecimento de fala de grandes vocabulários e no sistema probabilístico de recuperação de textos. O ABBOT é usado para transcrever o áudio da rede, transformando o problema em uma recuperação de texto.

O sistema ABBOT LVCSR é desenvolvido atualmente pela Universidade de Cambridge e Sheffield e mais a diante desenvolvido pela SoftSound. Este sistema difere dos demais, pois possui um modelo acústico baseado em redes de conexões. No ABBOT são usadas duas redes de treinamentos: *forward-in-time* e o *backward-in-time* (PLP front-end). Nas aplicações são usadas 64Kb de dicionário de pronúncia de palavras, junto com um modelo de idioma.

O ABBOT possui várias características, tornando-o satisfatório para aplicações de recuperação de documento por voz, incluído a performance do *realtime* (em tempo real), decodificadores com baixa latência (tempo de carregar informações da memória secundária, para a memória principal) e uma arquitetura simples. Particularmente o sistema tem evoluído em tarefas de notícias de radiofusão, usando apenas contextos independentes em modelos acústicos.

O sistema de pesquisa LVCSR possui um espaço de pesquisa muito grande. No ABBOT, é adotada uma pilha decodificadora de procuras estratégicas, envolvendo dois decodificadores: NOWAY, desenvolvido por Steve Renals e Mike Hochberge, e o CHRONOS, desenvolvido por Tony Robinson e James Christie. Os algoritmos adotados são capazes de fazer uso direto das estimativas da probabilidade produzidas pelo modelo acústico de redes neurais, interrompendo todas as ligações que possuem uma probabilidade posterior abaixo da média. Os objetivos dos decodificadores são alcançar:

Reconhecimento em tempo real: Usando um Pentium II 450Mhz rodando o sistema operacional UNIX, calculando a média da decodificação em tempo real usando uma memória de 256Mb. É importante para esta tarefa um sistema final de aproximadamente 1000 horas de áudio.

Decodificação de todo o resultado: A eficiente memória usada no CHRONOS, permite a decodificação de resultados longos, e então capacita o uso da acústica on-line como uma alternativa para técnicas de normalização mais comuns.

Medidas de confiança: Na decodificação de áudio-contínuo a taxa de erro varia de acordo com o seu tamanho. Temos uma média de confiança que permite a pesagem dos termos de freqüência na recuperação do texto.

Decodificação de sentenças cruzadas: Normalmente o nosso modelo de linguagem contém um símbolo especial <s> para indicar uma sentença limite. Dando a este símbolo uma percepção acústica um pequeno período de silêncio que permite ao decodificador a hipótese de limite.

A tabela a seguir nos mostra a taxa de erros de palavras do sistema analisadas em duas novas estações de rádio, a *BBC Nine O'Clock news* em 8 de maio de 1998, e a *BBC One O'Clock news* em 9 de fevereiro de 1999. O sistema padrão foi estabelecido para funcionar em tempo real. Em seu modelo é utilizada uma normalização acústica on-line ao invés da segmentação, usando o limite de sentença e aplicando o estudo baseado na confiança dos modelos acústicos. O sistema em tempo real mostra que foi cometido 2% a mais de erro com relação à velocidade, do que previa-se. Selecionado os dados usados, e usando uma linguagem mais apropriada, a performance da decodificação melhorou a taxa de erro suavemente. Pode-se verificar na tabela abaixo que a taxa de erro, em todos os casos, se manteve relativamente constante.

Tabela 07: Taxas de erro de palavra no sistema de radiofusão em inglês britânico

Sistema	Taxa de Erros de Palavras
Sistema Padrão	36,6%
4x o tempo real	35,9%
Todos os dados do Treinamento	37,1%
Com a radiodifusão Norte Americana	37,4%
Sem sentenças cruzadas.	36,9%

Fonte: ABBERLEY, D.; et al. Retrieval of broadcast news documents with the THISL system. In:. IEEE INTERNATIONAL CONFERENCE ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998, Seattle.

Segundo Abberley et al (1999), o sistema atual tem algumas diferenças em relação ao sistema usado na TREC-6 SDR (The 1997 THISL Spoken Document retrieval System in Sixth Text Retrieval Conference). Foi trocado o sistema PRISE de recuperação de texto por um sistema probabilístico padrão localmente implementado. O localizador de texto foi abolido para lidar com nosso vocabulário extra, a partir desta experiência descobriu-se que isto não é um sério problema, pois o recuperador por voz utiliza um vocabulário de cerca de 60.000 palavras (nesta avaliação usa-se o recuperador por voz respeitando também o vocabulário extra).

Passa-se a discutir o Sistems THISL de Recuperação de Informação, segundo (ABBERLEY et al., 1999).

4.2.2 O sistema THISLIR de recuperação de texto.

Anteriormente o THISL usava o sistema PRISE de reconhecimento de texto desenvolvido pela NIST, atualmente usa um sistema localmente implementado. O sistema THILS usa um modelo probabilístico padrão com listas curtas de paradas com 132 palavras (com uma lista de parada adicional com 78 palavras quando a *query* é processada).

Isto o torna essencialmente um "Sistema de Recuperação de Texto", usando uma lista de parada (remoção de palavras de ligação – *stopwords*), o algoritmo Porter e a função do termo de pesagem OKAPI. O algoritmo de Porter é um processo de *stemming* (reduzir todas as palavras com o mesmo radical a um índice simples), que remove os sufixos mais longos, usando uma lista de determinações. Este processo é adequado para a análise de tabelas ou sílabas. Especificamente é usado a função de

pesagem de termo $CW(t,d)$ para o termo t e o documento d conforme descrever-se-á mais a diante.

4.2.3 Pesos

A idéia atrás dos termos de pesagem é a seletividade. O que faz um bom termo, é o poder da escolha de quaisquer dos poucos documentos relevantes dos muitos não relevantes. Nem todos os termos são igualmente úteis para representar o conteúdo dos documentos. Geralmente termos menos freqüentes permitem identificar um conjunto mais restrito de documentos. A análise dos termos de pesagem, será descrita a seguir.

4.2.4 Frequência da coleção

Termos que ocorrem em apenas poucos documentos são freqüentemente mais valiosos que aqueles que ocorrem em muitos. Pesos de freqüência na coleção (também conhecido como documento inverso de pesos de freqüências) são definidos como a seguir, para o termo t(i):

dado:

n=o número de documentos contendo o termo t(i);

N=o número de documentos na coleção.

A coleção de pesos de freqüência, CFW(i), para o termo i é então:

$$CFW(i) = \log N - \log n. \qquad (18)$$

Na implementação, coleção de termos de freqüência é simplesmente contas anexadas ao arquivo invertido, e pesos são computados para os termos necessários solicitados. O logaritmo pode ser levado a uma base conveniente. Uma implementação

usa a base dois, assegurando que todos os pesos possam ser bem aproximados por inteiros entre –32K e +(32K).

4.2.5 Termos de freqüência

O segundo princípio da pesagem dos termos é a freqüência de termos nos documentos internos. A maior freqüência dos termos ocorrerem dentro dos documentos, o mais provável é ser importante para aquele documento. Assim enquanto a coleção dos termos de freqüência é a mesma para qualquer documento, a sua freqüência nos documentos varia. O termo de freqüência para o termo *t(i)* no documento *d(j)* é:

TF(i,j) = o número de ocorrências do termo t(i) no documento d(j).

Na prática, estas contagens são registradas juntas com a identificação dos documentos nos arquivos invertidos. Termos de freqüência não devem ser usados da mesma proporção que se eleva um fator de pesagem, mas deve ser relacionado a uma origem permanente de informações sobre documentos.

4.2.6 Comprimento do documento

A terceira contribuição da pesagem é o comprimento de um documento. Temos o comprimento documento *d(j)* da seguinte forma:

DL(j) = o total de termos ocorrentes no documento d(j)

O uso do tamanho documento descrito abaixo de fato normaliza a medida do comprimento de um documento *j* é:

$$NDL(j) = \frac{DL(j)}{\text{Média de DL para todos os Documentos}} \qquad (19)$$

Isto tem a vantagem de que as unidades na qual *DL* é contado, não importe muito. Uma medida muito simples para que o número de caracteres de *d(j)* possa ser bastante adequado para o número de ocorrência do termo.

4.2.7 Combinando a evidência

Os valores para cada termo, necessitam ser combinados juntos com os outros termos do pedido. Existem várias fórmulas para esta combinação, uma tem provado ser efetiva durante o programa TREC. Onde *TF(i,j)* é a freqüência de um termo *t(i)* em um documento *d(j)*, o Peso Combinado (CW) é:

$$CW(i,j) = \frac{CFW(i) * TF(i,j) * (Ki+1)}{Ki * ((1 - b) + (b * (NDL(j))) + TF(i, j)} \qquad (20)$$

Onde *(Ki e b)* são constantes direcionadas. A fórmula assegura que o efeito do termo da freqüência não é muito forte (dobrando TF não dobra o peso), e para o termo que acontece uma vez em um documento de comprimento médio, o peso é justamente CFW.

A contagem total de um documento *d(j)* é simplesmente a soma dos pesos e dos termos da *query* presentes no documento. Documentos são ordenados em ordem descendente em suas classificações, para a apresentação de seus usuários.

