UNIVERSIDADE DE BRASÍLIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
CURSO DE MESTRADO EM INFORMÁTICA

# APRENDIZAGEM ESTATÍSTICA PARA RECUPERAÇÃO DA INFORMAÇÃO

EDMILSON FARIA RODRIGUES

Dissertação submetida à avaliação
como requisito parcial para a obtenção do grau de
Mestre em Informática

Prof. Dr. Marcelo Ladeira
Orientador

Brasília, Distrito Federal

Fevereiro de 2008

# CIP - Catalogação na Publicação

Rodrigues, Edmilson Faria
Aprendizagem estatística para recuperação da informação/ Edmilson Faria Rodrigues. - Brasília: CIC da UnB, 2008.
63p.: il.

Dissertação (mestrado) – Universidade de Brasília. Programa de Mestrado em Informática, Brasília, BR – DF, 2008. Orientador: Ladeira, Marcelo.

1. Recuperação da Informação. 2. Tratamento Estatístico da Linguagem. 3. Expectation Maximization. 5. Representação do conhecimento. 6. Inteligência artificial. I. Ladeira, Marcelo.

# Agradecimentos

Primeiramente, agradeço aos meus pais por terem, desde muito cedo, ensinado a importância da educação, explicando sempre que era um bem inesgotável, o qual nunca poderia ser subtraído.

À Thaís, minha mulher, pelo carinho que me dedicou esses anos, pela compreensão e pelo apoio nos momentos mais difíceis dessa caminhada, sempre com muito otimismo e alegria. Não teria chegado até aqui sem o seu apoio.

Ao Aroldo, colega de mestrado, que, com seu otimismo inigualável, sempre estava pronto a dar um sorriso e dizer que tudo daria certo proferindo o seu tradicional: "Relaxa Edmilson", daquela maneira pausada que dava preguiça só de ouvir. Muito obrigado pelo apoio, pelas explicações em Teoria da Computação, e pelas relaxantes partidas de ping-pong!

Ao Geci, meu chefe no Tribunal de Contas da União, o qual, de pronto, compreendeu a importância do meu trabalho e guiou-me com sabedoria e equilíbrio nos momentos em que passei pela difícil tarefa de conciliar trabalho e estudos.

Ao meu orientador, Marcelo Ladeira, que foi um pai, um guru e um psicólogo para mim durante dois preciosos anos de minha vida. Sua companhia bem humorada e exigente nesses dois anos fará com que sempre lembre dele como um exemplo de conduta, compromisso e amor à pesquisa.

Ao Professor Flávio de Moura, pela maestria com que transmitiu o conteúdo da difícil disciplina de Teoria da Computação. Sua posição humilde, de quem entende a dificuldade do aluno, e se preocupa com seu sucesso, é um exemplo de amor à difícil arte de ensinar.

# Sumário

# Índice de Figuras

# Resumo

A recuperação da informação pode ser entendida como uma área da ciência que se dedica ao estudo de técnicas de armazenamento de documentos e de recuperação de informação neles contidas, utilizando ou não metadados que os descrevem. Nos dias atuais em que as ferramentas de busca na Internet tornaram possível pesquisar documentos produzidos pelo mundo inteiro, o acesso à informação relevante torna a precisão na recuperação da informação uma demanda que ganha cada vez mais importância. Da necessidade do Tribunal de Contas da União de melhorar os resultados da precisão e da revocação da sua pesquisa textual jurisprudencial nasceu a motivação para o presente trabalho. A precisão é o percentual de documentos relevantes em relação ao número de documentos retornados na consulta [Kent et al., 1955]. A revocação é o percentual de documentos relevantes em relação ao número de documentos relevantes do corpus de documentos [Kent et al., 1955]. Os mecanismos de recuperação da informação devem ser capazes de auxiliar o usuário que, em geral, não tem conhecimento da forma exata em que ocorrem os termos nos documentos que contém a informação que procura. Um esforço que tem sido feito no sentido de contornar esse problema é a utilização de ontologias ou tesauros para ampliar a consulta solicitada pelo usuário [Miller, 1990]. No entanto, essa alternativa envolve um esforço em recursos humanos, financeiro e tempo muito grande para a construção dessas estruturas. Nessa pesquisa é proposta a utilização de um modelo estatístico da linguagem, derivado da tradução estatística da linguagem [Brown et al.,1993], para ampliar a consulta solicitada pelo usuário. Nessa abordagem é utilizado um algoritmo de EM (do inglês, *Expectation Maximization*) [Dempster, Laird & Rubin, 1977] para estimar índices de similaridades entre termos dos documentos. Nesta abordagem, cada consulta retorna os documentos contendo os termos nela contidos e os termos que são similares àqueles. Com essa metodologia, espera-se melhorar a precisão sem reduzir a revocação. Para permitir uma avaliação experimental com corpus com milhares de documentos, o algoritmo EM foi alterado para permitir a manipulação de matrizes esparsas e gerência de memória virtual. Foram introduzidas alterações na ferramenta aberta de recuperação de informação Terrier [Ounis et al. 2006] visando permitir que a indexação e recuperação considerem similaridades. Os experimentos realizados consideram corpora em língua inglesa (Medline e CFC) para permitir utilizar a metodologia de avaliação da TREC (Text Retrieval Conference). Foram também realizados experimentos em língua portuguesa (corpus CETEN-Folha) mas para eles não foi possível aplicar a metodologia de avaliação internacional. Os resultados obtidos até o momento são iniciais e não permitem afirmar que a utilização da metodologia proposta no sistema de recuperação de textos do TCU possa superar o desempenho do sistema atual. No entanto, espera-se uma melhora potencial visto que os resultados obtidos com os corpora da TREC são relativamente próximos aos obtidos com os melhores algoritmos de recuperação implementados no Terrier.

PALAVRAS-CHAVES: Recuperação da Informação. Tratamento Estatístico da Linguagem. *Expectation Maximization*. Representação do conhecimento.

# Abstract

Information retrieval (IR) is the science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within databases. Nowadays, when search engines make worldwide browsing an ubiquitous activity, there is a growing demand for precise information retrieval. The motivation for the present work results from the Brazilian Court of Audit (from portuguese, TCU) requirements for achieving better results in terms of precision and recall of its information retrieval systems. Precision is measured as the fraction of documents retrieved that are relevant to the user's information need [Kent et al., 1955]. Recall is the fraction of documents that are relevant to the query that are successfully retrieved [Kent et al., 1955]. The retrieval mechanisms of information retrieval must be able to support the user that, in general, doesn´t know the exact word used in relevant documents to express the information needed. The use of an ontology such as WordNet [Miller, 1990] is a popular approach for addressing this issue. However, this approach implies in a huge effort by human specialists to build these structures. In the present survey , our approach is the use of an language model derived from statistical machine translation [Brown et al.,1993] to expand the user´s queries. We use Expectation Maximization algorithm [Dempster, Laird & Rubin, 1977] to estimate the similarities between terms within the documents. In this approach, each query retrieves not only the documents that contain the terms of the query but also the terms that are similar to them. With this methodology we hope to increase precision without decreasing recall. To allow for experimental evaluation on a corpus with thousands of documents, the EM algorithm was modified to allow the handling of sparse matrix and virtual memory management. The open platform for Information Retrieval, Terrier [Ounis et al. 2006], was modified in order to enable similarities to be handled by the indexing and retrieval modules. The accomplished experiments used corpora in english language (Medline and CFC) to allow the application of TREC international evaluation methodology. The results achieved so far are preliminary and cannot yet support the claim of having provide substantial improvements to the TCU´s information retrieval systems. Nevertheless, we hope a substantial improvement on these systems as far as the results obtained so far with TREC english corpora are comparable to those obtained with the state-of-the-art theoretically-founded models for IR that Terrier implements.

KEYWORDS: Information Retrieval, Statistical Language Modelling, Expectation Maximization, Knowledge Representation

# Capítulo 1. Introdução

Neste capítulo, é apresentada a especificação geral do problema abordado, sua relevância e as áreas de pesquisas relacionadas.

## 1.1 Definição do Problema

Um usuário pode procurar por documentos sobre um assunto sobre o qual tenha particular interesse basicamente de duas formas. Ele pode navegar sobre um conjunto de documentos, seguindo os elos relacionados aos assuntos que lhe despertam maior interesse ou pode dispor de um sistema de busca de documentos, para o qual ele apresenta uma consulta a ser utilizada para recuperar a informação. O primeiro caso é definido como *browsing,* já o segundo caso é conhecido como recuperação da informação.

A recuperação da informação é usualmente definida como o processo de selecionar e combinar dados explicitamente existentes ou deduzidos em um ou mais documentos em linguagem natural. Este processo envolve uma classificação semântica de peças de informação e pode ser entendido como uma forma de "entendimento" da linguagem natural [Moens, 2006]. Na recuperação da informação *ad hoc* o usuário apresenta uma consulta e o sistema recupera os documentos que julgou estarem relacionados com a consulta apresentada. Já na filtragem, não há uma consulta, o sistema recupera os documentos de acordo com o perfil do usuário, como em um serviço de notícias, por exemplo.

Um sistema de recuperação da informação deve atender principalmente a dois requisitos: precisão e revocação. A precisão é uma medida que indica a relação entre o número de documentos recuperados relevantes para a consulta realizada e o número de documentos recuperados. Já a revocação (do inglês, *recall*) é definida pela relação entre o número de documentos relevantes recuperados e o número de documentos relevantes existentes na base de documentos.

Um dos fatores que contribui para a deterioração dessas medidas nos algoritmos de recuperação de informação tradicionais é a incapacidade desses sistemas de recuperar palavras similares, ou seja, palavras cujo significado é próximo do significado da palavra utilizada na consulta.

Assim, a recuperação da informação (RI), exclusivamente via casamento de palavras, tende a apresentar uma baixa precisão, uma vez que os documentos recuperados, ainda que contenham as palavras empregadas na consulta, podem não contê-las com o mesmo significado empregado na consulta. Por exemplo, se o usuário faz a consulta: "Qual o tipo de adubo utilizado no plantio de mangas?", o sistema poderá recuperar documentos sobre peças de vestuário, pois a ocorrência da palavra "adubo" na consulta não desqualifica um documento sobre peças de vestuário, o que seria intuitivo do ponto de vista humano, já que a palavra "manga" contida em vestuário tem significado distinto daquele utilizado na consulta.

De igual sorte, documentos contendo palavras como “plantio”, “aragem”, “solo”, “pomar”, “agricultura”, dentre outros termos referentes ao mesmo contexto, podem ser considerados relevantes para a consulta apresentadas, mas provavelmente não seriam recuperados pelos mecanismos tradicionais de recuperação da informação.

Neste caso, faz-se necessário dispor de mecanismos, lingüísticos ou estatísticos, de identificação de relações de sinonímia, hiperonímia, hiponímia, meronímia, holomínia e antonímia para identificar os termos relacionados àqueles empregados na consulta.

Para obter esses relacionamentos é necessária a construção de uma ontologia, o que requer um grande esforço e a participação de especialistas no domínio onde o sistema de RI será utilizado, implicando em identificar milhões de relações existentes entre milhares de palavras.

Além dessa dificuldade, as relações acima descritas podem não ser suficientes para identificar todas as maneiras pelas quais dois termos quaisquer se relacionam. Por exemplo, em uma base de documentos do domínio do direito penal, um documento contendo as palavras “prisão”, “liberdade”, “cárcere” pode ser relevante para uma consulta contendo o termo “ditadura”, no entanto, nenhum daqueles possui qualquer uma das relações semânticas supracitadas com este.

É evidente que o problema da melhoria da precisão e da revocação na recuperação de documentos está relacionado ao problema de identificação de similaridades entre as palavras. Estas similaridades são dependentes do domínio onde os termos são empregados, e sua identificação manual é um esforço gigantesco que não necessariamente identifica todas as relações de similaridade existentes entre os termos em um determinado contexto.

É exatamente esse problema que a modelagem estatística da linguagem tenta focar. A criação de modelos que refletem as relações existentes entre as palavras das consultas com as palavras dos documentos é uma forma de capacitar os mecanismos de busca a buscar por palavras similares, o que, no contexto da recuperação da informação, é um problema fundamental e bem recorrente, já que o usuário não tem como prever como está empregado no texto o termo pelo qual ele procura, resultando em baixa precisão nos resultados apresentados e na conseqüente insatisfação do usuário [Baeza & Ribeiro-Neto, 1999]

## 1.2 Objetivo

O objetivo geral deste trabalho é pesquisar algoritmos de recuperação da informação *ad hoc* que permitam ampliar uma consulta, obtendo bom desempenho em termos de precisão média, sem requerer a construção de estruturas auxiliares tais como tesauros ou ontologias. A precisão média, conforme descrito na seção 4.4, é uma medida que privilegia sistemas de recuperação de informação que possuírem melhor precisão na média de todo intervalo de revocação possível. Tal medida foi escolhida um vez que avaliar um sistema de recuperação da informação segundo a sua revocação, exclusivamente, somente é aplicável a domínios onde se conhece de antemão o nível de revocação esperado pelo usuário, ou seja, há uma

ampla faixa de necessidades que são dependentes do domínio da aplicação do sistema de recuperação da informação. Essa faixa de necessidades varia desde aqueles domínios em que basta ao usuário encontrar um documento relevante, até aqueles em que o usuário deseja encontrar todos os documentos relevantes.

Os objetivos específicos são os seguintes:

- ✓ conhecer a arquitetura de sistemas de recuperação da informação e propor alterações visando obter maior desempenho com ampliação de consulta, com base na descoberta automática de similaridades entre termos.

- ✓ desenvolver ou aprimorar modelos baseados no tratamento estatístico da linguagem que identifiquem, em um domínio específico, o grau de similaridades entre os termos da base de documentos a ser pesquisada.

- ✓ utilizar mecanismos de avaliação de sistemas de RI, desenvolvidos pela Conferência Internacional em Recuperação da Informação (do inglês, *TREC*) [Ounis et al. 2006], para comparar performance na recuperação da informação.