O direcionamento da constante *Ki* modifica a extensão da influência dos termos de freqüência. Idealmente, deveria ser fixo depois de tentativas sistemáticas em uma coleção particular de documentos. Nos testes na TREC, com umas coleções mais heterogêneas de novos textos completos, alguns *abstracts*, e longos relatórios governamentais, o valor de *Ki=2* foi achado ser um valor seguro para iniciar. A constante *(b)*, o qual varia entre (0 e 1), modifica o efeito do tamanho do documento. Se *b=1* a suposição é aquele documento serem simplesmente longos porque eles são

repetitivos, enquanto se *b=0* a suposição é que eles são longos porque eles são de múltiplos tópicos. Assim fixando *b=0,75*, reduzirá o efeito do termo de freqüência, que é principalmente atribuível a repetição de termos. *Se b=0* não há nenhum efeito de ajuste de comprimento, quanto maior o comprimento maior será a repetição dos tópicos, entretanto esta situação não é predominantemente atribuída a verbosidade (repetição de termos). Encontra-se na TREC o estabelecimento de *b=0,75*, resolve assim o problema.

O uso apenas de uma coleção de pesos de freqüência, é apropriado para a recuperação de documentos onde apenas títulos ou textos curtos estão disponíveis para procura, os outros componentes de pesagens são pertinentes quando procura-se textos longos. Entretanto mesmo que esta coleção de freqüência de pesagem possa ser usada para obter uma melhoria, seria possível que outras informações não estejam disponível.

4.2.8 Procura interativa

Este é um desenvolvimento natural para a procura, usando pesos de coleção de freqüência como definido acima. Uma procura inicial é usada para obter alguns documentos que serão avaliados a sua relevância para a *query*, marcando assim os relevantes dos não relevantes. As informações assim obtidas podem ser usadas justamente também para um novo peso inicial de termos de procura, ou para modificar a *query* original adicionando um novo termo. A modificação dos pedidos dos termos únicos de pesagem, é freqüentemente chamado de relevância de pesagem, modificando-se o pedido de composição, tipicamente adicionando mais termos, é usualmente chamado de *query* de expansão. Em um processo, os termos originais são pesados novamente.

4.2.9 Relevância dos pesos

A relevância básica da pesagem é simplesmente a relação entre a relevância e a não relevância dos documentos distribuídos para a procura de termos modulados pela sua coleção de freqüência, desde que o termo pudesse estar em documentos relevantes. Relevância dos pesos são então estimativas, com o acesso de *(0,5seg)* como na fórmula abaixo.

Dado, o termo *t(i),*

r= o número de termos de documentos relevantes conhecidos contendo t(i)

R= o número de documentos relevantes conhecidos para o pedido

A Relevância dos Pesos é:

$$RW(i) = \log \frac{(r+0{,}5)(N-n-R+r+0{,}5)}{(n-r+0{,}5)(R-r+0{,}5)} \qquad (21)$$

Esta fórmula pode ser usada ao invés da CFW (fórmula 1) para todos os termos usados na segunda ou subseqüente interação. (Pode ser na realidade também para a primeira interação fixando *r=R=0*; o resultado da fórmula é uma aproximação muito próxima à freqüência de coleção de pesos dados na fórmula 1). A base do logaritmo pode ser igual como para CFW. A fórmula acima, referente à relevância dos termos, é a mesma usada no sistema OKAPI de IR.

4.2.10 Expansão da query

Diferentes combinações dos usuários de sistemas e esforço de sistemas podem ser usadas para selecionar novos termos. Todos termos dos documentos relevantes direcionados são ordenados de acordo com a Oferta de Pesos (OW):

$$OW(i) = r * RW(i) \qquad (22)$$

Então o topo dos 10 ou 20 termos ordenados são incluídos na procura, onde 10 ou 20 termos é uma probabilidade razoavelmente segura. Na maioria dos casos poderá ser incluído algum termo desnecessário, mas o efeito destes poderão ser superados pelos bons termos.

A seleção do procedimento, incluindo o poder do usuário, mostra seus termos em ordem de peso de oferta, atraindo ou rejeitando a seleção. Alternativamente ou adicionalmente , o usuário pode ser convidado/permitido a selecionar seus termos de qualquer texto exibido durante a procura.

Todos os novos termos podem ser pesados para a procura, usando pesos de relevância. A oferta de pesos é para a decisão da seleção.

As relevâncias dos pesos podem ser substituídas pela coleção de freqüência de peso na combinação de pesos (fórmula 20), dado a Combinação Interativa de Pesos (CIW):

$$CIW(i,j) = (RW(i) * TF(i,j) * (K1+i) / (Ki * ((i-b)+(b*(NDL(j))))+TF(i,j)) \quad (23)$$

Especificamente, o peso da expansão da *query QEW(Q,e)*, para a expansão do termo poctencial *(e)* e a *query Q*, através do conjunto *(nr)* de documentos pseudo relevantes é definido como:

$$QEW(Q,e) = CFW(e)\sum_{i\in Q} CFW(t)\sum_{i=1}^{nr} TF(e,di)TF(t,di) \quad (24)$$

Onde *QEW(Q,e)* é usado para ordenar a expansão dos termos, e o topo *(nt)* são escolhidos para a expansão *Q*.

4.2.11 Parâmetro da pesagem de termo

Investigando o efeito da variação dos parâmetros *b e K* na pesagem de termos na fórmula (5), pode-se notar que o desenvolvimento das *queries* alcança seu ponto máximo quando *K=0,25*, o qual corresponde uma diminuição na significação de *TF* comparada com *CFW*. Isto não está presente nas avaliações das *queries*. Há um outro máximo por volta de *(b;K)=(0,5; 1,0)*, para ambos os conjuntos de *queries*, os quais eram os parâmetros estabelecidos para todas as execuções submetidas.

4.2.12 Listas de parada

Conduz-se o experimento usando listas de parada construída manualmente, incluindo uma lista de parada de 23 palavras; lista de parada de 319 palavras usadas pela Universidade de Glasgow; listas de parada de 429 palavras usado na obra de Informação de Recuperação de Dados através de análises Léxicas; listas de Parada da Editora Prientice Hall; e uma lista de parada de 379 palavras desenvolvidas localmente com palavras extras adicionadas seguindo os padrões da análise anterior da TREC *queries*.

Adquirindo experiências de controle, usa-se uma listas de parada adequando as palavras mais freqüentes representada por (*n*)**,** e também uma lista de palavras de não-interrupção. Nos experimentos, foram constatados que nas listas manualmente construída a performance foi um pouco melhor do que a lista similarmente padronizada.

4.2.13 Resultados

Na TREC-9 SDR foram realizados dois experimentos: uma em condição de parâmetro de limite desconhecido (SU), e outra em condição de parâmetro de limite

conhecido (SK). Tais experimentos foram confrontados. As mesmas transcrições foram usadas em cada caso. Contudo diferentes parâmetros de recuperação de texto foram usados para as condições SU e SK, os parâmetros não eram dependentes das *queries* (reduzido ou abreviado).

As transcrições foram produzidas pelos nossos próprios sistemas reconhecedores (S1 e S2), usa-se também as seguintes transcrições:

1) Referência de transcrições preparadas para o fechamento dos capítulos (R1);
2) Base do reconhecedor de idiomas preparado pela NIST (National Institute of Standards and Technology) - (B1);
3) Reconhecedor de idiomas preparado pela LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de Ingénieur) - (LIMSI1 e LIMSI2);
4) Reconhecedor da transcrição do idioma preparado pela Universidade de Cambridge (CUHTK).
5) Número de documentos relevantes em um tópico (R).

Os resultados da condição SU na TREC-9 estão presentes na tabela a seguir. Em todos os casos, a precisão média foi de 20-25% abaixo que na TREC-8. Isto significa que na TREC-9 as *queries* devem ter sido mais difíceis em algumas situações, ou o sistema foi muito direcionado para as *queries* da TREC-8. Em segundo lugar, vê-se que a performance das *queries* abreviadas é similar as *queries* reduzidas. Note que na TREC-8, otimizou-se o sistema, usando *queries* reduzidas.

Tabela 08: Limite do registro, condição(SU) desconhecida da evolução da TREC-9 SDR.

Transcrições		Query Reduzido		Query Abreviado	
ID	**Taxa de Erro de Palavras**	**Precisão Média**	**Precisão em R**	**Precisão Média**	**Precisão em R**
R1	10,3	0,409	0,419	0,418	0,425
S1	32,0	0,392	0,399	0,392	0,396
S2	29,2	0,399	0,410	0,393	0,401
B1	26,7	0,287	0,401	0,384	0,398
CUHTK	20,5	0,373	0,388	0,373	0,387
LIMSI1	21,5	0,377	0,405	0,386	0,391
LIMSI2	21,2	0,395	0,407	0,397	0,421

Fonte: RENALS, S.; ABBERLEY, D. The THISL SDR system at TREC-9. In: NIST SPECIAL PUBLICATION XXX-XXX: Text Retrieval Conference, 9., 2000, Washington, DC.

Tabela 09: Limite do registro, condição(SK) conhecida da evolução da TREC-9 SDR.