- ✓ avaliar o algoritmo de recuperação da informação a ser proposto, baseado no tratamento estatístico da linguagem inglesa, utilizando a metodologia de avaliação da TREC, com "stopwords" e "stemmers".

- ✓ realizar uma análise da dificuldade de aplicação do modelo de RI proposto, a um corpus em língua portuguesa.

## 1.3 Áreas de Pesquisas Relacionadas

O termo Recuperação da Informação foi criado por Calvin Mooers entre 1948 e 1950 [Eugene Garfield, 1997], e o campo de pesquisa é interdisciplinar, baseado em muitas áreas. Por sua abrangência ele não é muito bem compreendido, sendo abordado tipicamente sob uma ou outra perspectiva. Ele está posicionado na junção de muitas áreas já estabelecidas, tais como psicologia cognitiva, arquitetura da informação, projeto da informação, comportamento da informação humana, lingüística, semiótica, ciência da informação, ciência da computação, biblioteconomia e estatística.

Para o objetivo geral desta pesquisa, podemos utilizar a lingüística e a ciência da informação para construirmos estruturas lingüística que contribuem para ampliar a consulta. Nesta pesquisa não abordaremos essa alternativa por ser intensiva em utilização de recursos

humanos especializados e de alto custo. Por outro lado, o uso de vocabulário controlado permitiria uma padronização nos termos a serem utilizados para descritores de um documento, facilitando a tarefa de recuperação em uma consulta. No entanto essa alternativa limitaria o uso do sistema de recuperação de informação, impedindo o uso de linguagem natural nas consultas, requerendo que os usuários fossem treinados e tivessem conhecimento do vocabulário controlado utilizado.

Já os modelos clássicos puros de recuperação da informação: booleano, vetorial e probabilístico, não consideram a utilização de estruturas linguísticas auxiliares. O modelo booleano trabalha com a lógica booleana para utilizar operadores "E", "OU" e "NÃO" na recuperação de documentos. Esses operadores não são de utilização intuitiva para usuários não treinados, sendo que a combinação deles podem ter significado diferente daquela que o usuário em geral imagina ao fazer a consulta.

Já o modelo vetorial identifica consultas e documentos como sendo vetores no espaço $\Re^n$ e um documento é tanto mais relevante para uma consulta quanto mais alinhados estiverem esses vetores. Embora essa abordagem seja mais flexível do que o modelo vetorial, a necessidade da ocorrência exata no documento do termo consultado obriga o usuário a conhecer as palavras que foram utilizadas nos documentos relevantes para expressar a informação que procura, o que, obviamente, é inviável em grandes bases de dados.

O modelo probabilístico procura atribuir pesos a cada ocorrência da palavra da consulta no documento e na própria consulta segundo a probabilidade de que aquele termo identifique o documento, dentro do conceito de que há termos que caracterizam um documento. No entanto, esse modelo sofre do mesmo problema que apontamos no parágrafo anterior para o modelo vetorial, qual seja o de buscar somente pela ocorrência exata da palavra.

Assim sendo, utilizaremos métodos derivados da estatística para construir estruturas auxiliares que aproximem estruturas lingüísticas convencionais tais como tesauros ou ontologias, como descrito nas Seções 4.2 e 4.3.

Por muitos anos, a modelagem estatística da linguagem foi usada primordialmente para reconhecimento de voz. Desde 1980, quando foi proposto o primeiro modelo significativo [Rosenfeld, 2000], a modelagem estatística de linguagem tornou-se um recurso fundamental no reconhecimento de voz, tradução automática, correção ortográfica, etc. A teoria que fundamenta o reconhecimento de voz é parte da teoria das Cadeias de Markov Ocultas (do inglês, HMM) que foi desenvolvida por Leonard Baum e seus colaboradores na IBM no início da década de 1970 [Rabiner, 1990][Jelinek, 1997]. Recentemente, as Cadeias Ocultas de Markov são estudadas como parte de um formalismo gráfico geral que engloba muitos dos modelos probabilísticos multivariados usados em estatística, teoria da informação e reconhecimento de padrões. Exemplos incluem redes bayesianas, análise de fatores e filtros de Kalman [Jordan, 1998] [Bengio, 1999].

A modelagem estatística da linguagem também manifestou ser bastante útil em

tarefas de processamento de textos, tais como geração de texto em linguagem natural e sumarização [Lawrie, 2003]. Em 1998, esta abordagem foi introduzida no campo da recuperação da informação, abrindo caminho para novas formas de se pensar o processo de recuperação.

O primeiro uso da modelagem estatística para RI focava na sua efetividade empírica usando modelos simples. O trabalho do sistema era então estimar a probabilidade de cada documento na coleção de documento (do latim, *corpus*) ser o documento procurado e avaliá-lo segundo esta estimativa. Este modelo, que foi primeiramente proposto por [Ponte & Croft, 1998], e posteriormente descrito em termos de um modelo de canal de ruído por [Berger & Lafferty, 1999], tem produzido resultados pelo menos comparáveis às melhores técnicas de recuperação da informação. O modelo básico foi estendido de várias formas. Exemplo disso são os modelos baseados em classes latentes estatísticas [Hofmann, 1999] e a combinação com outros modelos estatísticos [Song & Croft, 1999].

## 1.4 Organização desta Dissertação

Esta dissertação está organizada da seguinte forma: no Capítulo 2 é apresentado o estado da arte em modelos para recuperação da informação. No Capítulo 3 é descrita a arquitetura do Terrier, uma ferramenta de código aberto para pesquisa em recuperação da informação e que serviu de base para implementação das abordagens propostas no presente trabalho. No Capítulo 4 a abordagem proposta é apresentada e detalhada. No Capítulo 5, é apresentado o estudo de caso para duas bases no formato disponibilizado pela TREC e uma base em língua portuguesa, a qual não contém consultas nem julgamentos de relevância associados, os resultados obtidos para cada uma das bases são apresentados e analisados e comparados com outras abordagens. Por fim, conclusões e trabalho futuro são descritos no Capítulo 6.

# Capítulo 2. Modelos de Recuperação da Informação

Neste capítulo é apresentada uma revisão da literatura em modelos de recuperação da informação baseados no tratamento estatístico da linguagem. Este capítulo está organizado da seguinte maneira: a Seção 2.2 discute o modelo booleano, na Seção 2.3 é apresentado o modelo vetorial e na Seção 2.4 , a modelagem estatística da linguagem.

## 2.1. Introdução

Sistemas de recuperação da informação geralmente se baseiam na criação de índices. Um índice é uma estrutura de dados que permite a esses sistemas identificar onde um termo ocorre. No entanto, não podemos perder de vista que a criação de um índice é baseada nas ocorrências de um conjunto de palavras, ou seja, um conjunto de sinais, os quais podem ser usados para representar um ou mais conceitos. Mais ainda, esses conceitos podem não trazer muito ou mesmo nenhum significado quando apresentados isoladamente, pois tendem a ser usados para formar uma sentença, a qual pode ser uma proposição, uma suposição, etc.

Assim, fica claro que muito da semântica se perde quando se substitui uma frase, ou conjunto de frases em um texto por um conjunto de palavras (saco de palavras), logo, o casamento puro de palavras da consulta e palavras do texto, embora seja uma abordagem simples e frequentemente utilizada, geralmente não gera bons resultados, uma vez que não expressa, necessariamente, conteúdo semântico. Se considerarmos também a falta de treinamento do usuário em apresentar as consultas, o resultado dessa abordagem pode ser desastroso. A insatisfação dos usuários com os mecanismos de busca na Web é um exemplo disso [Baeza & Ribeiro-Neto, 1999].

Outro aspecto a ser considerado é que quando o usuário quer recuperar um documento, não há um meio pelo qual este possa saber como o termo procurado está colocado no documento, conceitos semelhantes podem estar sendo empregados com termos distintos nos documentos e nas consultas. Particularmente com consultas de tamanho reduzido, como, por exemplo, é bem típico no caso dos mecanismos de busca na internet onde o tamanho médio é de dois termos por consulta [Croft et al., 1995], existe uma menor probabilidade de que os termos que ocorrem na consulta, ocorram também nos documentos. Embora, uma consulta com apenas dois termos seja um extremo na maioria das aplicações em recuperação da informação, esta constatação sugere que deve haver um meio de tratar este tipo de problema. Nesse contexto, é muito difundido o método de expansão de consultas.

Basicamente, esta abordagem consiste em expandir a consulta apresentada pelo usuário com termos que possuam uma relação semântica com aqueles que pertencem à consulta original. A finalidade é possibilitar a recuperação de documentos que, mesmo sem possuir termos com a mesma grafia daqueles que foram apresentados na consulta original, são relevantes por tratarem do mesmo assunto buscado pelo usuário, com palavras distintas às que este apresentou. O que distingue os vários tipos de expansão de consultas é o método pelo

qual esses termos complementares são escolhidos. Duas abordagens podem ser adotadas: o uso de dicionários de sinônimos e o uso de palavras que co-ocorrem com os termos das consultas em documentos da coleção. No caso de dicionários de sinônimos os resultados obtidos não são em geral muito bons, em virtude da baixa precisão resultante da grande quantidade de documentos recuperados em corpus de grandes dimensões [Moens, 2006]. Melhorias consideráveis foram alcançadas quando se considerou análise automática de termos que co-ocorrem em documentos da coleção.

Um sistema de recuperação da informação trabalha de acordo com a noção de relevância de um documento, para obter essa medida o sistema se baseará em um modelo que utilizará um conjunto de premissas para estabelecer a relevância de um documento. Assim, por exemplo, um determinado modelo pode estabelecer que a ocorrência exata de qualquer termo da consulta dentro do documento contribui com igual peso para definição da medida de relevância de um documento, já outro modelo pode estabelecer que palavras que tem alta freqüência relativa (razão entre freqüência no documento e freqüência no *corpus*) contribuem com um peso maior na medição da relevância de um documento.

A pesquisa em recuperação da informação tem seguido diversas vertentes, dentre as quais podemos destacar a booleana, a vetorial, probabilística e lingüística.

No tocante à vertente lingüística, até o presente momento não foi descoberta uma maneira de se agregar a semântica existente numa frase ao processo de recuperação da informação, mesmo porque a mesma frase pode ser escrita de diversas maneiras, e qualquer tentativa de interpretar conceitos expressos em frases deve ser capaz de tratar elipses, inversões, figuras de linguagem, etc. Por isso há certa relutância em incorporar o tratamento de linguagem natural ao processo de recuperação da informação [Lewis and Jones, 1996].

No modelo booleano, os documentos e as consultas são representados como conjuntos de termos, e as operações de busca de documentos são baseadas em operações da teoria dos conjuntos.

No modelo do espaço vetorial, o documento e as consultas são representados como vetores, onde cada índice representa a ocorrência de um determinado termo no documento e na consulta, a definição de relevância do documento é definida a partir do grau de alinhamento entre o vetor que define a consulta e o vetor que define o documento.

Já o modelo probabilístico trata o processo de recuperação de documentos por meio de inferência probabilística, são calculadas similaridades como a probabilidade de que um documento seja relevante para uma dada consulta. Teoremas probabilísticos como o teorema de Bayes são freqüentemente usados nestes modelos.

## 2.2.O Modelo Booleano

O modelo booleano é o primeiro e provavelmente o mais criticado modelo de recuperação da informação. O modelo pode ser entendido se pensarmos que um termo de uma consulta é uma definição não ambígua para um conjunto de documentos. Por exemplo, o

termo da consulta “Tribunal” simplesmente define um conjunto de documentos que estão indexados com o termo “Tribunal”. Usando os operadores da lógica matemática booleana, os termos das consultas e seus correspondentes conjuntos de documento podem ser combinados para formar um novo conjunto de documentos, ou seja, os documentos são recuperados de acordo com a consulta definida com base em operadores lógicos. Assim, se a consulta for “Tribunal” E “Contas” E NÃO “União”, o sistema recuperará qualquer ocorrência de “Tribunal de Contas”, mas não retornará um documento que contenha, por exemplo, “Tribunal de Contas da União”.

Embora alternativas para o modelo booleano tenham sido propostas desde o fim da década de 1960, o modelo booleano foi o mais utilizado em sistema de recuperação comerciais até meados da década de 1990. Há duas principais razões para a predominância deste modelo. Primeiramente, o modelo permite ao usuário experiente ter uma sensação de controle sobre o que o sistema recupera, deste modo, está perfeitamente claro porque um documento foi recuperado, se o conjunto de documentos recuperados está muito pequeno ou muito grande, é perfeitamente claro quais operadores irão produzir respectivamente um conjunto maior ou menor de documentos. Em segundo lugar, o modelo pode ser estendido com operadores de proximidade e operadores coringa de uma maneira bem matemática, o que o torna também um poderoso candidato para sistemas de busca de texto completo. Outra razão, mais prática, dessa prevalência do modelo em sistemas comerciais são os custos de modificações maiores em softwares e estrutura de banco de dados e o fato de que a comunidade de usuários está treinada em modelos booleanos existentes [Rasmussen, 1999].

Para os usuários inexperientes, de um modo especial, o modelo tem muitas desvantagens. A principal desvantagem é que ele não fornece uma forma de atribuição da relevância de um documento recuperado. O modelo ou recupera um documento ou não recupera. Por exemplo, seja a consulta “Tribunal” E “Contas” E “União”, o sistema não recuperaria um documento indexado pelos termos “futebol”, “televisão” e “esporte”, mas também não recuperaria um documento indexado com com as palavras “Tribunal” e “Contas”. No entanto, é muito provável que o segundo documento seja mais relevante que o primeiro.

Uma outra desvantagem é que a rígida diferença entre os operadores booleanos E e OU não existe entre as palavras “e” e “ou” em linguagem natural. Por exemplo, alguém interessado em documento sobre auditoria e fiscalização, deveria fornecer a consulta “auditoria” OU “fiscalização”. Na verdade, o modelo booleano é mais complexo do que as reais necessidades dos usuários poderiam justificar.

## 2.3.O Modelo do Espaço Vetorial

No modelo de espaço vetorial [Salton, 1989], documentos e consultas são representados como vetores em um espaço p-dimensional:

$$\vec{D}=[d_1, d_2, ..., d_n]$$
$$\vec{Q}=[q_1, q_2, ..., q_n] \quad (1)$$

onde *m* é o número de dimensões do espaço, ou seja, no caso considerado, p é o número de palavras que serão consideradas na definição da orientação do vetor associado ao documento.

Cada posição no vetor comumente representa os termos do vocabulário pelos quais os documentos do corpus são indexados (ou seja, os distintos termos de um índice) e os valores $d_i$ ou $q_i$ são os pesos que estão associados ao i-ésimo termo no documento e na consulta, respectivamente, no espaço vetorial definido pelos *n* termos. Os pesos dos termos podem ser binários indicando a presença ou ausência do termo no documento ou na consulta. No modelo de espaço vetorial os pesos têm um valor numérico e indica a importância dos termos nos documentos ou na consulta. Por exemplo, pesos podem ser computados por um esquema de atribuição de pesos conhecido por tf-idf (do inglês, *term frequency-inverse document frequency*), onde o peso do termo é proporcional ao número de vezes em que o termo ocorre no documento ou consulta considerada (*tf*) e pode ser normalizado por um fator que representa o comprimento do texto, e onde *idf* é um fator que é inversamente proporcional ao número de documentos da coleção em que o termo ocorre.

A comparação do vetor que representa o documento e do vetor que representa a consulta é feita através do cálculo da similaridade ou distância entre os vetores. As funções de similaridade mais comuns são o produto interno entre dois vetores (Equação 2) e função cosseno do ângulo formado entre eles (Equação 3)

$$simil(\vec{d}, \vec{q}) = \sum_{k=1}^{n} d_k . q_k \quad (2)$$

$$simil(\vec{d}, \vec{q}) = \frac{\sum_{k=1}^{n} d_k . q_k}{\sqrt{\sum_{k=1}^{n} (d_k)^2} . \sqrt{\sum_{k=1}^{n} (q_k)^2}} \quad (3)$$

Assim, se consulta e documento não têm termos em comum tanto a similaridade calculada com base no produto interno quanto aquela calculada com base no cosseno entre os vetores resultará no valor zero. Senão retornará um valor diferente de zero. No caso do cálculo pelo cosseno entre vetores, o resultado é normalizado, e o valor máximo é 1. Na Figura 1.1 é possível visualizar um exemplo de vetor representando um documento e uma consulta no espaço vetorial $\Re^3$ gerado pelos três termos "Tribunal", "Contas" e "União", de onde é possível verificar que a consulta tem apenas as palavras "Tribunal" e "Contas", enquanto que o documento tem as palavras "Tribunal", "Contas" e "União".

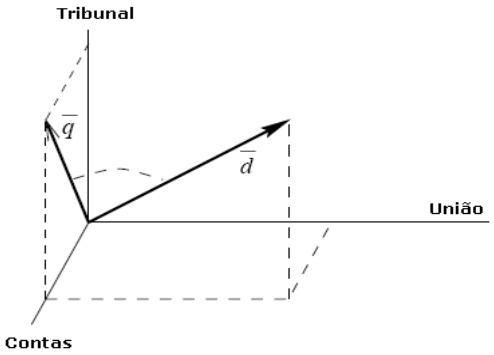



***Figura 1.1: Representação no modelo de espaço vetorial***

A principal desvantagem do modelo de espaço vetorial é que não há consenso sobre como atribuir os pesos que compõem os componentes dos vetores. Experimentos anteriores realizados por [Salton, 1971] já sugeriram que a atribuição de pesos não é um problema trivial. Uma segunda desvantagem é que não é possível incluir dependências entre os termos no modelo, por exemplo, para considerar frases ou termos adjacentes. É possível, entretanto, dar uma interpretação geométrica às consultas booleanas que vimos na Seção 2.2. Um terceiro problema é quanto à sua implementação. Um cálculo do cosseno entre vetores necessita dos valores de todos os componentes dos vetores, o que não está disponível em uma arquitetura que opera com índices invertidos. Dessa forma, Ou os pesos normalizados devem ser armazenados no índice invertido ou os valores necessários à normalização devem ser armazenados separadamente. Ambos tomariam muito mais espaço para armazenamento do que seria necessário no modelo booleano [Witten, Moffat and Beel, 1994]

## 2.4.A modelagem estatística da linguagem

Os documentos representam uma certa distribuição de informação que é indicado pela distribuição das palavras, mas também pela distribuição do conteúdo semântico que forma a informação. Em um modelo estatístico da linguagem, também conhecido, por modelo de linguagem (do inglês, *language model*), modela-se estatisticamente o conteúdo do documento. Nos últimos anos, a modelagem estatística da linguagem tem se tornado uma importante abordagem de modelagem da recuperação da informação [Croft & Lafferty, 2003]. Tipicamente, um documento é visto como um modelo e uma consulta como uma seqüência de texto extraída aleatoriamente deste modelo. A maioria das abordagens fazem o *ranking* dos documentos na coleção pela probabilidade de que a consulta $Q$ seja gerada dado o documento $D_j$ . No modelo de linguagem a consulta é vista como um conjunto de termos que são considerados como condicionalmente independentes dado o documento e, portanto, a probabilidade de ocorrência de uma consulta pode ser representada como um produto da probabilidade individual de ocorrência de cada um dos seus termos:

$$P\langle q_{1,\ldots}, q_m | D_j \rangle = \prod_{i=1}^{m} P\langle q_i | D_j \rangle \qquad (4)$$

onde $q_i$ é o $i-ésimo$ termo da consulta em uma consulta composta por $m$ termos, e $P\langle q_i|D_j\rangle$ é obtido pelo modelo de linguagem do documento. Computar a probabilidade de que um termo apareça no documento $D_j$ com a Equação 4 pode gerar uma probabilidade 0. Então, geralmente é escolhido um modelo que permita uma "suavização" do peso deste fator no cálculo da probabilidade $P\langle q_{1,\dots}, q_m|D_j\rangle$ . Frequentemente, a probabilidade de ocorrência de um termo no corpus é utilizada para "suavizar" as probabilidades geradas a partir do documento, produzindo o seguinte modelo composto (do inglês, *mixture model*):

$$P\langle q_{1,\dots}, q_m|D_j\rangle = \prod_{i=1}^{m} (\alpha P\langle q_i|D_j\rangle + (1-\alpha) P\langle q_i|C\rangle) \quad (5)$$

onde $C$ é a coleção de documentos. O peso para interpolação $\alpha$ é ajustado empiricamente ou ajustado a partir do treinamento com um corpus com julgamentos de relevância.

É possível projetar diversos modelos de linguagem que modelem probabilisticamente o conteúdo do documento com base nas palavras dos documentos e conceitos a elas associados. Por exemplo, [Cao et al., 2005] incorporou ao modelo os relacionamentos entre palavras obtidos a partir da ontologia mundial WordNet [Miller, 1990], o que é aplicável quando se deseja modelar um sistema de recuperação de aplicação geral, como um sistema de busca para a Web. No entanto, quando se tem um domínio restrito, esse modelo tende a apresentar resultados inferiores, uma vez que o grau de similaridade entre palavras varia de acordo com o domínio em que é utilizado. Por isso, como veremos adiante, dentro do desenvolvimento do presente trabalho, apresentamos um modelo para identificação das probabilidades dos termos da consulta dado um determinado documento, obtendo-se assim o modelo de linguagem associado àquele documento.

Este modelo tem a vantagem de considerar na recuperação de documentos não apenas os termos fornecidos na consulta, mas também todos aqueles que tenham alguma similaridade com o termo da consulta. Por se tratar de um modelo probabilístico, a atribuição de nível de relevância é direta e bem fundamentada, pois está diretamente associada à probabilidade de que o modelo de linguagem do documento considerado gere a consulta fornecida pelo usuário.

Por outro lado, o modelo tem algumas limitações. A premissa feita de que os termos da consulta são independentes pode ser equivocada, por exemplo, se o usuário apresenta a consulta "manga espada", a premissa de independência entre os termos leva o sistema a recuperar as similaridades dos termos dos documentos com o termo manga e com o termo espada, isoladamente, o que pode deteriorar a qualidade dos resultados. Nesse caso, a qualidade do resultado apresentado dependerá também do grau de especificidade da base pesquisada, já que, quanto mais especializado o domínio, mais os graus obtidos para similaridade entre palavras da consulta e do documento ficarão próximos dos conceitos empregados naquele domínio de conhecimento. No exemplo dado, em um domínio de

pesquisa agropecuária certamente haverá poucos ou nenhum documento que trate espada como uma arma branca, ou que trate manga como uma parte de uma peça de vestuário.

# Capítulo 3. A arquitetura do Terrier

Este capítulo mostra a arquitetura dos módulos de recuperação e de indexação do Terrier, a plataforma para pesquisa de modelos em recuperação da informação que é utilizada no presente trabalho. A seção 3.2 mostra a arquitetura do Indexador. A seção 3.2 mostra a arquitetura do Recuperador.

## 3.1. Introdução

Terrier, acrônimo utilizado para Terabyte Retriever, é uma plataforma modular, desenvolvida em java e com código aberto, para o rápido desenvolvimento de aplicações para recuperação da informação em larga escala. Ele é capaz de indexar vários tipos de coleções de documentos, incluindo coleções TREC e coleções WEB, sendo também utilizado para recuperação de informação corporativa ou ainda para recuperar informações em computadores pessoais. Para cada tipo de aplicação da plataforma, existe uma API própria com documentação disponível no sítio da ferramenta em http://ir.dcs.gla.ac.uk/terrier de onde também se pode baixar a última versão da ferramenta.

Esse projeto foi iniciado pela Universidade de Glasgow em 2000, com a finalidade de prover uma plataforma flexível para o desenvolvimento de aplicações em em RI, motivo pelo qual possui uma arquitetura "componentizada" e bastante configurável. Os dois principais componentes são o indexador e o recuperador.

No indexador está implementado todo o processo pelo qual o Terrier analisa uma coleção de documentos e armazena o seu conteúdo na forma de diversos índices que contém alguns dados estatísticos baseados na freqüência de ocorrência do termo em um documento e na coleção inteira. No recuperador, o Terrier aplica os pesos aos termos de acordo com os diversos modelos de recuperação da informação que estão implementados na plataforma e estima a relevância de um documento para uma determinada consulta.

## 3.2.O Indexador

Durante o processo de indexação, os termos passam por um *pipeline* de termos. Neste *pipeline* podem ser "encaixados" quaisquer tipos de processadores de texto que se desejar, bastando para isso implementar as interfaces apropriadas. Assim, nesta etapa os termos do documento podem ser pré-processados de diversas maneiras tais como para eliminação de stopwords, stemming, expansão de acrônimos e assim por diante. O resultado fornecido pelo *pipeline* é enviado de volta ao Indexador, que cria em disco as quatro principais estruturas do índice: o Léxico, o Índice Invertido, o Índice de Documentos e o Índice Direto.

O léxico armazena estatísticas globais acerca de cada termo que ocorre na coleção, ou seja, ele armazena o número de vezes que ele aparece e o número de documentos

diferentes em que ele aparece. Além disso, para facilitar a recuperação dos documentos, cada entrada no léxico contém um apontador para a entrada correspondente no índice invertido.

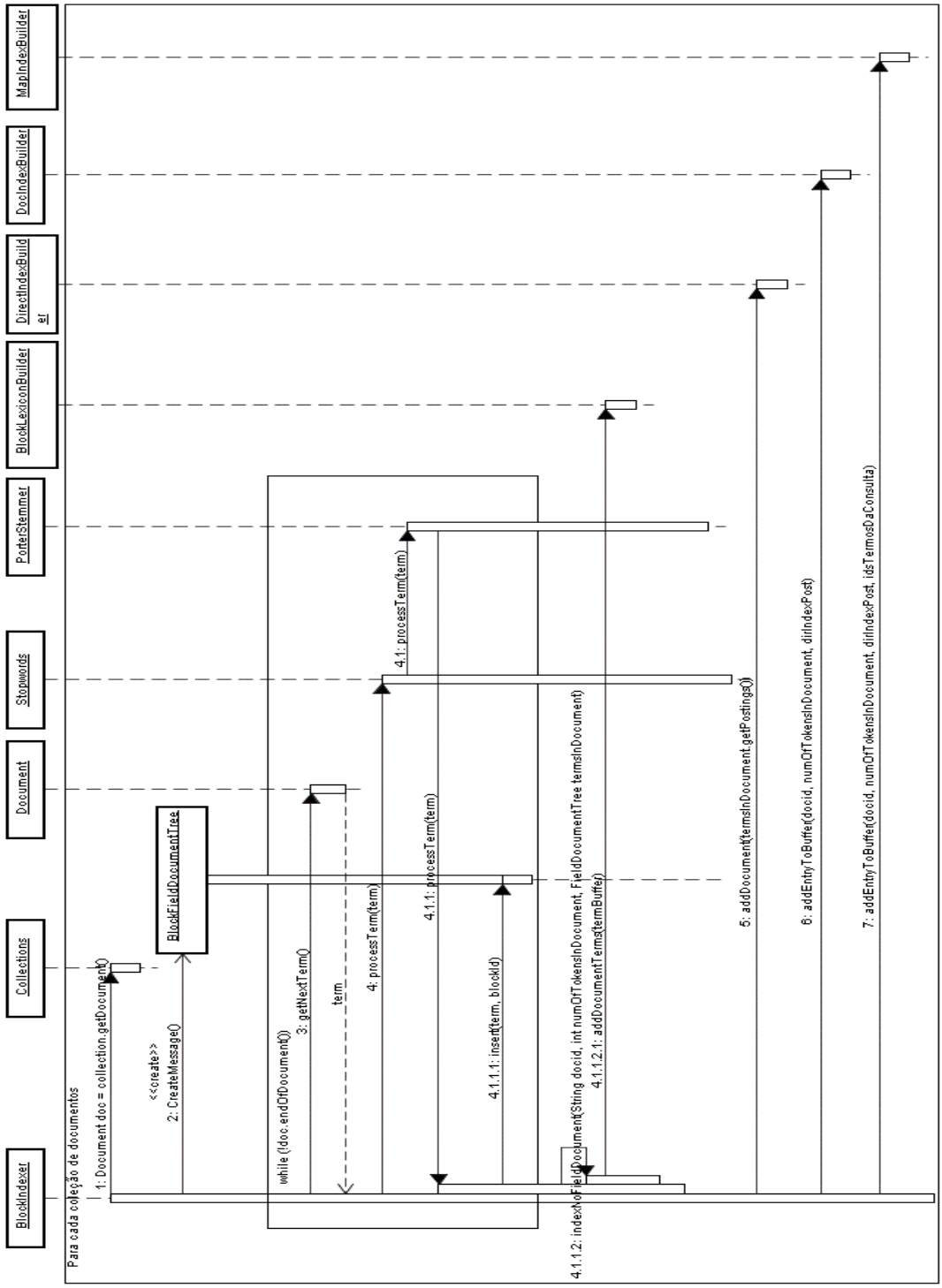



***Figura 2.1: Seqüência de eventos realizados durante a indexação de um documento no Terrier***

O índice invertido armazena uma lista para cada termo do léxico contendo os identificadores dos documentos em que aquele termo ocorre, bem como a sua freqüência de ocorrência nestes documentos. Opcionalmente, essa lista pode conter também a posição ou campos (por exemplo, a *tag* HTML) em que esse termo ocorre. Informações sobre a posição permite a realização de busca por proximidade ou de busca por frases.