Transcrições		Query Reduzido		Query Abreviado	
ID	**Taxa de Erro de Palavras**	**Precisão Média**	**Precisão em R**	**Precisão Média**	**Precisão em R**
R1	10,3	0,509	0,489	0,492	0,477
S1	32,0	0,464	0,441	0,475	0,463
S2	29,2	0,465	0,435	0,478	0,463
B1	27,7	0,462	0,447	0,469	0,451

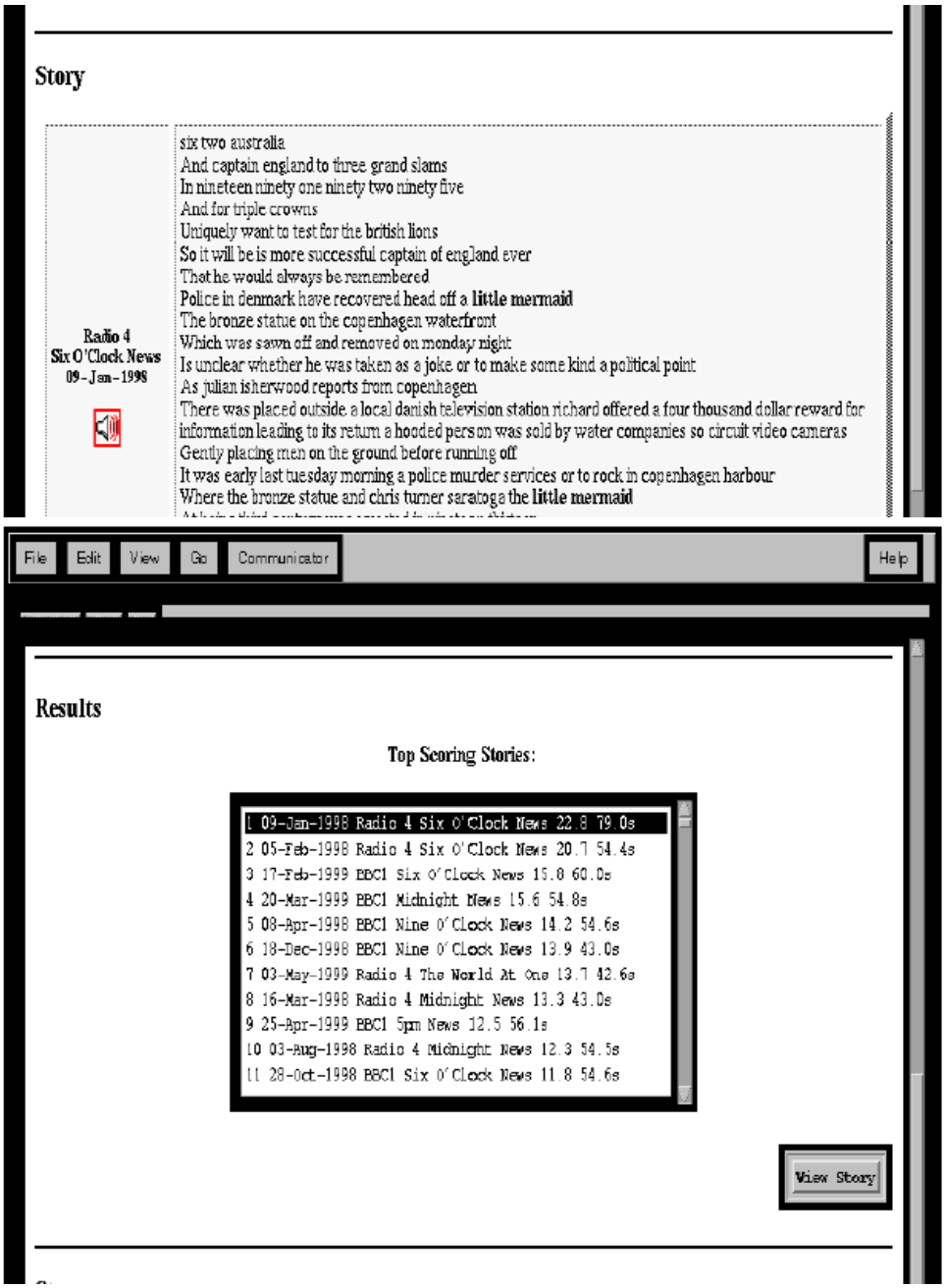
Fonte: RENALS, S.; ABBERLEY, D. The THISL SDR system at TREC-9. In: NIST SPECIAL PUBLICATION XXX-XXX: Text Retrieval Conference, 9., 2000, Washington, DC.

Finalmente, seguindo as tendências da evolução, a ligação entre a média das palavras erradas e a precisão da recuperação de texto é muito fraco. Realmente, fora de toda transcrição de reconhecimento de idiomas, a alta precisão das *queries* reduzidas é arquivada usando S2 (com taxa de erro de palavras de aproximadamente 29%).

Para contraste, resultados do caso SK na TREC-9 SDR estão presentes na tabela acima. Estes resultados seguem a mesma forma dos resultados do SU, indicando que a baixa média de precisão não é devido à segmentação/união de procedimentos. O relativo intervalo da precisão entre SK e SU está entre 10-20%.

Na figura a seguir pode-se ver um exemplo do sistema THISL respondendo à seguinte *query:* "What happened to the little mermaid?" – "O que aconteceu com a pequena sereia?", mostrando uma lista de resultados com suas respectivas estórias.

Figura 18: Exemplo do sistema THISL de Recuperação de Informação



Fonte: RENALS, S.; ABBERLEY, D. The THISL SDR system at TREC-9. In: NIST SPECIAL PUBLICATION XXX-XXX: Text Retrieval Conference, 9., 2000, Washington, DC.

4.3 Sistema de Recuperação de Texto OKAPI

4.3.1 Introdução

O sistema de busca Okapi usado pela Text Retrieval Conference (TREC) é descendente do sistema Okapi desenvolvido pelo Centro a Politécnico de Londres, agora Universidade de Westminster, debaixo de várias concessões da British Library Research & Development Departament. Sistemas anteriores da Okapi foram sistemas experimentais de recuperação do tipo probabilístico altamente interativo, alguns dos quais caracterizou uma expansão automática de *query*. Okapi é simples e robusto, baseado no modelo probabilístico generalizado com facilidades para uma busca completa, mas também de grande abrangência para operações Booleanas determinísticas e operações semi-Booleanas.

Todo o trabalho da Okapi em relação aos TRECs 1-6 foi desenvolvido pelo Departamento de Ciência da Informação da Cidade Universitária, em Londres. Maior parte da produção da Okapi na TREC-7 foi desenvolvido pela Microsoft Research, Cambridge (UK).

Em relação ao TREC-1, o baixo desempenho das funções de procura foram generalizados e divididos em uma biblioteca externa separada, a Okapi Basic Search System (BSS). Usuários da interface ou processamento de documentos em lote acessando a BSS usando um simples comando de linguagem como protocolo. Entretanto, os nossos resultadas do TREC-1 foram muito ruins, devido ao modelo clássico de pesagem feito por Robertson/Spark Jones no qual os sistemas Okapi não

costumavam determinar nenhum tamanho de documento externo ou interno nos termos de freqüência.

Durante a TREC-2 e TREC-3 um considerável número de novas funções em termos de pesagem e combinações foram experimentadas; a determinação do tempo de execução do processo e procura de pacotes foram adicionadas ao BSS; e métodos de seleção de termos aceitáveis para a busca de queries foram desenvolvidos. Durante a TREC-2 um trabalho de expansão de *queries*, sem qualquer desenvolvimento de pesquisa, foi experimentada pela primeira vez nos experimentos automáticos do *ad-hoc*, entretanto não foi utilizado oficialmente até a TREC-3.

Nossas rotinas automáticas no TREC-3 e resultados no *ad-hoc* foram bons. Na TREC-4 não foi obtido maiores desenvolvimentos. Métodos de seleção de termos de busca foram melhorados.

Na TREC-5 muitos pesquisadores usaram a expansão não oficial do *ad-hoc*, vários deles com mais sucesso que a OKAPI. Na busca, tentou-se otimizar os pesos dos termos após a seleção dos termos aceitáveis; nossos resultados eram novamente os melhores, como foi na filtragem dos lotes.

Na TREC-6 continua-se a investigação da expansão não oficial, com resultados variados. Foi introduzido também uma nova função de pesagem designada a fazer o uso de documentos conhecidos ou assumidamente não ser não relevante. Na busca e filtragem, tentou-se estender o processo de otimização, incluindo uma versão de simulação mais concreta. Novamente nossos resultados de busca e filtragem estavam entre os melhores. A Okapi BSS foi melhorada para controlar grandes bancos de dados para o VLC-Track (expansão do a*d-hoc).* Introduziu-se uma filtragem adaptável, com

resultados muito bons, usados primeiramente no TREC-7. Busca e filtragens em lote foram abandonadas.

Neste trabalho enfocar-se-á, segundo Robertson e Walker (1999), o projeto e a implementação do Sistema OKAPI, seu funcionamento e resultados.

4.3.2 O sistema Okapi

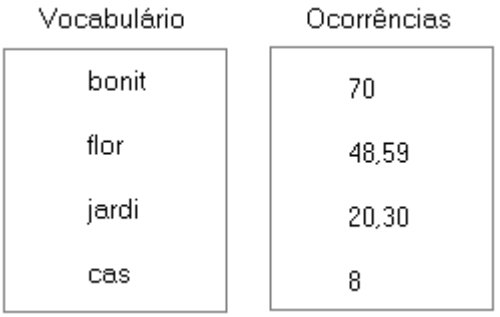
No laboratório da Microsoft Research em Cambridge, está se desenvolvendo um sistema de avaliação, para experimentos de recuperação de informação em grande quantidade. Este sistema é chamado de Keenbow. A Okapi BSS é agora visto como um componente da Keenbow.