O índice do documento armazena para cada documento, o seu comprimento e um ponteiro para a entrada correspondente no índice direto.

O índice direto armazena para cada documento, o identificador do termo e a sua freqüência naquele. Similarmente ao índice indireto, as posições de cada termo no documento também podem ser armazenadas.

As estruturas de dados descritas acima são altamente comprimidas, permitindo que uma grande coleção de documentos seja indexada com a utilização de pouco espaço em disco.

Na Figura 2.1 podemos visualizar a seqüência de eventos que ocorrem quando é realizada a indexação de um documento. Existem dois indexadores básicos: *BlockIndexer* e *BasicIndexer*. *BlockIndexer* faz indexação por blocos de informação, nesse caso a posição da palavra é indicada pelo número do bloco em que a informação se encontra, se o tamanho do bloco for igual a 1, então este indexador executa da mesma forma que o *BasicIndexer*, onde é atribuída a posição da palavra no texto, cada palavra correspondendo a um posição. Ambas as classes estendem a classe *Indexer*, a qual contém os métodos comuns a qualquer indexador, como criar índice direto, cria índice invertido, combinar índices, carregar o *pipeline*, etc.

O primeiro passo na indexação consiste em recuperar a coleção de documentos. Essa solicitação é feita a um objeto que implementa a interface *Collection*, que é basicamente um iterador sobre um conjunto de documentos. Ela gera objetos do tipo *Document* para cada nova requisição de documento que lhe é feita e é capaz de identificar o tipo de documento que está retornando. A versão mais recente da plataforma Terrier é distribuída, por padrão, já contendo coleções que manipulam uma pasta de documentos, uma coleção codificada em um arquivo xml e uma coleção no formato distribuído pela TREC.

Uma vez recuperado o documento, o Indexador passa a processá-lo. O processamento consiste em recuperar cada termo do documento e enviar ao *pipeline*. A forma como se dá a recuperação de cada termo depende da implementação da interface Document, por padrão a versão 2.0 do Terrier oferece a possibilidade de se trabalhar com documento de texto puro, HTML, MS Word, MS Excel, MS Power Point, e documentos no formato distribuído pela TREC.

A seguir o termo seguirá por tantos *pipelines* quantos estiverem descritos nos arquivos de configuração do Terrier. O principal arquivo de configuração do Terrier é chamado por padrão de *terrier.properties*, nele é possível configurar o *pipeline*, os modelos de recuperação de informação utilizado ou o esquema de atribuição de pesos do modelo selecionado, é possível ainda definir o tipo de indexador (*BlockIndexer* ou *BasicIndexer*, ou

ainda qualquer outro definido pelo usuário). Há também arquivos de configuração para identificar o local onde se encontram os documentos no sistema de arquivos, local onde se encontram as consultas a serem processadas em modo *batch*, ou os arquivos que contêm os documentos que devem ser recuperados com os respectivos níveis de relevância, dentre tantos outros itens configuráveis no Terrier.

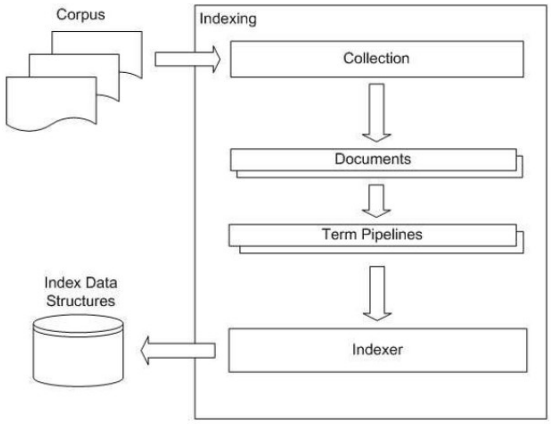



***Figura 2.2: Visão geral da arquitetura do módulo de indexação do Terrier. Um corpus de documentos é manipulado pelo componente Collection, que gera uma corrente (stream) de objetos Document. Cada um desses objetos gera uma corrente de termos, que são transformados por uma série de componentes Pipeline, após isso os índices são escritos no disco. Fonte [Ounis et al. 2006]***

No *pipeline* geralmente há filtros para eliminar palavras de pouca significação (*stop words*), e normalizar os termos reduzindo-os a seus radicais, processo conhecido como *stemming*. Os termos que não são excluídos pelo *pipeline* retornam ao Indexador que implementa a saída do *pipeline*, e é responsável por enviar o termo a uma estrutura de dados, a *BlockFieldDocumentTree*, que armazena o termo e o número do bloco onde ele ocorre, quando for o caso. Quando o indexador encontra o fim do documento processado, esta estrutura é enviada para a classe responsável pela construção do índice direto, *DirectIndexBuilder,* nesse momento também é enviado ao construtor de índice de documentos, *DocumentIndexBuilder,* uma solicitação para acrescentar uma entrada correspondente ao documento processado no índice de documentos, por meio de uma requisição contendo o identificador do documento, seu tamanho e uma referência para sua entrada no índice direto. Por fim, é enviada também uma requisição ao construtor do índice do Léxico, *LexiconBuilder*, para acrescentar as entradas referentes aos termos que ocorrem no documento corrente, se eles não existirem, ou atualizar a sua freqüência de ocorrência no corpus, acrescentando o número de ocorrências do termo naquele documento.

O índice invertido é criado, após o término do processamento de todos os documentos da coleção, por meio da inversão do índice direto.

## 3.3.O Recuperador

Nesta etapa, os pesos de relevância são atribuídos aos documentos de acordo com o modelo de recuperação utilizado.

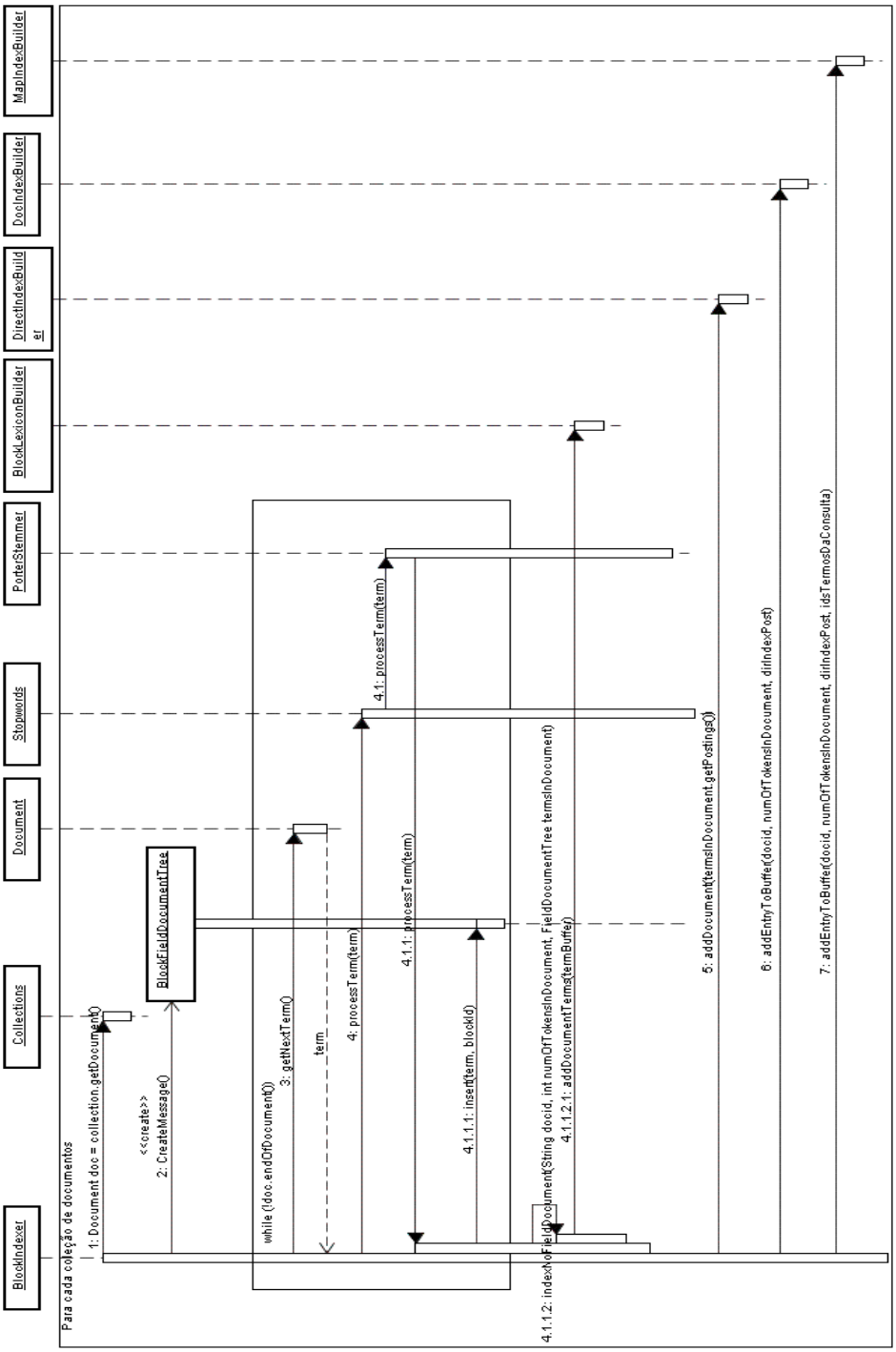



***Figura 2.3: Seqüência de eventos realizados durante a recuperação de um documento no Terrier***

De uma maneira geral, a cada termo da consulta é atribuído um peso, que representa a importância daquele termo no documento. Esses pesos são então utilizados para fazer o casamento (do inglês, *matching*) com os termos da consulta, e fazer a classificação dos documentos de acordo com essa relevância estimada do documento para a consulta. Terrier fornece uma gama de modelos de RI de atribuição de pesos, dentre eles podemos citar o modelo DFR, TF-IDF, Ponte-Croft´s Language Modelling e o modelo probabilístico Okapi's BM25.

Outra flexibilidade que esta etapa oferece é a possibilidade de que os níveis de relevância dos documentos recuperados possam ser alterados em qualquer estágio do processo de recuperação.

Na Figura 2.3 visualizamos os eventos que ocorrem durante a recuperação de documentos no Terrier. Quando uma consulta é apresentada ao sistema é instanciado um objeto que implementa a interface *SearchRequest,* o qual contém os detalhes que serão utilizados pelo mecanismo de busca para o processamento da consulta, esses detalhes incluem a consulta em si, o *ResultSet*, e os controles. Controles servem para ajustar parâmetros específicos do mecanismo de busca, ou para definir a necessidade de pré-filtros ou pós-filtros. Esses controles são, fundamentalmente, os nomes dos módulos que devem ser usados em cada estágio da busca:

- o modelo de recuperação da informação e o modelo de atribuição de pesos a ser usado.

- os módulos de pré e pós-processamento a serem aplicados a fim de modificar o *ResultSet*

O *Manager* coordena a ocorrência dos eventos de busca, ele é responsável por ativar o mecanismo de busca, de pré-processamento e de pós-processamento, encaminhando o objeto *SearchRequest* para cada um desses mecanismos. No *Matching* ocorre a recuperação da informação propriamente dita, pois é onde será calculada a relevância de cada documento para cada termo da consulta em *SearchRequest*. Para atribuir o valor de relevância de um documento para um termo de consulta específico, *Matching* instancia o modelo de atribuição de pesos que lhe foi informado também via *SearchRequest*. A forma como um modelo de atribuição de pesos calcula a relevância de um documento para um termo da consulta varia de modelo para modelo. Um *framework* de modelo de atribuição de pesos bastante utilizado, e amplamente implementado no Terrier, chamado *Divergence From Randomness* (DFR), efetua uma série de cálculos baseados na teoria das probabilidades para aferir a quantidade de informação que um termo pode ter associado em um determinado documento, basicamente, esses modelos consideram que quanto maior a divergência entre a freqüência do termo no documento e a freqüência do termo na coleção, maior a quantidade de informação que ele traz ao documento considerado.[Ounis et al. 2006]

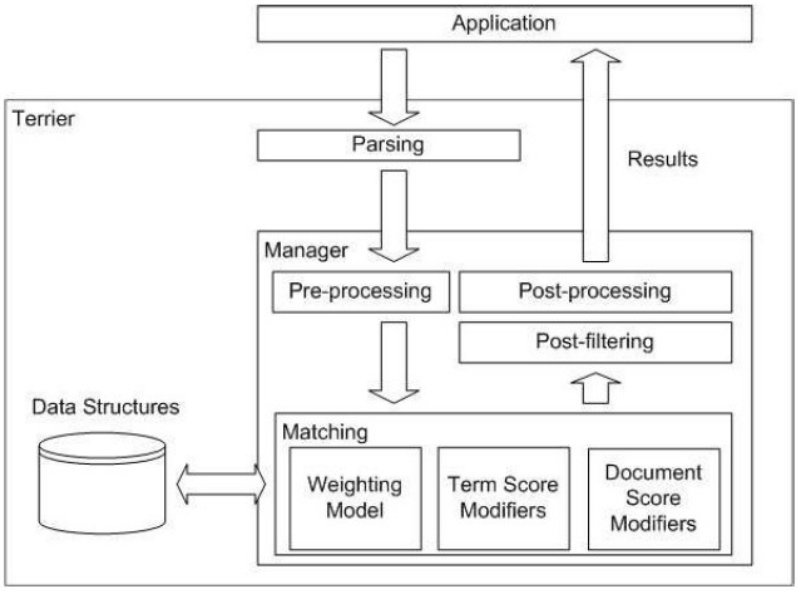



***Figura 2.4: Visão geral da arquitetura do módulo de recuperação do Terrier. A aplicação se comunica com o Manager, o qual, por sua vez, executa o módulo de Matching solicitado. Matching atribui níveis de relevância para os documentos utilizando uma combinação de modelos de atribuição de relevância e modificadores de relevância. Fonte: [Ounis et al. 2006]***

Após a atribuição de um nível de relevância de um documento para uma palavra da consulta, este pode ser alterado pela utilização de um objeto que implemente a interface *TermScoreModifier*. Por exemplo, um *TermFieldModifier* pode ser aplicado a fim de assegurar que os termos da consulta ocorrem em determinado campo do documento ou para atribuir uma relevância maior aos documentos quando os termos pesquisados aparecem nesses campos. A aplicação da expansão da consulta pode ser habilitada para se comportar de maneira diferente para cada consulta, a depender dos resultados recuperados na pré consulta. Um outro possível exemplo de pós-processamento poderia ser a aplicação de *clustering*.

De maneira similar, a modificação do nível de relevância de um documento recuperado, agora considerando o nível de relevância atribuído após processado todos os termos da consulta, pode ser feita pela aplicação de um *DocumentScoreModifier*. Um modificador desse tipo é *PhraseScoreModifier*, que emprega a informação de posição armazenada no índice do Terrier e zera a relevância de um documento recuperado no qual os termos da consulta não aparecem como uma frase, ou dentro de um determinado número de blocos. Geralmente modificadores do nível de relevância do documento são ideais para aplicar evidências que não dependem da consulta, como análise da estrutura de *hyperlinks* ou da URL dos documentos.

Após a eventual aplicação de qualquer desses modificadores, o conjunto de documentos recuperados pode ainda ser alterado pela aplicação de pós-processadores ou de pós-filtragem. Pós-processadores são apropriados para implementar funcionalidades que requerem uma modificação da consulta original. Um exemplo de pós-processador é a

expansão da consulta, pela seleção das palavras que ocorrem com mais frequência nos documentos recuperados segundo a consulta original, conforme observado no capítulo anterior.

A aplicação de pós-filtragem é a última etapa no processo de recuperação do Terrier, onde uma série de filtros pode remover arquivos já recuperados que não satisfazem a uma determinada condição. Por exemplo, no contexto da um mecanismo de busca da internet, a pós-filtragem poderia ser utilizado para reduzir o número de documentos recuperados de um mesmo sítio na internet, a fim de aumentar a diversidade dos resultados.

# Capítulo 4. Abordagem Proposta

Este capítulo apresenta um detalhamento da abordagem proposta. A seção 4.2 mostra a fundamentação matemática que nos leva à formulação do procedimento de EM no qual está baseado o modelo proposto. A seção 4.3 explica em detalhes como o algoritmo de EM funciona. A seção 4.4 trata sobre todas as medidas avaliação em sistemas de recuperação da informação utilizadas no presente trabalho.

## 4.1.Introdução

A utilização de modelos de linguagem para recuperação da informação tem sido amplamente pesquisada recentemente [Berger & Lafferty, 1999][Hiemstra & Kraaij, 1999][Miller et al.. 1999][Ponte & Croft, 1998] . Em [Berger & Lafferty, 1999], a relevância de um documento está diretamente relacionada com o grau de semelhança que as suas palavras têm com as palavras que compõem a consulta fornecida pelo usuário.

A idéia por trás dessa proposta é a de que o usuário, ao submeter uma consulta tem apenas uma noção daquilo que espera que o sistema de recuperação de informação lhe forneça. Assim a consulta fornecida pelo usuário traz uma idéia do contexto que deve ser localizado, de certo modo, isso indica que a procura por termos contextualmente semelhantes aos apresentados pelo usuário, ou seja, procuramos termos semelhantes àquele que foi fornecido pelo usuário dentro de um determinado contexto.

Segundo essa visão, a ocorrência de uma palavra literalmente igual à fornecida pelo usuário não significa, necessariamente, que aquele documento esteja sendo buscado pelo usuário, uma vez que aquela palavra pode estar sendo usada em um contexto (sentido) diferente daquele que foi imaginado pelo usuário. Do mesmo modo, uma palavra diferente daquela que foi fornecida na consulta pode ter sentido igual, indicando um maior potencial de que aquele documento seja relevante para o usuário.

A idéia básica é calcular a probabilidade $P\langle Q|D\rangle$ , ou seja, a probabilidade de gerar a consulta $Q$ dada a observação do documento $D$ . Na maioria das abordagens pesquisadas, este cálculo é decomposto em dois passos distintos: (1) A estimativa de um modelo de linguagem para o documento; (2) O cálculo da probabilidade da consulta baseado em algum modelo da consulta e usando o modelo de linguagem do documento estimado. Por exemplo, [Ponte & Croft, 1998] enfatizou o primeiro passo e usou várias heurísticas para suavizar a estimativa de máxima verossimilhança (do inglês, *Maximum Likelihood Estimate*, MLE) dos modelos de linguagem dos documentos e assumiu que as consultas eram geradas segundo um modelo de Bernoulli multivariado. O método BBN [Miller et al.. 1999] enfatiza o segundo passo e usa um modelo de cadeias de Markov de dois estados como base para a geração das consultas, o que, na verdade, suaviza a verossimilhança máxima com uma interpolação linear, uma estratégia também adotada por [Hiemstra & Kraaij, 1999]. Em [Zhai & Lafferty, 2001] foi verificado que a performance da recuperação é afetada tanto pela precisão da estimativa do modelo de linguagem do documento quanto da apropriada

modelagem da consulta, e um método de suavização de dois estágios foi sugerido para abordar explicitamente esses dois passos distintos.

Uma deficiência comum nessas abordagens é que todas aplicam o modelo de linguagem do documento estimado diretamente para gerar as consultas, mas consultas e documentos são gerados a partir de processos diferentes, já que possuem características bem diferentes. Assim, existe uma lacuna entre o modelo de linguagem do documento e o modelo de linguagem da consulta. Na maior parte dos casos essa lacuna é preenchida por suavização (*smoothing*), no entanto, seria ideal que pudéssemos estimar um modelo de linguagem da consulta e a partir da observação de um documento gerar uma consulta segundo esse modelo. Escolher a evidência a ser utilizada para estimar o modelo de linguagem das consultas é uma tarefa bastante desafiadora, já que a informação disponível nesse processo consiste apenas de uma coleção de documentos.

No presente trabalho, nós usamos as palavras-chaves e os títulos como evidências a partir das quais é gerado um modelo de tradução de documentos em consultas. A motivação é baseada na observação que consultas são mais próximas aos títulos ou às palavras chaves. Geralmente, um usuário ao apresentar uma consulta deseja fazer um resumo do que ele considera que é importante um documento ter, um trabalho semelhante ao que um autor realiza quando escolhe um título para o seu texto. De igual modo, as palavras chaves têm um modelo semelhante aquele de apresentação das consultas, qual seja o de identificar os principais assuntos contidos no documento.

No entanto, um documento pode conter diversos títulos, pois, cada pessoa, ao ler o documento, pode criar títulos diferentes, de acordo com suas impressões pessoais daquilo que é mais relevante no texto ou de acordo com as peculiaridades do seu vocabulário pessoal, uma vez que essa pessoa pode usar palavras do seu próprio vocabulário ao invés de usar palavras literalmente expressas no texto. Certamente, estes títulos são similares entre si e estão carregados da mesma semântica, mas não tem as mesmas palavras, o que leva a crer que um modelo que limite o conjunto de palavras para geração de consultas àquelas encontradas no título ou nas palavras-chave irá sofrer do problema de esparsidade de dados. Portanto, é necessária uma forma de "traduzir" as palavras que representam a consulta idealmente gerada para os documentos (título e palavras chaves criados pelo autor do documento) para as palavras apresentadas na consulta do usuário. Esse mecanismo de tradução é baseado na tradução estatística da linguagem [Brown et al.,1993], cujos detalhes e adaptação ao escopo da recuperação da informação é abordado na seção seguinte.

## 4.2.Fundamentação Matemática

Suponha que, dado um conjunto de documentos de um domínio específico, um especialista nesse domínio procure encontrar consultas que representem cada um desses documentos. Esse processo de geração de consultas pode ser entendido como um processo de tradução, onde o especialista "traduz" o documento em uma consulta.

Para realizar esta "tradução", o especialista escolherá aleatoriamente algumas palavras no texto que ele julgue ser uma representação do texto. Evidentemente, é improvável que o analista escolha várias ocorrências da mesma palavra. Nesse sentido, seria mais

intuitivo crer que se uma palavra ocorre com muita freqüência no texto $f\langle w|d\rangle$ , então é muito provável que ela ou alguma palavra próxima a ela seja escolhida pelo analista para fazer parte da consulta. Assim, a probabilidade de escolher uma consulta **q** como aquela que representa o documento **d** é:

$$p(\mathbf{q}|\mathbf{d}) = \sum_{w \in d} f(w|d)\sigma(q|w) \qquad (6)$$

Com essa abordagem, podemos encontrar semelhanças entre esse mecanismo de criação de consultas com o mecanismo de tradução de sentenças de uma língua para outra. Na tradução da linguagem, de igual sorte, o especialista (no domínio tradução de linguagens nesse caso) procura encontrar qual a palavra, ou conjunto de palavras tem mais probabilidade de ser a representação correta na linguagem destino da palavra na linguagem de origem. Nesse processo, o tradutor procura não somente encontrar uma tradução para aquela palavra na língua de origem, mas também qual é aquela palavra da língua de destino que mais se aplica ao contexto.

Para estabelecer um modelo que atenda tanto a mecanismos de tradução quanto ao de geração de consultas, é preciso, antes de tudo, entender como se dá o mecanismo de tradução e o de geração de consultas e quais são as semelhanças entre eles. Está evidente, pelo disposto até aqui, que várias palavras e/ou conjunto de palavras na linguagem destino podem ser uma tradução para uma palavra ou conjunto de palavras na linguagem de origem. Assim o trabalho consiste em encontrar a tradução mais provável. Desta forma, é intuitivo que algum formalismo probabilístico atenderá à tarefa proposta.

Peter Brown [Brown et al.,1993] apresenta um formalismo probabilístico no qual a tradução de uma sentença para uma linguagem de destino é encontrada automaticamente pelo treinamento do grau de relação entre as palavras de uma língua e de outra a partir de um conjunto de pareamento de sentenças nas duas línguas. Através da utilização do algoritmo de *Expectation Maximization* [Dempster, Laird & Rubin, 1977] são encontrados os graus de proximidade entre as palavras que atendem da melhor forma possível a todos os pareamentos de sentenças.

A partir do trabalho de Brown, [Berger & Lafferty, 1999] procura estabelecer uma metodologia análoga para o processo de recuperação da informação. A fim de encontrar uma consulta que representa um documento, imagine que um especialista siga os seguintes passos:

1. Escolha um comprimento m para a consulta, de acordo com o modelo de tamanho da amostra $\varphi(m|\mathbf{d})$.

2. Para cada posição $j \in [1...m]$ na consulta:

   2.1. Escolha uma palavra $d_i \in \mathbf{d}$ no documento a partir do qual desejamos gerar a próxima palavra da consulta;

2.2. Gere a próxima palavra da consulta por meio da "tradução" de $d_i$, ou seja, tomando-se uma palavra de acordo com a distribuição $\sigma(\cdot \mid d_i)$ do grau de proximidade entre as palavras do corpus e a palavra $d_i$ do documento.

Um alinhamento entre um documento e uma consulta é a identificação de qual ou quais palavras em uma consulta representa(m) uma palavra no documento. Assim, no passo 2.1 a palavra escolhida para a consulta estaria alinhada com a palavra $d_i$ do documento.

Usando o alinhamento $a$, a probabilidade de que uma consulta **q** represente um documento **d**, p(**q**|**d**) pode ser decomposta usando o teorema da probabilidade total da seguinte forma:

$$p(\mathbf{q}|\mathbf{d}) = \sum_a p(\mathbf{q}, a|\mathbf{d}) = \sum_a p(\mathbf{q}|a,\mathbf{d})\, p(a|\mathbf{d}) \qquad (7)$$

Ou seja, a probabilidade de que uma consulta **q** represente um documento **d** é dada pela somatória das probabilidades de **q** em cada um dos seus possíveis alinhamentos $a$.