4.3.3 O sistema básico de procura da Okapi

O sistema básico de procura (BSS), que tem sido usado em todos os experimentos da Okapi na TREC, é um sistema de saída posicionado de forma orientado, designado primeiramente para modelo probabilístico de recuperação de informação textual, usando índices invertidos. A estrutura do índice invertido consiste nos vocabulários, que é o conjunto de todos os termos distintos no texto, e as ocorrências, que são as listas contendo toda a informação para cada termo do vocabulário. Tem-se abaixo, um exemplo de um índice invertido, considerando a posição e não a pesagem dos termos:

```
1       8    13   17 20        30      37  41      48     56 59    66  70
Aquela casa tem um jardim. O jardim tem muitas flores. As flores são bonitas
```

Segundo a frase acima o seu índice invertido ficaria da seguinte forma:



O algoritmo de pesquisa do índice invertido se processa em três fases:

1) Pesquisa do vocabulário, onde os termos presentes na pergunta são pesquisados no vocabulário;
2) Retorno das ocorrências, onde as listas das ocorrências de todos os termos encontradas são devolvidas;
3) Manipulação das ocorrências, onde as ocorrências são processadas por forma a responder às perguntas.

Existe uma família de funções de pesagem que estão descritas na fórmula abaixo. Índices são justamente um tipo convencional de inversão.

$$\sum_{T \in Q} W(1) \frac{(k1+1)tf}{k+tf} \frac{(k3+1)qtf}{k3+qtf} \qquad (25)$$

Onde *Q* é a *query* contendo o termo *T*, e *W(1)* é o peso proposto por Robertson/Spark do termo *T* em *Q*, onde *W(1)* é igual a:

$$\log \frac{(r+0{,}5)/(R-r+0{,}5)}{(n-r+0{,}5)/(N-n+r+0{,}5)} \qquad (26)$$

(N) é o número de itens (documentos) na coleção;

(n) é o número de documentos contendo o termo;

(R) é o número de documentos declarados ser relevante para um tópico específico;

(r) é o número de documentos relevantes contendo o termo

(K) é k1((1-b) + b*dl/avdl);

(k1, b e k3) são parâmetros que dependem da natureza das *queries*, e possivelmente do banco de dados; k1 e b a princípio valem 1,2 e 0,75 respectivamente, mas valores menores que b são algumas vezes vantajosas; em *queries* muito longas k3 é freqüentemente é fixado entre 7 a 1000 (efetivamente infinito);

(tf) é a freqüência da ocorrência do termo dentro de um documento específico;

(qtf) é a freqüência do termo dentro do tópico, do qual Q foi derivado;

(dl and avdl) são respectivamente os comprimentos médios do documento medidos por uma unidade satisfatória.

O TSV (Term Ranking for Selection), é fornecido pela fórmula *TSV=r*W(1)*.

4.3.4 Busca e Determinação da passagem

A partir da TREC-3 a BSS tem tido facilidades para a identificação do tempo de busca e de pesagem de qualquer sub-documento contendo um número integral de parágrafos consecutivos. Desta forma a busca da passagem quase sempre aumenta a média de precisão entre 2% a 10%, bem como a recuperação e a precisão em atalhos maiores. Freqüentemente, talvez surpreendentemente, reduz-se a precisão nos atalhos menores.

4.3.5 Hardware

Todo o processamento da TREC-8 foi realizado pela Microsoft Research, Cambridge. As maiorias dos trabalhos eram realizados em uma Sun Ultra 10 300 MHz com 256 MB de memória RAM, e um Dell com dois processadores Pentium 400 MHz com 512 MB de memória RAM (testes realizados em 1999). Ambas as máquinas rodavam o Solaris 2.6. Principalmente para suprir a localização no banco, com um armazenamento em disco de 170 GB. As maiorias dos experimentos eram efetuadas na Sun. A rede ethernet era de 100Mbits.

Ad-hoc representa uma busca única obtendo assim uma lista de resultados, deve-se levar em consideração que esta lista de resultados poderá apresentar 1% ou menos de probabilidade nula.

> *A recuperação ad-hoc ou adhoc, possui na terminologia TREC, uma busca única (o usuário entra no sistema e requisita a ele uma query e recebe uma lista contendo os resultados). Esta é uma das duas tarefas centrais desde o início da TREC, e permaneceu até que a TREC abandonou a idéia "de tarefas principais" (isto todo participante deveria fazer). Todavia o objetivo da rede foi acertado como uma tarefa ad-hoc, no ano passado, e alguns dos outros sistemas são similares. Ou seja ad-hoc é uma busca liberada. A busca ad-hoc faz parte da terminilogia TREC, terminologia esta na qual uma de suas tarefas principais é onde o usuário entra com uma query e recebe como resultados uma lista de informações. Entretanto estas tarefas principais foram abandonadas pela TREC, determinando que isto era o mínimo que cada sistema participante deste tratado deveria fazer, isto aconteceu no ano de 1998. Todavia os sistemas de busca*

> *da Web ficaram acertados como por exemplo "uma tarefa ad-hoc", e os outros sistemas são similares a este. (ROBERTSON; WALKER, 1999).*

4.3.6 Banco de dados e processamento de tópicos - indexação

Para a maioria dos dados da TREC, saídas de registros possuem três campos: número do documento, qualquer conteúdo que está indisponível para indexação e porções similares. Entretanto usar-se-á apenas dois campos para a nossa coleção.

Todas as indexações de textos da TREC eram do tipo de palavra chave. Algumas palavras compostas como "New York", "friendly fire", "vitamin E" eram pré-definidas, e havia a pré-indexação facilitando a elaboração de grupos com relacionamento próximos ou sinônimos como "operations research" and "operational research" ou "CIA" and "Central Intelligence Agency". O processo de derivação foi aplicado e modificado, e a lista passou a possuir cerca de 220 palavras. Na TREC-8 foi implantado uma nova facilidade em retirar algumas expressões negativas, tais como "document", "describe(s)".

4.3.7 Significância estatística de novos termos

Procuram-se novos termos, o qual são associadas fortemente com a relevância, para a melhoria da performance. Uma medida que possa satisfazer a requisição anterior, seria uma medida de significância estatística de qualquer termo associado com a relevância.

Importância das medidas estão tipicamente em uma escala que permite uma comparação com absoluta possibilidade de uma hipótese nula.(isto é, uma absoluta

probabilidade de observação é dada por uma hipótese de nenhuma associação). Este valor absoluto pode ser 5%, 1% ou 0,1%; embora as escolhas do nível são amplamente arbitrárias, qualquer escolha do valor fixo satisfaria o requerimento dado acima, quando aplicado em diferentes tópicos ou em diferentes números de documentos relevantes.

Como na maioria de nossas medidas de termos de seleção, olhou-se primeiramente para a presente ou ausência de um termo (têm-se tido pouco sucesso com métodos o qual leva a contar a freqüência do termo). Assim a relação entre o termos e a relevância é definida por uma tabela usual 2 x 2:

Tabela 10: Tabela de termos de relevância.

	Relevante	**Não Relevante**	
Termo "t"presente	*rt*	*nt-rt*	*nt*
Termo não presente	*R-rt*	*N-R-nt+rt*	*N-nt*
	R	*N-R*	*N*

Fonte: ROBERTSON, S; WALKER, S. Okapi/Keenbow at TREC-8. In: NIST SPECIAL PUBLICATION 500-246: Text REtrieval Conference, 8., 1999.

A hipótese nula é que o termos não é associado a nenhum documento com relevância. As probabilidades de hipóteses nulas, conforme os dados anteriores, podem ser aproximadas. A probabilidade da ocorrência do termo em um documento levado ao acaso é *nt/N*. Assume-se a ocorrência do termo em uma pequena proporção da coleção como um todo. Então a probabilidade da ocorrência do termo em uma porção *rt,* fora da relevância de *R,* é aproximadamente:

$$\left(\frac{nt}{N}\right)^{rt}\left(\frac{R}{rt}\right)\text{onde}\left(\frac{R}{rt}\right)=\frac{R!}{rt!(R-rt)!} \qquad (27)$$

4.3.8 O critério

Como foi relatado, o critério normal para rejeição de uma hipótese nula seria normalmente menor que 1%. Entretanto, se fosse aplicado tal critério, o da hipótese nula, poderia se obter muitas problemas (obter-seá muitos resultados não relevantes juntamente com os relevantes). A razão disto, é porque existem muitos termos para recuperação de um texto a partir do seu início. Suponha-se que o total de vocabulários no sistema seja de 100.000 termos indexados. Desta quantidade, 1000 destes termos podem exceder este critério, perfazendo o percentual de 1% ou menos de termos não relevantes, mesmo se existindo ou não a hipótese nula.

Isto sugere que deve-se adaptar o critério da rejeição da hipótese nula ao tamanho do vocabulário *(V)*. A princípio *1/V* implicaria que deve-se esperar (aceitando a hipótese nula) em cima de todos os termos. Seguramente isto representa inicialmente que muitos termos do vocabulário terão uma associação negativa com a relevância, não sendo relevantes para a nossa busca. Nos experimentos, tem-se usado uma busca inicial $1/Ve^c$, para algumas constantes *(c)*. Assim a expressão do critério é:

$$\left(\frac{nt}{N}\right)^{rt}\left(\frac{R}{rt}\right) < \frac{1}{Ve^{c}} \text{ ou } (rt)\log\frac{N}{nt} - \left(\frac{R}{rt}\right) - \log(V) > c \qquad (28)$$

Se a inicial *(c)* está estabelecido em 0, isto é equivalente à abordagem da inicial de *1/V*, isto é aceitando um termo de interferência abaixo do modelo discutido acima. Um *(c)* positivo é um início específico; um valor de *(c=4,6)* corresponde para ter menos que 1% de chance de aceitar qualquer termo de interferência. Experimentos sugeriram que pude-se dispor um valor flexível da inicia *(c)*, usou-se alguns valores negativos de *(c)*.