Se cada palavra pudesse fazer parte de apenas uma das consultas **q** presente na coleção de pareamentos (**q**,**d**), então:

$$p(\mathbf{q}, a|\mathbf{d}) = \prod_{i=1}^{m} \sigma(q_i \mid d_{a_i}) \qquad (8)$$

Onde $d_{a_i}$ é a palavra do documento que está alinhada com a $i$-ésima palavra da consulta $q_i$ e $\sigma(q_i \mid d_{a_i})$ é a probabilidade de que $q_i$ esteja alinhada com $d_{a_i}$.

Podemos incluir nos documentos também uma palavra nula. A finalidade da palavra nula é gerar na consulta palavras espúrias ou livres do contexto, o que permitiria consultas do tipo: "Encontre todos os documentos".

Existem $(n+1)^m$ alinhamentos possíveis, pois cada uma das $(n+1)$ palavras do documento podem ser alinhadas com cada uma das $m$ palavras da consulta:

$$p(\mathbf{q}|\mathbf{d}) = \frac{p(m \mid d)}{(n+1)^m} \sum_a \prod_{i=1}^{m} \sigma(q_i \mid d_{a_i}) \qquad (9)$$

onde $p(m \mid d)$ é a probabilidade de que a consulta tenha tamanho $m$ dado o documento **d**.

Dada uma coleção de pares de consulta/documento $C = \{(\mathbf{q}^1,\mathbf{d}^1),(\mathbf{q}^2,\mathbf{d}^2),...,(\mathbf{q}^s,\mathbf{d}^s)\}$ queremos ajustar os parâmetros $\sigma(q_i \mid d_{a_i})$ da equação (9) a fim de maximizar a probabilidade da ocorrência de $C$. No entanto esta maximização está sujeita a seguinte restrição:

$\sum_{q} (q \mid d) = 1$ onde $q$ é uma palavra em **q**, $d$ é uma palavra em **d**$\in C$ (10)

uma vez que o conjunto de palavras $q$ é exaustivo. Usando multiplicadores de Lagrange para maximizar (9), sujeita a restrição estabelecida em (10) vem:

$$\sigma (q \mid d) = \lambda^{-1} \sum_{a} \quad \mathrm{p}(\mathbf{q}, a \mid \mathbf{d}) \sum_{i=1}^{m} \delta (q, q_i) \delta (d, d_{a_i}) \tag{11}$$

onde $\delta$ é a função de Kronecker, cujo resultado é 1 se os seus dois argumentos são iguais e 0 se os dois argumentos são diferentes. Assim, $\sum_{i=1}^{m} \delta (q, q_i) \delta (d, d_{a_i})$ é a quantidade de vezes em que $q$ está alinhada a $d$ no documento **d** para o alinhamento $a$ Se considerarmos a probabilidade da ocorrência do alinhamento $a$, chamamos o número esperado de vezes que $q$ se conecta a $d$ no pareamento (**q**,**d**) de contador de $q$ dado $d \in$ **d**, e o denominamos $c(q \mid d;$ **q,d**):

$$c(q \mid d; \mathbf{q},\mathbf{d}) = \sum_{a} p(a \mid \mathbf{d},\mathbf{q}) \sum_{i=1}^{m} \delta (q, q_i) \delta (d, d_{a_i}) \tag{12}$$

onde:

$$\mathrm{p}(\mathrm{a}|\mathbf{d},\mathbf{q}) = \frac{\mathrm{p}(\mathbf{q},\mathrm{a} \mid \mathbf{d})}{\mathrm{p}(\mathbf{q},\mathbf{d})} \tag{13}$$

Como $\lambda$ é uma parâmetro que iremos calcular dentro do procedimento de EM, podemos fazer $\lambda^{-1} = \lambda^{-1} \cdot \mathrm{p}(\mathbf{q}|\mathbf{d})$, já que um parâmetro multiplicado por uma constante continua sendo um parâmetro. Pela substituição de (13) em (12), e do resultado desta substituição em (11) vem:

$$\sigma (q \mid d) = \lambda^{-1} c(q \mid d; \mathbf{q},\mathbf{d}) \tag{14}$$

Na prática, nosso conjunto de treinamento consiste de um conjunto de pareamentos $\{(\mathbf{q}^1,\mathbf{d}^1),\mathbf{q}^2,\mathbf{d}^2),...,\mathbf{q}^S,\mathbf{d}^S)\}$, então a equação (14) assume a forma:

$$\sigma (q \mid d) = \lambda^{-1} \sum_{s=1}^{S} \quad c(q \mid d; \mathbf{q}^S,\mathbf{d}^S) \tag{15}$$

Pela avaliação direta da equação (9) podemos verificar que:

$$\sum_{a_1=0}^{l} \ldots \sum_{a_m=0}^{l} \prod_{i=1}^{m} (q_i \mid d_{a_i}) = \prod_{j=1}^{m} \sum_{a_j=0}^{l} \sigma\,(q_i \mid d_{a_i}) \tag{16}$$

Assim, podemos trocar as somas com os produtos na equação (9) para obter:

$$\mathrm{p}(\mathbf{q}|\mathbf{d}) = \frac{p(m \mid d)}{(n+1)^m} \prod_{i=1}^{m} \sum_{a_i=0}^{l} \sigma\,(q_i \mid d_{a_i}) \tag{17}$$

Então, obtendo novamente o contador de $q$ dado $d \in \mathbf{d}$ segundo a equação (17) vem:

$$c(q \mid d; \mathbf{q},\mathbf{d}) = \frac{\sigma\,(q \mid d)}{\sigma\,(q \mid d_1) + \ldots. + \sigma\,(q \mid d_l)} \cdot \sum_{j=1}^{m} (q, q_j) \sum_{i=0}^{l} (d, d_i) \tag{18}$$

onde $\sigma\,(q \mid d_l)$ é o número de vezes em que $q$ aparece alinhado a $d_l$ no pareamento $\mathbf{q}^S$,$\mathbf{d}^S$) dividido pelo número de vezes que $d_l$ aparece em **d**, ou seja, é a freqüência relativa dos alinhamentos de $q$ a $d_l$ em **d.**

Assim, o algoritmo de EM deve executar os seguintes passos para encontrar o grau de proximidade $\sigma\,(q \mid d)$:

1. Escolha valores iniciais para $\sigma\,(q \mid d)$

2. Para cada par de consulta/documento em $C = \{(\mathbf{q}^1,\mathbf{d}^1),(\mathbf{q}^2,\mathbf{d}^2),\ldots,(\mathbf{q}^s,\mathbf{d}^s)\}$ use a equação (18) para calcular $c(q \mid d; \mathbf{q},\mathbf{d})$

3. Para cada $d$ que aparece pelo menos uma vez em $\mathbf{d}^S$ faça:

   3.1. Calcule $\lambda$ de acordo com a equação:

$$\lambda = \sum_{q} \sum_{s=1}^{S} c(q \mid d; \mathbf{q}^s,\mathbf{d}^s)$$

   3.2. Para cada $q$ que aparece pelo menos uma vez em $\mathbf{q}^s$, use a equação (15) para obter um novo valor de $\sigma\,(q \mid d)$

4. Repita os passos 2 e 3 até que os valores de $\sigma(q|d)$ tenham convergido a um grau desejável.

Desta sorte, este algoritmo considerará, para o cálculo de $\sigma(q|d)$, as palavras da consulta $\mathbf{q}^s$, associada ao documento $\mathbf{d}^s$ onde $d$ ocorre. Ou seja, o parâmetro $\lambda$ depende do número esperado de vezes em que $d$ se conecta a cada uma das palavras $q$ em cada um dos pareamentos em $C$.

Desta maneira, podemos fixar $d$, e encontrar $\sigma(q|d)$ para cada uma das palavras $q$ que ocorrem na consulta. Tal abordagem se mostra interessante, uma vez que não poderemos considerar todas as combinações entre $q$ e $d$ possíveis. Com esta abordagem, ficamos restritos às palavras da consulta que possuem maior proximidade com a palavra do documento, o que reduzirá substancialmente o número de cálculos necessários para atribuir a relevância de um documento para a consulta realizada .

Ao se deparar com a fundamentação matemática exposta na seção anterior, a questão que naturalmente surge é como encontrar uma coleção de pares de consulta/documento $C = \{(\mathbf{q}^1,\mathbf{d}^1),(\mathbf{q}^2,\mathbf{d}^2),...,(\mathbf{q}^s,\mathbf{d}^s)\}$ que nos permita encontrar um grau de proximidade entre as palavras, ingrediente fundamental para a metodologia proposta.

Como enfatizado no início desse capítulo, devemos escolher um modelo de linguagem da consulta. Assim, as consultas $\mathbf{q}^1,\mathbf{q}^2,...,\mathbf{q}^s$ serão formadas pela obtenção dos títulos e palavras chaves dos documentos $\mathbf{d}^1,\mathbf{d}^2,...,\mathbf{d}^s$ a serem indexados.

Após determinada a coleção de pares de consultas e documentos $C$, o grau de proximidade $\sigma(q|d)$ é encontrado segundo o procedimento descrito na seção anterior.

Terminada a fase de determinação dos parâmetros do nosso modelo, passamos a uma versão inicial do algoritmo de recuperação de documentos:

***Algoritmo 1**: Recuperação básica de documentos*

---

*Entrada:* Consulta $\mathbf{q}=\{q_1,q_2,...,q_m\}$ ;
Coleção de documentos $D = \{d_1,d_2,...,d_n\}$ ;
Grau de proximidade entre palavras $\sigma(q|d)$ para todo par $(q,d)$

*Saída:* Nível de relevância $\rho_q(\mathbf{d})$ para cada documento **d**

1. Para cada documento $d \in D$ na coleção faça:
   2. Atribua $\rho_q(\mathbf{d}) \leftarrow 1$
   3. Para cada palavra da consulta $q \in \mathbf{q}$ faça:
      4. Calcule $p(\mathbf{q}|\mathbf{d})$ de acordo com a equação (6)
      5. Atribua $\rho_q(\mathbf{d}) \leftarrow \rho_q(\mathbf{d}) \times p(\mathbf{q}|\mathbf{d})$

---

Uma estrutura de dados tipicamente utilizada em sistemas de recuperação da informação é o índice invertido. Cada palavra que ocorre no texto tem uma entrada no índice, esta entrada está mapeada para todos os documentos onde a palavra ocorre. Geralmente, os sistemas de recuperação da informação fazem a busca exata dos termos existente na consulta, nesse sentido, a recuperação dos documentos se dá pela busca direta no índice invertido desses mesmos termos, o que resulta em um tempo de busca bastante reduzido, pois é proporcional ao tamanho da consulta $|\mathbf{q}|$.

No entanto, como já bem destacado até aqui, o índice de revocação desse tipo de consulta não é apropriado, pois um documento que contenha palavras com significado próximos àquelas que foram utilizadas pelo usuário podem ser tão ou mais relevantes do que outros que contenham algumas palavras idênticas àquelas que foram utilizadas pelo usuário.

A técnica de expansão de consultas procura contornar esse problema expandindo a consulta original com os termos encontrados nos documentos recuperados e refazendo a consulta, o que, apesar de fazer a consulta duas vezes, ainda se demonstra rápida, uma vez que a consulta a uma entrada do índice continua sendo feita pela palavra exata e não por palavras próximas.

Quando se procura por termos próximos aos termos apresentados pelo usuário, um índice invertido, na forma usual, não atende ao propósito, uma vez que não há como derivar relação de proximidade entre as palavras a partir da consulta ao índice. A alternativa proposta até agora, qual seja a de atribuir o grau de relevância de um determinado documento de acordo com o grau de proximidade de seus termos e dos termos da consulta, requer um tempo proporcional a $|\mathbf{q}| \times |\mathbf{d}|$: o produto do tamanho da consulta pelo tamanho do documento, o que torna essa abordagem impraticável.

Uma alternativa para esse problema deriva da constatação de que o grau de relevância de um documento para uma determinada palavra da consulta não precisa ser calculado no momento da consulta, ele pode ser pré-computado. Ou seja, para cada palavra possível de ocorrer na consulta $q$ e para cada documento $\mathbf{d} \in D$ é calculado o valor de $p(q|\mathbf{d})$ o qual é armazenado em uma matriz $I$.

Pré-calcular as células de $I$ e então usar esses valores no Algoritmo 1 reduz o tempo da consulta de $|D| \times |\mathbf{q}| \times |\mathbf{d}|$ para $|D| \times |\mathbf{q}|$. Infelizmente, torna-se proibitivo calcular e armazenar a matriz $I$, com tantas colunas quantos documentos na coleção e tantas linha quantas palavras existentes na coleção de documentos $C$. Uma coleção de 100000 documentos com um vocabulário de 100000 palavras requer uma matriz de 400 Gb. Então poderíamos fazer a seguinte aproximação da equação (1):

$$p(\mathbf{q}|\mathbf{d}) = \sum_{w \in \tau^n} f\langle w|d\rangle \sigma\langle q|w\rangle \qquad (19)$$

Onde $\tau^n(q) \stackrel{def}{=} w : \sigma(q \mid w)$ está entre os n maiores valores de $\sigma$ qualquer que seja $w$

Ou seja, $\tau^n(q)$ é o conjunto das $n$ palavras mais similares a $q$ em um determinado documento. O valor de $n$ será selecionado experimentalmente. Essa aproximação resulta que muitos dos valores de $p(q \mid \mathbf{d})$ serão reduzidos a zero, tornando $I$ uma matriz esparsa, facilmente armazenável usando técnicas de armazenagem de matrizes esparsas.

Assim, o Algoritmo 1 com as melhorias de performance sugeridas até aqui resulta no seguinte algoritmo:

***Algoritmo 2**: Recuperação otimizada de documentos*

---

*Entrada:* Consulta **q=** $\{q_1, q_2, ..., q_m\}$ ;
Coleção de documentos $D = \{d_1, d_2, ..., d_n\}$ ;
Grau de proximidade entre palavras $\sigma(q \mid d)$ para todo par $(q, d)$
Matriz $I$ de mapeamentos entre palavras $q$ e documentos **d**

*Saída:* Nível de relevância $\rho_q$ (**d**) para cada documento **d**

1. Para cada documento $d \in I$ (**q**) na coleção faça:
1. Atribua $\rho_q$ (**d**)$\leftarrow$ 1
2. Para cada palavra da consulta $q \in$ **q** faça:
   - Calcule $P$ (**q**|**d**) de acordo com a equação (19) (valor pré-computado)
   - Atribua $\rho_q$ (**d**)$\leftarrow$ $\rho_q$ (**d**) $\times$ $P$ (**q**|**d**)

---

## 4.3.Cálculo da similaridade entre palavras

O procedimento de EM descrito na seção 4.2 foi implementado dentro da classe *Index* do terrier. Assim, após a criação dos índices mencionados na seção 3.2 foram criados também dois outros índices: *MapIndex* e *Similaridades*. Além desses índices, o resultado do algoritmo de EM é uma matriz de similaridades entre as palavras que é utilizada como estrutura de dados intermediária para criação do índice de Similaridades entre palavras da consulta e palavras do documento.

O *MapIndex* é um índice que armazena, para cada documento da coleção, os identificadores dos termos que fazem parte da consulta associada. A consulta associada ao documento, como mostrado nas seções 4.1 e 4.2, pode ser entendida como a tradução do modelo de linguagem do documento no modelo de linguagem da consulta. Assim, as consultas obtidas dessa forma funcionam como um identificador do documento, ou seja, é a forma pela qual o especialista, no caso, o autor, descreve o seu documento, e pode ser entendida como a "consulta ideal", ou seja, aquela que melhor descreve o documento. Mais

ainda, esta consulta está sujeita ao modelo de linguagem da consulta, não obstante ser um identificador do documento, pois o especialista ao criar o título e ao selecionar as palavras chaves atende a um modelo específico, intuitivo, segundo o qual as palavras que compõem o título, por exemplo, não são necessariamente aquelas que mais ocorrem no texto e também podem não ser aquelas que têm maior frequência relativa no texto. Em vez disso, elas são escolhidas segundo o grau de proximidade entre a sua semântica e a semântica do texto, ou a mensagem que o texto deseja passar.

A definição de *stop-words*, por exemplo, não é específica o suficiente para garantir que palavras muito freqüentes em um documento deveriam ter sido consideradas como *stop-words* e não o foram. A palavra "lei", por exemplo, certamente não está entre as *stop-words* da língua portuguesa, no entanto, em determinados domínios, como a pesquisa jurisprudencial, a palavra "lei" isolada não diferencia os documentos. Uma possível solução para esse problema seria considerar a freqüência relativa, no entanto, experimentos realizados no presente trabalho com o corpus em língua portuguesa do CETEN-FOLHA [Linguateca] mostrou que a utilização da frequência relativa para criação da consulta associada aos documentos, acaba por escolher palavras que, em virtude da sua baixa frequência em todo o corpus, possui frequência alta nos poucos documentos em que ocorre. A título de exemplo, a palavra "esmeradamente" ocorre apenas uma única vez em todo o corpus, ou seja, no documento em que ela ocorrer terá frequência relativa igual a 1, e certamente será escolhida para formar a consulta associada àquele documento, já que possui frequência relativa máxima.

A definição de consultas associadas a partir do título e de palavras chaves do texto procura superar esses problemas ao mesmo tempo em que tenta capturar o modelo implícito, resultante da perícia do especialista humano, utilizado para gerar os títulos e para escolher as palavras chaves.

O índice de Similaridades é uma matriz esparsa que contém o grau de similaridade entre as palavras da consulta escolhidas conforme descrito acima e os documentos que compõem a coleção. Se imaginarmos esse processo de passagem do modelo de linguagem do documento para o modelo de linguagem da consulta como uma tradução entre modelos podemos fazer uma analogia ao processo de tradução automática da linguagem humana. Um falante da língua portuguesa segue um modelo ao pronunciar sentenças nessa língua. Apesar de ser um modelo intuitivo e até hoje nunca completamente descrito apesar dos esforços dos pesquisadores, é certo que existe um critério para formação de sentenças em uma língua. Nenhum falante da língua portuguesa vai esperar ser entendido ao pronunciar a seqüência de palavras "estilo casa cadeira" ou "mais menos ou". Embora, seja possível definir algumas regras para impedir a criação de "frases" como essas, identificar um modelo que descreva completamente a forma pela qual toda e qualquer sentença é construída é ainda uma tarefa sem solução

No entanto, é possível aprender um modelo que se aproxime ou que se adapte a um conjunto de sentenças gerado em uma linguagem. Desta sorte, se possuímos um conjunto de sentenças em língua portuguesa e outro em língua inglesa, e um mapeamento de sentenças em língua para sentenças em outra língua, então podemos aprender um modelo que se ajuste a esses mapeamentos.

Esse trabalho foi apresentado por [Brown et al.,1993] e consiste nos fundamentos utilizados para obtenção dos graus de similaridade na "tradução" que fazemos entre documentos e consultas. O que desejamos é estimar a probabilidade de "tradução" $\sigma\langle q|d\rangle$ de um corpus paralelo, ou seja, com consultas alinhadas a documentos. No entanto, nós não possuímos os alinhamentos, pois se tivéssemos os alinhamentos, nós poderíamos estimar os parâmetros do modelo, ou se tivéssemos os parâmetros do modelo nós podíamos estimar os alinhamentos: o problema do ovo e da galinha!

O algoritmo de *Expectation Maximization* [Dempster, Laird & Rubin, 1977] é aplicável nesse caso, pois, através dele, se temos dados incompletos podemos estimar o modelo, e se temos o modelo podemos estimar os dados que estão faltando. Em resumo, o procedimento de EM consistirá nas seguintes etapas:

1. Inicialize os parâmetros do modelo (de maneira uniforme, por exemplo)
2. Estabeleça estimativas para os dados faltantes
3. Construa o modelo segundo os parâmetros estimados
4. Repita os passos 1 a 4, até que os valores estimados convirjam

[Brown et al.,1993] mostrou que o algoritmo de EM para tradução estatística da linguagem, em algum momento, converge e, mais ainda, como a função que descreve o modelo de tradução tem um único ponto de máximo, o algoritmo sempre converge para um máximo global.

A Figura 4.1 mostra os passos seguidos pelo algoritmo de EM no domínio da tradução estatística da linguagem. Nas figuras temos um par de sentenças, uma em inglês e outra em português, para os quais desejamos descobrir os alinhamentos entre as palavras. A medida que o algoritmo itera os alinhamentos que reforçam o modelo de tradução proposto são reforçados, até a convergência.

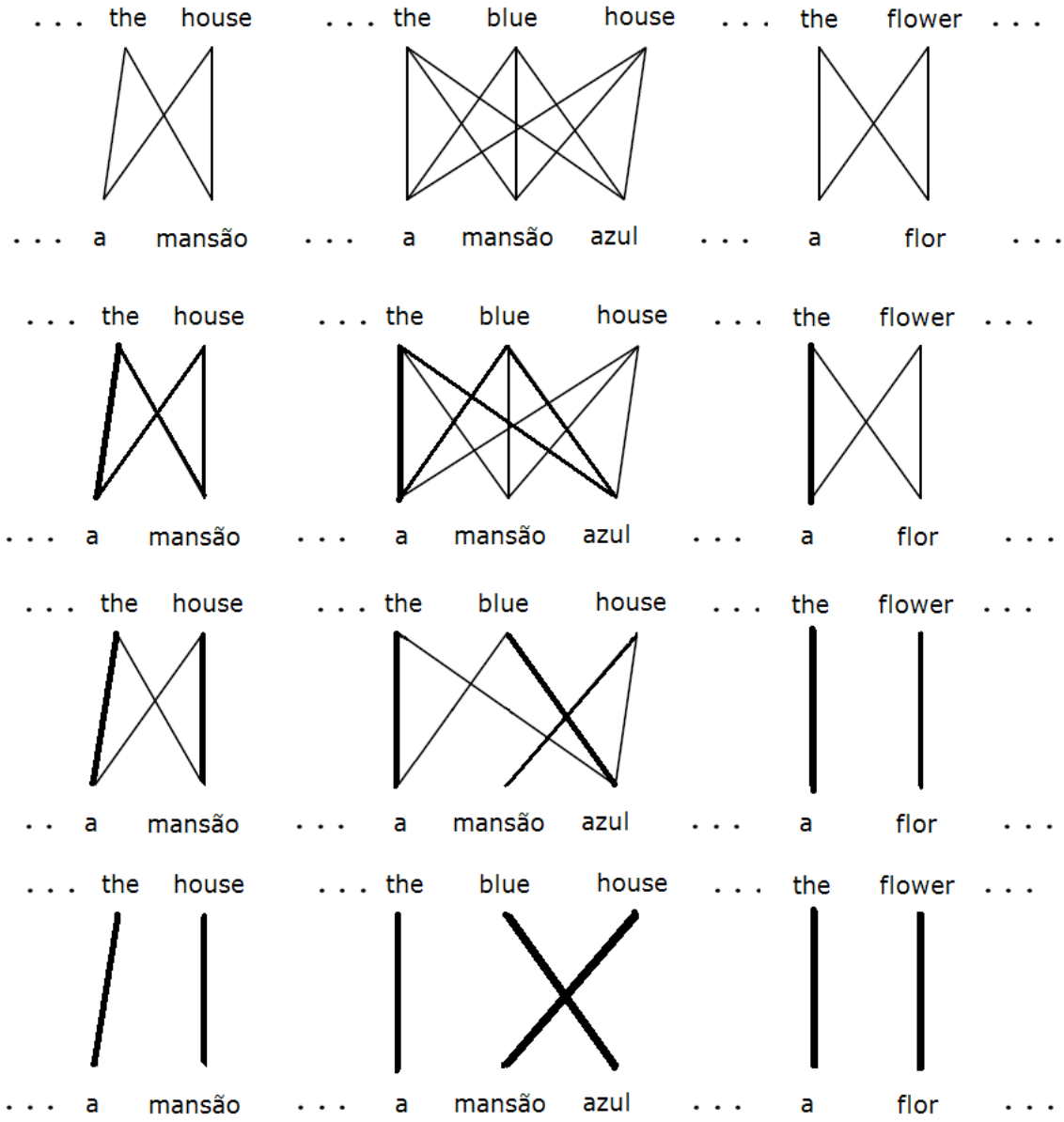



***Figura 4.1: Quatro iterações do algoritmo de EM: os alinhamentos são reforçados conforme o algoritmo vai aprendendo o modelo por meio da co-ocorrência das palavras***

A fundamentação matemática dessas abordagens e da sua adaptação para o contexto da recuperação da informação se encontram descritos na seção 4.2. Aqui pretendemos recuperar o contexto da obtenção de graus de similaridade via EM a fim de construirmos um pseudocódigo para o mesmo, o qual será implementado dentro do módulo de indexação do Terrier.

Analisando os passos 2 e 3 do procedimento de obtenção dos graus de similaridade entre termos da consulta e $q$ e termos dos documentos $d$ , descrito na seção 4.2 vemos a obtenção dos contadores $c(q \mid d; \mathbf{q},\mathbf{d})$ é a principal etapa na execução do algoritmo de EM. A

finalidade do contador é utilizar o grau de similaridade entre $q$ e $d$ estimado pelo algoritmo de EM no passo anterior e a contagem da suas co-ocorrências nos documentos com a ponderação da sua freqüência de ocorrência no respectivo documento, a fim de obter uma nova estimativa para o grau de similaridade entre esses termos.

No entanto a idéia de similaridade entre termos deve considerar não somente a frequência com que esses termos co-ocorrem, mas também a frequência em que $d$ co-ocorre com outro termo da consulta, pois a co-ocorrência entre $q$ e $d$ será tanto mais representativa da similaridade entre esses termos quanto mais ele estiver próximo da quantidade total de vezes em que $d$ ocorre, ou seja, a similaridade entre $q$ e $d$ depende da similaridade de $d$ com os outros termos da consulta, uma vez que esta similaridade está sujeita a restrição estabelecida na equação 10.

Portanto, do ponto de vista da implementação, necessitamos a criação de uma variável temporária para acumular o valor da similaridade que $d$ possui com outros termos da consulta, à medida que percorremos a coleção de documentos. Esta é a variável *temp* que aparece no pseudocódigo do Algoritmo 3.

Uma vez calculado o denominador da equação 18 através da variável *temp* podemos calcular o valor do contador, conforme descrito no passo 2.a.i do Algoritmo 3, onde *temp* é o denominador utilizado para o cálculo do contador.

A próxima etapa, segundo o passo 3.1 do procedimento descrito na seção 4.2 é calcular o valor de $\lambda$. Pela análise da somatória ali descrita, percebemos que $\lambda$ é a somatória para todos os termos da consulta da somatória para todos os pareamentos $\langle q|d \rangle$ dos contadores. Ora, como descrito acima, somatória para todos os pareamentos $\langle q|d \rangle$ dos contadores é o valor que acabamos de armazenar na variável *temp,* portanto para calcularmos o valor $\lambda$ basta iterarmos sobre as palavras da consulta, acumulando o valor obtido para a variável *totalContador* em $\lambda$, conforme descrito no passo 2.a.ii do Algoritmo 3. O valor de $\sigma\langle q|d \rangle$ é então obtido pela razão entre o *totalContador e* $\lambda$ conforme descrito na equação 15.

O pseudocódigo deve ser construído segundo as limitações que impõem uma linguagem de programação, por isso, foi necessária essa análise detalhada da seqüência de passos do procedimento de EM. A necessidade da criação da variável *temp* é um exemplo dessa limitação, pois no momento em que estamos iterando sobre palavras da consulta e palavras do documento, necessitamos saber o grau de similaridade do termo da consulta que estamos considerando naquela iteração com todos os termos dos documentos, não somente com aquele termo do documento considerado naquela iteração, daí a necessidade de percorrer primeiro todos os mapeamentos e colocar essas similaridades em uma matriz temporária.