4.3.9 Execução do programa e processamento do banco de dados

Os programas serão processados de acordo com a forma *ok8xyz*, onde *x* é o tratamento:1- Base (baixo nível de inicialização, inicialização sem otimização); 2 - alto nível de inicialização; 3 - alto nível de inicialização + inicialização com atimização, y é a utilização das funções: linear ($LF1= 3*R^{+}-2*N^{+}$), não linear ($LF1=6*(R^{+})^{0,5}-N^{+}$); linear ($LF2= 3*R^{+}-N^{+}$), não linear ($LF2=6*(R^{+})^{0,8}-N^{+}$), onde $R^{+}/R^{-}/N^{+}/N^{-}$ é o número de documentos em cada categoria. Um parâmetro positivo pode ser analisado como o valor de cada documento, um parâmetro negativo como custo de classificar o documento na categoria, e *(z)* é o escore de normalização: 1 Linear; 2 Não-linear.

Antes da indexação, o início do texto foi reduzido removendo-se linhas iniciado com “Server:”, “Content-type:”, “Last modified:”, etc. O número do documento foi então identificado, seguido pela remoção de todo o texto dentro “<...>”. Datas e URLs foram retidas, mas não indexados. Isto reduz a indexação do texto em aproximadamente 50%, aproximadamente 50 Gb.

Na análise de alguns documentos, foram constatadas que havia muitos materiais não textuais (dados comprimidos etc). Foi decidido que não seria possível a prática da remoção destes materiais. Isto resultou em um índice com arquivos de dicionários muito grande com cerca de 70 milhões de termos que não são essenciais. Na tabela a seguir pode-se verificar a melhor performance da precisão em relação ao retorno.

Tabela 11 – Retorno e precisão dos resultados. Média de todos os anos, dos resultados das relações entre precisão e retorno, baseado em 49 tópicos.

Processo	Precisão	Retorno
Ok8f111	0,174	0,338
Ok8f112	0,160	0,343
Okf8121	0,168	0,362
Okf8122	0,151	0,375
Okf8211	0,239	0,257
Okf8212	0,227	0,274
Okf8221	0,220	0,295
Okf8222	0,203	0,326
Okf8311	0,248	0,252
Okf8312	0,238	0,265
Okf8321	0,236	0,271
Okf8322	0,226	0,289

Fonte: ROBERTSON, S; WALKER, S. Okapi/Keenbow at TREC-8. In: NIST SPECIAL PUBLICATION 500-246: Text REtrieval Conference, 8., 1999.

O banco e de dados foi reindexado sem a informação de posição para TREC-8. O resultado do tamanho do índice foi de 14 GB (comparado com 34 GB de toda indexação usada na TREC-7). O total de números de itens indexados de 18,6 milhões de documentos foi de aproximadamente de 5.800 milhões (média de 312 por documento), e o correspondente representante para os modelos foi de 2600 milhões (140 por documentos).

As execuções oficiais do sistema são *ok8v1* e *ok8v2*. Todas as execuções usam um plano não expandido de procura *ad-hoc*. A lista de parada padrão usada por *ok8v1*, contém 222 palavras. A lista de parada estendida usada por todas as outras execuções contém também “i”, “inform..”, “does”, “me”, “find”; estes são adicionados com a intenção de acelerar as procuras. Com um banco de dados normal e com uma memória de 256-512 MB de alcance, a BSS procura usualmente mais depressa, se os conjuntos de saídas são formados na memória virtual do que explicitamente no disco

físico; mas para o VCL[1] muito mais memórias físicas podem ser requeridas para esta operação. Finalmente, a BSS facilita a limitação heurística do tamanho do conjunto de saídas, tem uma marca efetiva quando é requerida a saída de poucos documentos em um grande banco de dados.

Tabela 12 – Resultado do VCL

Execução	Condições	Tempo da query (s)	Precisão média de (P)	P10[2]	P30
Ok8v1	Sem expansão e sem passagens	5,88	431	568	528
Ok8v2	Como em v1 mas listas de parada parcialmente maior.	4,30	445	560	538
Ok8v23	Como em v2 mas arquivos temporários ao invés de memória	3,82	445	560	538
Ok8v22	Como v23 mas sem limitação do tamanho do conjunto de saída	7,90	445	560	538

Fonte*:* ROBERTSON, S; WALKER, S. Okapi/Keenbow at TREC-8. In: NIST SPECIAL PUBLICATION 500-246: Text REtrieval Conference, 8., 1999.

Os métodos usados na Okapi em geral continuam dando bons resultados em vários trabalhos. Entretanto são notáveis as três modificações introduzidas no sistema: absoluta expansão dos termos de seleção, remoção de contradições, modelo não-linear para escore de calibração.

Para alcançar resultados razoáveis nas filtragens, necessita-se mover a expansão da *query*. Entretanto, a concentração na configuração inicial foi útil, e o aprimoramento em relação à performance no terceiro ano de simulação provavelmente indica que está efetuando procedimentos que outros métodos não conseguiram alcançar.

[1] VCL é o processamento no banco de dados. Neste trabalho o VCL efetua duas buscas Ad Hoc.
[2] Como o VCL track funciona com duas buscas *Ad Hoc*, P10 significa a primeira busca e P30 a segunda busca.

5 ANÁLISE DOS MODELOS E SISTEMAS

Verifica-se que em todos os sistemas analisados neste trabalho, houve a contribuição da matemática e probabilidade. Pode-se notar a presença dos logaritmos em vários tópicos deste trabalho, mas principalmente nas fórmulas de relevância e pesagem dos termos. A aplicação dos logaritmos nas aproximações e nos tráfegos de informações, vem sendo intensificada desde a década de quarenta. Como um dos precursores do uso do logaritmo e da probabilidade no campo da Computação, destaca-se Claude Elwood Shannon (Claude, 2001), denominado o pai do bit. Nascido nos EUA em 1916, com graduação e pós graduação em cursos de Ciências Exatas e Tecnológicas. Shannon (Claude, 2001) criou a *Teoria da Informação*.

Shannon (Claude, 2001) mostrou que aos elementos básicos do trabalho científico, "*massa e energia*", precisa-se acrescentar um terceiro elemento: a *informação*. Usando seus conhecimentos da Matemática com as Teorias das Probabilidades, mostrou como medir a quantidade de informação, introduzindo a unidade bit (*binary digit*) de informação. A solução deste problema é dada pela fórmula de Shannon (Claude, 2001), apresentado a seguir:

$$C_{\max} = B * \log_2 \left(1 + \frac{S}{N}\right) \qquad (29)$$

onde a velocidade máxima (C_{max}) é determinada pela potência (S) em watts que podem passar por um canal de comunicação, o qual deixa passar sem distorção apenas os sinais de freqüência até (B) hertz, e que produz ruídos de potência de no máximo (N) watts. Pode-se notar ao longo dos anos, a associação da Informática com a Física, Matemática e Estatística. E graças a isto pode-se projetar fórmulas e funções para a área da IR.

O sistema INQUERY usa o Modelo de Redes de Inferências, usando arquitetura Bayesiana. Como foi visto, a Rede de Inferência é um dos modelos baseados em Redes Bayesianas, que tem atraído muitos sistemas de recuperação de informações. Um dos componentes deste sistema que tem contribuído para a sua eficiência, é o analisador Parser, responsável em transformar os documentos em um formato canônico, ou seja, em um formato padrão para o INQUERY. A eficiência do INQUERY, em parte, é devido ao seu analisador Léxico do Parser associando as StopWord e Stemming.

As Redes de Inferência tratam a relação entre o documento e a *query* através de cálculos complexos de relevância de probabilidade. Elas permitem a combinação de múltiplas *queries* e estratégias de procura múltipla. Por este motivo, o sistema INQUERY tem sido eficiente com grandes bancos de dados. Apesar do processamento variar de acordo com a complexibilidade da *query* e o número de termos na coleção, o sistema requer relativamente muito espaço em disco e memória RAM. Seja uma coleção com 5 Gb de documentos, o INQUERY necessitaria de 25 Gb de espaço livre para construir a sua base de dados e 7,5 Gb para armazenar a base de dados de documentos e memória superior ou equivalente a 300Mb de RAM. Tanto o analisador Parser, como os cálculos complexos de relevância de probabilidade com combinação múltipla de queries, como visto, requerem muito recurso computacional. Construído para lidar com grandes bancos de dados, em muitas situações fica difícil prever a quantidade de memória física ou virtual.

Um processo que eventualmente poderia ser feito, seria intensificar os esforços nas StopWords, através de listas de paradas mais eficientes, podendo assim reduzir consideravelmente o índice final. O grande problema é que poderia reduzir a

eficiência do retorno. Uma outra alternativa, seria o uso do Algoritmo de Porter, removendo os sulfixos mais longos, usando uma lista de determinação.

Umas das vantagens deste sistema é a utilização de linguagem natural estruturada para os operadores *queries*. Outra vantagem do INQUERY é a utilização de um sistema operacional estável e confiável, o UNIX. Nota-se que as redes Bayesianas são eficientes na busca de informações em grandes bancos de informações. A fórmula de relevância dos termos do INQUERY tem sido muito eficiente, pois é mantido a mesma fórmula desde a sua primeira implantação.

O sistema THISL é na verdade um sistema de recuperação de informação por voz. Um componente que se destaca neste sistema é a grande performance do identificador de voz ABBOT, que reconhece a fala de grandes vocabulários, proporcionando ao sistema um grande diferencial, pois trabalhando com uma linguagem natural através de comando de voz, tornando o sistema acessível a um número maior de usuários, mesmo àqueles que não tem muito conhecimento na área da IR.

Após a solicitação do usuário, o sistema de recuperação por voz torna-se um sistema de recuperação de texto, o THILIR. Este sistema utiliza o modelo probabilístico padrão usando a teoria da probabilidade. Um dos grandes problemas deste modelo, é determinar verdadeira relação entre a coleção de documentos e a *query* fornecido pelo usuário, ou seja, determinar a probabilidade de relevância entre a necessidade de informação e a coleção de documentos.

Após processos heurísticos, o resultado da busca é geralmente uma lista de documentos ordenados em ordem de relevância. O THILS, especificamente utiliza-se de uma lista de parada, do algoritmo de Portes, e uma função de pesagem de termos. Na TREC-9 este sistema teve menos êxito do que na TREC-8, porque houve uma evolução no sistema e não houve um acompanhamento adequado em relação ao modelo adotado. Não

pode-se contestar a eficiência das listas de parada (*StopWord*), do algoritmo de Porter em remover sufixos longos. O sistema OKAPI foi projetado para trabalhar em grandes bancos de dados, requerendo muita memória virtual e física. O sistema THISL possui características diferentes, pois trabalha com comando de voz, exige uma maior performance em RealTime e possui uma arquitetura mais simples. Como visto, o THILS utiliza a função dos termos de pesagem da OKAPI.

$$RW(i) = \log \frac{(r+0{,}5)(N-n-R+r+0{,}5)}{(n-r+0{,}5)(R-r+0{,}5)} \qquad (30)$$

Um dos possíveis fatores na redução da precisão do sistema, da TREC-9 em relação a TREC-8, poderia ser o uso da mesma função de pesagem de termos, tendo características diferentes. Seria interessante o estudo de novas funções de pesagem de acordo com a realidade do sistema THILS, bem como a viabilidade da implementação de outro modelo probabilístico.

O sistema OKAPI utiliza um modelo probabilístico formal generalizado. Ou seja probabilidade relativa *P(L/D),* probabilidade relativa da ocorrência de um termo relevante *(L),* dado um conjunto de documentos *(D)*. Para a resolução de muitos problemas envolvendo a probabilidade, recorre-se à probabilidade condicional, em casos particulares necessita-se inverter esta probabilidade. Ao processo de inversão de uma probabilidade condicional, recorre-se ao Teorema de Bayes. Segundo BUSSAB, MORETITIN (2000), o Teorema de Bayes expressa uma probabilidade condicional em termos de outras probabilidades condicionais e marginais. Ou seja, dado *P(L/D)* necessita-se encontrar *P(D/L),* a probabilidade relativa de um conjunto de documentos, dado a ocorrência de um termo.

Este sistema atualmente é desenvolvido pela Microsoft Research, com Sistema Operacional Solaris 2.6 da SUN. Este sistema requer processamento paralelo com

processadores equivalentes ou superiores a 400 Mhz com 512 Mb de RAM ou superior. O espaço requerido em disco é alto, ou seja 170Gb com uma rede de 100Mbits. Como este sistema utiliza-se de uma busca única com uma lista de resultados, os esforços seriam para minimizar os itens desnecessários. O modelo adotado pelo sistema OKAPI, foi projetado para grandes índices e grandes quantidades de documentos. Num total de 18,6 milhões de documentos, tendo em média 312 índices por documentos, temos então 5.800 milhões de índices, ou seja, um índice de 14 Gb. Pode-se notar que o sistema operacional UNIX tem se mantido confiável em sistemas de IR. Apesar dos esforços em relação à melhoria da filtragem dos dados continue, pode-se notar que o modelo probabilístico adotado por este sistema, tem se tornado adequado e confiável, diante do tamanho e complexibilidade do sistema. Nota-se que o Teorema de Bayes foi de grande importância neste modelo.

As elaborações dos índices invertidas, não são processos muito complexos, ou de grande esforço computacional. A partir do momento que existe a necessidade de se incluir a pesagem dos termos nos processos de inversão dos índices, o processo se torna um pouco mais complexo. Pois o processo de inversão de índice toma três fazes:

1) Pesquisa em vocabulário, onde os termos presentes na pergunta são pesquisados no vocabulário;
2) Retorno das ocorrências, onde as listas de ocorrências de todos os termos encontradas são devolvidas;
3) Manipulação das ocorrências, onde as ocorrências são processadas por forma a responder à perguntas.

Diferente do sistema INQUERY, que dependendo da operação enfrenta a dificuldade em prever o tamanho da memória física e virtual, o sistema OKAPI realmente requer muita memória física e virtual. Uma solução para a diminuição dos índices, seria trabalhar com endereçamentos por blocos. Segundo Marinheiro (2001), onde se pode endereçar palavras, documentos ou blocos de 256 K. O endereçamento por blocos reduz consideravelmente o esforço computacional e memória exigida. Poderia-se implementar no

sistema OKAPI, outro método de endereçamento para a diminuição de seus índices, desde que não se perca a eficiência da IR.

O índice invertido é uma das estruturas mais adequadas de indexar grandes coleções textuais, pois evidencia-se isto nos estudos efetuados no sistema INQUERY. O índice invertido tem tido bom êxito em documentos semi-estáticos ou relativamente voláteis, como na coleção de documentos do sistema THILS, uma opção adequada para pesquisas on-line em tempo real.

.

QUADRO II - Quadro de comparação dos sistemas probabilísticos de IR

$$w_{t,d} = 0{,}4 + 0{,}6 * \frac{tf_{t,d}}{tf_{t,d} + 0{,}5 + 1{,}5\frac{length(d)}{avg\ len}} * \frac{\log\frac{N+0{,}5}{n_t}}{\log N + 1}$$

Sistemas	Desenvolvimento	Modelo Probabilístico	Método de Aproximação	Fórmula da Relevância dos Temos	Característica da Busca	S.O/ Hardware	Característica	Deficiência

$$\sum_{T \in Q} W(1)\frac{(k1+1)tf}{k+tf}\frac{(k3+1)qtf}{k3+qtf}$$

$$\log\frac{(r+0{,}5)/(R-r+0{,}5)}{(n-r+0{,}5)/(N-n+r+0{,}5)}$$

INQUERY	Universidade de Massachusetts – EUA.	Redes de Inferência . Redes Bayesianas.	Listas de proximidade, através de uma análise léxica (Parser), associado com StopWords e Stemming..		Filtragem com experimentos *ad-hoc*. Rede de Documentos Análise Léxica e Sintática Análise de Contexto Local	UNIX./ Digital Equipament Corporation e SUN Microsystem.	Eficiência em grandes bancos de dados. Habilidades em IR complexas. *Query* com linguagem natural.	Dificuldade em prever memória virtual e física. Dificuldade de prever o tempo de processo.
OKAPI	Microsoft Research – UK.	Probabilístico Generalizado.	Inversão de índice. Pesagem de termo.	onde W(1) é iqual a:	Filtragem adaptável Expansão do *ad-hoc*. Banco de dados com processamento de tópicos e indexação	SOLARIS/ SUN Ultra 10, Dell com 2 processadores e Processadores Intel Pentium.	Eficiente método de pesagem de termos. Eficiência em grandes bancos de dados.	Requer muita memória virtual e física. Tamanho dos índices elevado necessitando de uma considerável memória física.
THISL	Universidade de Sheffiel – UK. SoftSound – UK.	Probabilístico Padrão.	Lista de parada (StopWords) Algoritmo de	onde CFW(i) é igual a :	Filtragem. Identificador de Voz ABBOT	UNIX. Processadores Intel Pentium.	Query através de comando de voz com tecnologia de	Redução da precisão do sistema em IR, da TREC-9 em relação a TREC-8.

$$CW(i,j) = \frac{CFW(i) * TF(i,j) * (Ki + 1)}{Ki * ((1 - b) + (b * (NDL(i)))) + TF(i,j)} \log \frac{(r + 0{,}5)/(R - r + 0{,}5)}{(n - r + 0{,}5)/(N - n + r + 0{,}5)}$$

			Porter. Pesagem de termo da Okapi		Reconhecimento em tempo real Pesagem de Termos através da Seletividade		redes neurais. Performance RealTime. Baixa latência Arquitetura simples.	

Pode-se verificar no quadro anterior, os principais Sistemas de IR's referenciados na TREC. O Sistema INQUERY tem se mostrado significativamente robusto, primeiramente por adorar Modelo Probabilístico de Redes Bayesianas, eficiente nas operações em grandes bancos de informações, e por possuir o analisador léxico de padronização de textos, o analisador Parser. Lembrando que uma análise léxica, é converter uma seqüência de caracteres (o texto do documento) em uma seqüência de palavras, candidatas a serem termos índices. É importante mencionar a confiabilidade do INQUERY, pois tem utilizado a mesma fórmula de relevância dos termos desde as suas primeiras versões. Pode-se verificar a predominância do Sistema Operacional UNIX em sistemas de IR.

O Sistema OKAPI adota em suas aproximações, inversão de índices com pesagem de termos. Este procedimento admite que o sistema utilize um Modelo Probabilístico Generalizado. Grande parte da sua eficiência é devido a eficiente fórmula de pesagem dos termos definindo bem os temos relevantes dos não-relevantes. O Sistema Okapi, em termos de Sistema Operacional e Hardware se mostra também robusto, pois utiliza produtos da SUN. A OKAPI, juntamente com o INQUERY, enfrenta praticamente a mesma problemática, a de requerer considerável performance de Hardware. Nota-se que um dos elementos que coopera na nesta exigência, é a criação dos índices. Seria interessante estudar um outro método de tratamento destes índices.

O Sistema de IR THISL possui uma característica muito importante, a solicitação de da informação através de comando de voz. Uma vez solicitada a informação, o Identificador de Voz ABBOT converte o a solicitação em um Sistema de IR textual. O THILS trabalha com listas de parada (StopWords), que tem a finalidade de retirar do texto as palavras de grande freqüência, e reduzir também as palavras de

ligação; com o algoritmo de Porter, que trata em remover sufixos longos; e pesagens de termos da OKAPI. Este procedimento permite que o Sistema THILS trabalhe com um Modelo Probabilístico Padrão. A característica de recuperação em tempo real, aliado a uma arquitetura simples, tem sido um mecanismo eficiente nas buscas em bancos não muito complexos. Mesmo usando a tecnologia de Redes Neurais em seus algoritmos, à medida que este banco aumenta, exige-se uma maior eficiência, para que a performance em tempo real não seja afetada.

6 CONCLUSÃO

Neste trabalho, tratou-se conceitos e modelos de IR, abordando suas dificuldades e restrições; abordagens probabilísticas de IR; e estudos de alguns modelos e sistemas probabilísticos em Recuperação de Informação em base textual.

Foi possível verificar neste trabalho, a veracidade do Modelo Conceitual de Furh em IR, como visto anteriormente na figura 09. No processo de IR, três fatores devem ser colocados em destaque: o modelo adotado pelo sistema; fórmula de relevância de termos e o método de aproximação.

No Sistema INQUERY, o modelo probabilístico adotado é o de Redes de Inferência. Este modelo possui uma rede estática de documentos e uma rede dinâmica de *query*. Este modelo permite a exploração da representação de múltiplos documentos e *queries*, dentro do contexto de uma coleção particular de documentos. Apesar disso, dado uma única necessidade de informação, é possível combinar resultados de múltiplas *queries* e estratégias de procura múltipla. A fórmula de relevância dos termos do INQUERY, tem se mostrado eficiente em todas as versões do sistema. No método de aproximação do INQUERY, destaca-se a eficiência do Analisador Léxico do Parser, convertendo o texto do documento numa seqüência de palavras candidatas a serem os termos índices. Pode-se verificar que as Redes de Inferência realmente são eficientes em grandes bancos de informações, e que a eficiência do sistema é devido sua fórmula de relevância dos termos e do Analisar Léxico do Parser.

O Sistema OKAPI teve um grande avanço desde a TREC-1. Da TREC-1 até a TREC-6, o sistema foi desenvolvido pela Universidade de Westminster. A partir da TREC-7 o sistema está sendo desenvolvido pela Microsoft Research.. Este sistema utiliza um modelo probabilístico generalizado localmente implementado, utilizando

índices invertidos em suas aproximações. Um fator que se destaca neste sistema é a eficiência da fórmula de relevância de termos, através do peso proposto por Robertson/Spark. A Microsoft está direcionado seu esforço para que o Sistema OKAPI opere com bons resultados em grandes bancos de dados. Pode-se notar neste sistema, a notável interação da fórmula de relevância de termos com os índices invertivos.

O Sistema THILS possui uma característica que o diferencia dos demais sistemas de IR. Este sistema opera com recuperação de informação através de comando de voz. No interior do THILS existe um sistema auxiliar chamado de ABBOT, que transcreve o áudio da rede, transformando o problema em uma recuperação de texto. A troca do sistema PRISE de reconhecimento de texto por um sistema localmente implementado, usando um modelo probabilístico padrão, trouxe resultados mais precisos ao sistema. Isto mostra a eficiência deste modelo, visto que o sistema opera em tempo real como o usuário. O THILS opera com o algoritmo de Porter, reduzindo palavras a um radial mais simples. Um fator que pode ter contribuído para a sensível perda de performance na TREC-9, é a prática de uma fórmula de relevância de termos mais decisiva, pois várias fórmulas foram testadas durante o programa TREC. Pode-se notar a contribuição da pesagem de termo do Sistema OKAPI neste sistema, e a significativa eficiência da fórmula de pesagem de termos com o algoritmo de Porter.

Pode-se concluir que os modelos probabilísticos deram considerável contribuição para sistemas de IR, e que os tais modelos foram os primeiros modelos em IR com firme fundamentação teórica, e por causa disto, é bastante usado em grandes centros de desenvolvimento, como por exemplo na Microsoft. Pode-se verificar também, que a teoria da probabilidade é largamente usada no campo da informática, principalmente em situações incertas, característica da IR. Muitos são os sistemas que

lidam com incertezas, cada um com seu grau de heurística e complexidade, implicando numa diversidade de modelos probabilísticos.

Nota-se que os sistemas estão requerendo mais as filtragens do que as procuras *ad-hoc* . Como já abordado, nas filtragens a *query* se mantém relativamente estática, enquanto que o banco de informações se mantém em constante evolução. Pode-se dizer, em parte, que isto se deu com o advento da Internet.

Este estudo serviu para verificar se o funcionamento e limitações de alguns modelos e sistemas probabilísticos. No campo da IR, seja ela textual, visual ou auditiva, a grande barreira ainda é conseguir traduzir ao sistema, a necessidade do usuário, no intuito do retorno ser o mais significativo possível. Para isso, sistemas com processamento de linguagem natural e modelos probabilísticos estão sendo desenvolvidos e aperfeiçoados.

Muitas informações estão disponíveis, e os meios de acessá-las não as acompanham na mesma proporção. Um dos grandes problemas encontrados hoje é o tempo necessário para a análise da lista de retorno. Uma solução para a melhoria da qualidade da IR é criar novos modelos baseados na Teoria da Probabilidade ligados com a Inteligência Artificial, a fim de criar sistemas interativos capazes de assimilar ao máximo a necessidade do usuário.

Muitos sistemas transformam a linguagem natural em uma *query* estruturada. Seria salutar que os sistemas de recuperação investissem mais na interação da Inteligência Artificial com as *queries*, a fim de melhorar a eficiência do processo de recuperação. Por exemplo, se um usuário fizer uma busca com a seguinte query: ”*necessito de informação sobre energia atômica”*, poderia existir um agente inteligente que pudesse dialogar com o usuário, recuperando somente as informações relevantes à

sua pesquisa, o que não ocorre hoje em dia. Outro fator detectado é a importância de uma regulamentação para o estabelecimento de um padrão nos processos de recuperação de informações, principalmente na Rede Mundial de Computadores, como o padrão *ad-hoc* de busca adotado nos Estados Unidos.

As principais contribuições deste trabalho foram:

- Informar a grande necessidade do usuário em recuperação textual, embora existam outros objetos de recuperação;
- Mostrar que existe muita informação textual, e que a eficiência das ferramentas de buscas não acompanham na mesma proporção;
- Apresentar a importância da probabilidade em Recuperação de Informação em base textual;
- Informar a eficiência dos Modelos Probabilísticos em grandes bancos de informações, pois alguns dos sistemas analisados trabalham com estas características;
- Apresentar o funcionamento de alguns sistemas probabilísticos em IR;
- Necessidade do aumento dos esforços para a elevação na qualidade e velocidade das informações recuperadas.

Futuramente pode-se estudar a integração dos modelos probabilísticos com os demais modelos usados em Recuperação de Informação, tornando-os compatíveis e melhorados. Visto que um dos grandes meios de veiculação da informação atualmente é a Internet, pode-se propor estudos sobre novos sistemas de busca operando com modelos inteligentes e linguagem natural, ao invés dos tradicionais sistemas que operam através de palavras-chaves. Pode-se também efetuar uma análise estatística, objetivando

o índice de satisfação e compreensão dos usuários perante os diversos sistemas de recuperação existentes no mercado.

REFERÊNCIAS

ABBERLEY, D.; et al. Retrieval of broadcast news documents with the THISL system. In:. IEEE INTERNATIONAL CONFERENCE ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998, Seattle. **Procedings...** [S. l.: s. n.] p. 128-137. Disponível em: <http://citeseer.nj.nec.com/114622.html>. Acesso em: 15 abr. 2001.

ABBERLEY, D. et al. The THISL broadcast news retrieval system. In: ESCA WORKSHOP ON ACCESSING INFORMATION IN SPOKEN AUDIO, 1999. **Workshop...** Disponível em: <http://citeseer.nj.nec.com/abberley99thisl.html> Acesso em: 25 abr. 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023:** informação e documentação: referências: elaboração. Rio de Janeiro, 2000.

BAEZA YATES, R.; RIBEIRO NETO, B. A. **Modern information retrieval.** New York: ACM Press ; Harlow: Addison-Wesley, 1999. 513p.

BUSSAB, W.O.; MORETTIN, P. A. **Estatística básica.** 4.ed. São Paulo: Editora Atual, 2000. 321p.

CALLAN, J.P. et al. The INQUERY retrieval system. In: DATABASE AND EXPERT SYSTEMS APPLICATIONS: PROCEDINGS OF THE INTERNATIONAL CONFERENCE, 3., 1992, Valência. **Procedings...** New York: Springer Verlag, 1992. Disponível em: <http://citeseer.nj.nec.com/26307.html> Acesso em: 04 mar. 2001.

CLAUDE SHANNON, O PAI DO BIT. Disponível em: <http://athena.mat.ufrgs.br/~portosil/passa1c.html> Acesso em: 10 set. 2001..

COOPER, William S.; GEY, Fredric C.; DABNEY, Daniel P. Probabilistic retrieval based on staged logistic regression. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 15., 1992, Copenhagen Denmark. **Procedings...** New York: ACM, 2000. Disponível em: < http://www.acm.org/pubs/citations/proceedings/ir/133160/ p198-cooper/> Acesso em: 24 jul. 2001.

CRESTANI, F. **A survey on the application of Neural Networks's supervised learning procedures in Information Retrieval**. Rapporto Tecnico CNR 5/85, Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo - P5: Linea di Ricerca Coordinata Multidata, Dec. 1991. Disponível em: <http://www.cs.strath.ac.uk/~fabioc/papers/94-icnn.pdf> Acesso em: 23 jan. 2001.

CRESTANI, F. et al. “Is this document relevant? ... probably”: a survey of probabilistic models in information retrieval. **ACM Computing Surveys**, New York, v.30, no.4, p. 528-552, Dec. 1998. Disponível em: <http://www.acm.org/pubs/citations/journals/surveys/ 1998-30-4/p528-crestani/> Acesso em: 08 out. 2000.

DOWNING, D.; CLARK, J. **Estatística.** São Paulo: Saraiva, 1998. 455p.

FUHR, N. Integration of probabilistic fact and text retrieval. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 15., 1992, Copenhagen, Dennmark. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/citations/ proceedings/ir/133160/p211-fuhr> Acesso em: 10 fev. 2001.

FUHR, N.; BUCKLEY, C. Probabilistic document indexing from relevance feedback data. In: ANNUAL ACM CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL; INTERNATIONAL CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 30., 1990, Brussels, Belgium. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/ citations/proceedings/ir/96749/p45-fuhr/> Acesso em: 24 jul. 2001.

FUHR, N. Probabilistic models in information retrieval. **The Computer Journal**, v.35, no.3, p.243-255, 1992. Disponível em: <http://citeseer.nj.nec.com/fuhr92probabilistic.html> Acesso em: 02 fev. 2001, 24 jul. 2001.

FUHR, Norbert. A probabilistic relational model for the integration of IR and databases. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN **N**FORMATION RETRIEVAL. 16., 1993, Pittsburgh, PA USA. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/ citations/proceedings/ir/160688/p309-fuhr/> Acesso em: 25 jul. 2001.

FURNAS, G. W. et al. Information retrieval using a singular value decomposition model of latent semantic structure. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 11., 1988. **Proceedings...** New York: ACM, 1988. Disponível em: <http://www.acm.org/ pubs/citations/proceedings/ir/62437/p465-furnas> Acesso em: 04 abr. 2001.

HAINES, David; CROFT, W. Bruce. Relevance feedback and inference networks. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 16., 1993, Pittsburgh, PA USA. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/citations/proceedings/ir/160688/p2-haines/> Acesso em: 24 jul. 2001.

HULL, D. A.; ROBERTSON, S. The TREC-8 filtering track final report. In: NIST SPECIAL PUBLICATION 500-246: Text Retrieval Conference. 8., 1999, Washington, DC. **Proceedings...** Washington, DC: NIST, 1999. Disponível em: <http://trec.nist.gov/pubs/trec8/ t8_proceedings.html > Acesso em: 02 jun. 2001, 28 jul. 2001.

JONES, K. S. et al. **A probabilistic model of information retrieval**: development and status. Cambridge, UK: University of Cambridge. Computer Laboratory, 1998. Technical report. Disponível em: <http://citeseer.nj.nec.com/jones98probabilistic.html> Acesso em: 17 de maio 2001.

KARP, R. M. **An introduction to randomized algorithms**. Berkeley: University of California, Computer Science Division, [1990]. Technical Report TR-90-024.

MARGULIS, Eugene L. N-Poisson document modelling. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 15., 1992, Copenhagen Denmark. **Procedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/citations/proceedings/ir/133160/p177-margulis/> Acesso em: 24 jul. 2001.

MARINHEIRO R. N. Pesquisa de Informação em Texto – Introdução e Indexação. Disponível em: <http://iscte.pt/~rmnm/web/ipbdm/acetatos/acet05.pdf> Acesso em: 11 set. 2001..

NATIONAL INSTITUT STANDARDS AND TECHNOLOGY. Text retrieval conference (TREC): overview. Apresenta eventos sobre Recuperação da Informação. Disponível em: <http://trec.nist.gov/overview.html> Acesso em: 20 abr. 2001.

RENALS, S.; ABBERLEY, D. The THISL SDR system at TREC-9. In: NIST SPECIAL PUBLICATION XXX-XXX: Text Retrieval Conference, 9., 2000, Washington, DC. **Proceedings...** Washington, DC: TREC, 2000. Disponível em: <http://trec.nist.gov/ pubs/trec9/ t9_proceedings.html> Acesso em: 05 maio 2001.

RIBEIRO, Berthier A. N.; MUNTZ, Richard. A belief network model for IR. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 19., 1996, Zurich, Switzerland. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/citations/proceedings/ir/243199/p253-ribeiro/> Acesso em: 27 jul. 2001.

ROBERTSON, S. Re: Ad Hoc (Mensagem Pessoal). Mensagem recebida por <ser@microsoft.com> Acesso em: 21 de jun. 2001.

ROBERTSON, S.; HULL, D.A. The TREC-9 filtering track final report. In: NIST SPECIAL PUBLICATION XXX-XXX: Text Retrieval Conference, 9., 2000, Washington, DC. Proceedings... Washington, DC: TREC, 2000. Disponível em:<http://trec.nist.gov/pubs/ trec9/t9_proceedings.html> Acesso em: 28 jul. 2001.

ROBERTSON, S; WALKER, S. Okapi/Keenbow at TREC-8. In: NIST SPECIAL PUBLICATION 500-246: Text REtrieval Conference, 8., 1999, **Proceedings...** Washington, DC. Washington, DC: NIST, 1999. Disponível em: <http://trec.nist.gov/pubs/trec8/ index.track.html> Acesso em: 15 maio 2001.

SALTON, G. et al. Extended Boolean information retrieval. **Communications of the ACM**, v.26, no.11, p.1022-1036, Nov. 1983. ISSN 0001-0782. Disponível em: <http://www.acm.org/pubs/citations/journals/cacm/1983-26-11/p1022-salton> Acesso em: 18 mar. 2001.

SAVOY, J. **Underlying problems of information retrieval**. Artigo apresentado no site institucional da Université de Neuchâtel. Institut Iterfacultaire D'informatique. Disponível em: <http://www-seco.unine.ch/Info/GI/ir.htm> Acesso em: 10 dez. 2000.

TAK, Woon Yan, et al. From user acess patterns dynamic hypertext linking. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 5., 1996, Paris, France. **Proceedings...** Disponível em: <http://www5conf.inria.fr/fich_html/papers/P8/Overview.html> Acesso em: 20 de nov. 2000.

TURTLE, Howard; CROFT, W. Bruce. Evaluation of an inference network-based retrieval model. **ACM Transactions on Information Systems**, v. 9, no. 3, p.187-222, 1991. Disponível em: <http://www.acm.org/pubs/citations/journals/tois/1991-9-3/p187-turtle/> Acesso em: 24 jul. 2001.

VOORHESS, E.; HARMON, D. Overview of the Fifth Text REtrieval Conference (TREC-5). In: NIST SPECIAL PUBLICATION 500-238: Text Retrieval Conference, 5., 1996, Washington, DC. **Proceedings...** Washington, DC: NIST, 1996. Disponível em: <http://trec.nist.gov/pubs/trec5/t5_proceedings.html > Acesso em: 11 abr. 2001.

WALKER, S. ET AL. Okapi at TREC-6 Automatic ad hoc, VLC, routing, filtering and QSDR. In: NIST SPECIAL PUBLICATION 500-240: Text Retrieval Conference, 6., 1996, Washington, DC. **Proceedings...** Washington, DC: NIST, 1996. Disponível em: <http://trec.nist.gov/pubs/trec6/t6_proceedings.html> Acesso em: 28 maio 2001.

WEIDE, Th. P. van den. **Information Discovery,** 2001. Disponível em: <http://www.cs.kun.nl/is/edu/ir1> Acesso em: 24 jul. 2001.

WOND, S.K.M. et al. Generalized vector space model in information retrieval. In: ANNUAL ACM CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL; ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 8., 1985, Montreal, Canada. **Proceedings...** New York: ACM, 2000. Disponível em: <http://www.acm.org/pubs/ citations/proceedings/ir/253495/p18-wong> Acesso em: 05 mar. 2001.

WONG, S. K. M.; YAO, Y. Y. On modeling information retrieval with probabilistic inference. **ACM Transactions on Information Systems**, v.13, no.1, p.38-68, 1995. Disponível em: <http://www.acm.org/pubs/citations/journals/tois/1995-13-1/p38-wong> Acesso em: 24 jul. 2001.