Mas não são somente essas limitações que devem ser consideradas no momento da construção do código, é necessário fazer considerações sobre a utilização de memória e o tempo de processamento. As primeiras implementações deste algoritmo ficou bastante limitado pela memória de 2 GB do computador onde estavam sendo feitos os testes, não sendo possível processar mais que 1000 documentos naquele momento. Como as matrizes

eram mantidas em memória, a solução inicial foi tentar reduzir o número de similaridades a serem calculadas e armazenadas, pois percebeu-se, pelo exame um pouco mais atento do procedimento de EM descrito na seção 4.2 que somente ira ter grau de similaridade maior que zero, termos que co-ocorrem nos pareamentos consulta/documento, como se pode perceber na equação 12, pois, de outra maneira, o contador é igual a zero. No entanto, essa otimização, embora tenha melhorado o consumo de memória, não resolveu o problema: a memória continuava sendo o limitante para processar bases maiores. A pergunta que naturalmente surge é: por que manter as matrizes em memória ao invés de armazená-la em disco? O impedimento para essa alternativa é que o acesso, tanto de leitura quanto de escrita, se dava de maneira aleatória, segundo pode ser observado no pseudocódigo proposto acima.

***Algoritmo 3**: Estimativa do grau de similaridade entre palavras via Expectation Maximization*

---

*Entrada:* Corpus paralelo $MP = \{(\mathbf{q}^1,\mathbf{d}^1),(\mathbf{q}^2,\mathbf{d}^2),...,(\mathbf{q}^s,\mathbf{d}^s)\}$

Estimativa inicial uniforme de similaridades entre as palavras da consulta $\sigma(q \mid d)$ para todos os pares $(q,d)$

*Saída:* Grau de similaridade $\sigma(q \mid d)$ estimado via EM para todos os pares $(q,d)$

- Até a convergência de $\sigma(q \mid d)$ faça:

1. Para cada $q \in MP$ distinto faça:
   a. Para cada $(\mathbf{q}^s,\mathbf{d}^s) \in MP$ faça:
      i. Atribua $temp[q][s] \leftarrow 0$
      ii. Para cada $d \in \mathbf{d}^s$ faça :
      $\Rightarrow temp[q][s] += \sigma(q \mid d)$
2. Para cada $d \in MP$ distinto faça :
   a. Para cada $q \in MP$ distinto faça :
      i. Para cada $(\mathbf{q}^s,\mathbf{d}^s) \in MP$ faça :
      $$\Rightarrow totalContador[q][d] \mathrel{+}= \frac{f(q) \times f(d) \times \sigma(q \mid d)}{temp[q][s]}$$
      ii. $lamda[d] += totalContador[q][d]$
3. Para cada $d \in MP$ distinto faça :
   a. Para cada $q \in MP$ distinto faça :
      i. $$\sigma(q \mid d) \leftarrow \frac{totalContador[q][d]}{lambda[d]}$$

---

No entanto, um rearranjo dessas operações tornou possível que elas fossem processadas em seqüência de linhas. Segundo a documentação da API da ferramenta [Dragon Toolkit] o processamento de uma matriz da matriz esparsa implementada nessa ferramenta, quando ocorre por linha, é muito rápido, pois uma certa quantidade de linhas é mantida em cache durante o processamento da matriz e são periodicamente descarregadas ao se atingir um limite máximo ou pela chamada ao método flush. Além dessas modificações, o código dessa

ferramenta foi alterado de modo a não fazer o ordenamento das colunas antes de gravar as colunas em disco, pois o ordenamento de dezenas de milhares de colunas realizados milhares de vezes (cada vez que uma linha da matriz era descarregada em disco) fez diminuir em muito a velocidade de processamento das matrizes. Embora as colunas devam estar necessariamente ordenadas para se fazer a recuperação das entradas da matriz, o processamento das linhas e colunas já ocorriam de maneira ordenada, portanto, quando a linha era submetida para gravação em disco, ela já se encontrava ordenada. Uma última melhoria em termos de performance foi a definição de que a descarga em disco só seria feita com uma determinada quantidade de linhas processadas completamente, pois quando uma linha é inserida pela metade a ferramenta tenta fazer uma mescla das colunas da linha no momento da próxima escrita.

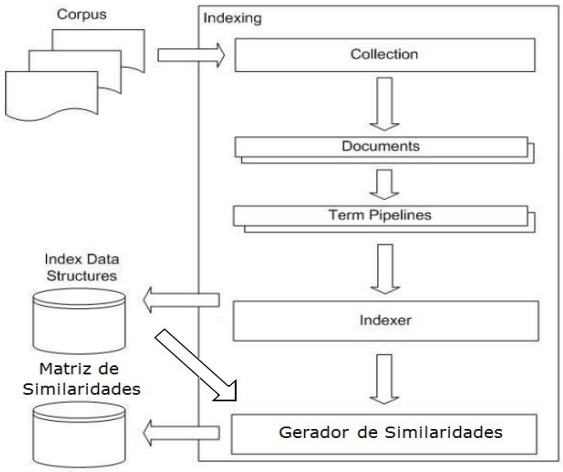



***Figura 4.2: Arquitetura de indexação segundo o modelo de recuperação de informação com a identificação de Similaridade entre as palavras***

## 4.4.Avaliação em Sistemas de Recuperação da Informação

A avaliação em sistemas de recuperação da informação tem a finalidade de mostrar o quanto o sistema atende o seu usuário final, não apenas em casos individuais, mas coletivamente, para todos os usuários reais e potenciais na comunidade. [Tague-Sutcliffe, 1996]. Embora alguns aspectos de um sistema de recuperação da informação possam ser aferidos sem a participação do usuário, a última palavra em termos de desempenho de um sistema de recuperação da informação só pode ser dada após alguns usuários reais ou potenciais tiverem usado o sistema em um experimento controlado de recuperação da informação. Fazer isso envolvendo pessoas reais, não é apenas um trabalho caro, é também difícil de controlar e de replicar. Por esta razão, tem sido desenvolvidos métodos para criar coleções de teste. Estas coleções de teste, como as coleções TREC descritas na seção anterior são criadas por meio da consulta de usuários reais, mas uma vez criadas, elas podem ser usadas para avaliar sistemas de recuperação da informação sem a necessidade de consultar os usuários novamente, o que permite rapidez e padronização nos métodos de avaliação. Para se fazer uma avaliação é necessária uma coleção de teste, composta de documentos, consultas e

níveis de relevância para as consultas apresentadas, além de uma metodologia estatística que determine se as diferenças observadas no desempenho entre os sistemas analisados são estatisticamente significantes. As coleções de teste consistem de documentos, consultas e julgamentos de relevância ("as respostas certas"). A efetividade de uma busca é geralmente medida pelo combinação de precisão e revocação, também conhecida por cobertura. A revocação é definida como a fração de documentos relevantes que foram recuperados pelo sistema. A precisão é definida como a fração dos documentos recuperados que são efetivamente relevantes.

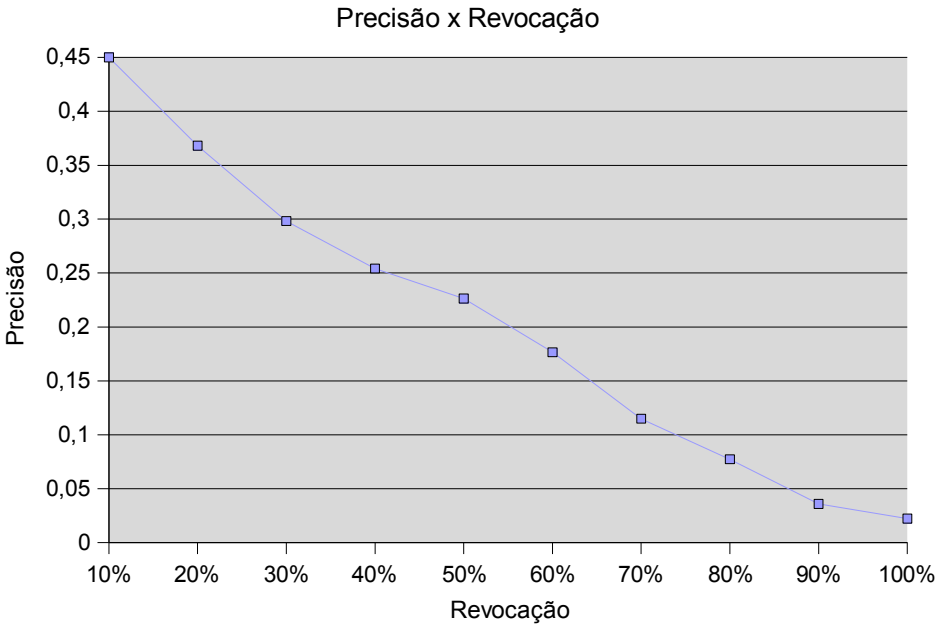
$$precisão = \frac{Nr\ Relevantes\ Recuperados}{Nr\ de\ Recuperados} \tag{20}$$

$$revocação = \frac{Nr\ Relevantes\ Recuperados}{Nr\ Total\ de\ Relevantes} \tag{21}$$

Como a relevância é um valor binário (relevante ou não relevante), então o desempenho na recuperação da informação é usualmente medido pela combinação de precisão e revocação. A avaliação geral de desempenho de um sistema é determinada pelo cálculo da média precisão e da revocação sobre um número suficientemente grande de consultas. Se o sistema faz uma classificação dos documentos em ordem decrescente de relevância, então é possível obter as médias de precisão e revocação de alguma forma considerando os diversos tamanhos de conjuntos de documentos recuperados, ou seja, considerando faixas de relevância de documentos: os primeiros 10%, 20%, 30% ... 100% dos documentos recuperados. A idéia é dar um número de medidas de avaliação para diferentes tipos de usuários. Em um extremo desse espectro está o usuário que está satisfeito com qualquer documento relevante, por exemplo, um usuário que procura na internet o resultado do jogo Cruzeiro no campeonato mineiro no último final de semana. No outro extremo está o usuário que somente estará satisfeito com o documento mais relevante, ou somente com a recuperação de todos os documentos relevantes, por exemplo, um analista do Tribunal de Contas da União que procura por jurisprudência em caso correlato ao que trata o processo de tomada de contas no qual ele está trabalhando. Na TREC três diferentes medidas de avaliação são usadas: precisão em níveis específicos de revocação, precisão em pontos específicos na lista de documentos recuperados e precisão média sobre os documentos recuperados.

Na precisão em níveis específicos de revocação é escolhido um número de níveis de revocação, por exemplo, 10 níveis: {0.1,0.2,0.3,...1.0}. Os níveis correspondem à usuários que estão satisfeitos se eles encontram 10%, 20%, 30%,...,100% dos documentos relevantes. Para cada um desses níveis a precisão correspondente é determinada calculando-se a precisão nesses níveis de revocação. Assim, por exemplo, se desejamos encontrar a precisão no nível 0.5 calculamos a razão entre o número de documentos relevantes recuperados e o número de documentos recuperados quando o total de documentos relevantes recuperados corresponde a 50% do total de documentos relevantes existentes na base de documentos. Esta informação geralmente é visualizada em gráficos. A Figura 4.3 mostra um exemplo de um gráfico desse tipo.

***Figura 4.3: Exemplo de gráfico Precisão X Revocação***




O gráfico mostra o comportamento típico de sistemas de recuperação da informação. Aumentando a revocação de uma busca, a precisão diminui. Ou seja, a medida que percorremos a lista de documentos recuperados em ordem decrescente de precisão em busca por mais documentos relevantes torna-se mais provável encontrar um documento não relevante do que um documento relevante.

A revocação pode não refletir a medida de satisfação do usuário com a abrangência dos documentos recuperados. Por exemplo, suponha que uma consulta tem 20 documentos relevantes enquanto outra tem 200. Uma revocação de 50% pode ser um objetivo razoável no primeiro caso, mas pode ser algo difícil de manipular para a maioria dos usuários no segundo caso [Hull, 1993]. Um método mais orientado ao usuário poderia ser simplesmente escolher um número fixo de pontos na lista de documentos recuperados, por exemplo: os 5, 10,15, 20, 30, 100, 200, 500 e 1000 mais relevantes documentos recuperados. Estes pontos correspondem a usuários que estão buscando 5, 10, 15, 20...1000 documentos por busca. Um problema potencial com essa medida, no entanto, é que embora a precisão e a revocação estejam em uma faixa compreendida entre 0 e 1, muitas vezes elas estão restritas a pequenas faixas, ou pontos fixos, da lista de documentos recuperados. Por exemplo, se, para uma consulta, existem 30 documentos relevantes na base, a precisão no ponto fixo 100 será de no máximo 0.3, e no ponto fixo 200 será de no máximo 0.15, o que não reflete o bom desempenho de um sistema de busca que esteja recuperando todos esses 30 documentos relevantes.

A precisão R é a precisão depois que R documentos relevantes foram recuperados, onde R é o número de documentos relevantes para a consulta considerada. A precisão R média para uma avaliação TREC completa é calculada tomando-se a média das precisões R de cada uma das consultas da avaliação. Por exemplo, suponha que uma avaliação consista de duas consultas, uma com 50 documentos relevantes e outra com 10 documentos relevantes. Se

o sistema recupera 17 documentos relevantes entre os 50 primeiros da lista de documentos recuperadas para a primeira consulta e 7 documentos relevantes entre os 10 primeiros documentos da lista para a segunda consulta, então a precisão R será:

$$precisão\ R = \frac{\frac{17}{50} + \frac{7}{10}}{2} = 0.52$$

A precisão média é um valor que reflete o desempenho do sistema sobre todos os documentos relevantes. Essa medida privilegia sistemas que recuperam primeiramente os documentos relevantes. Não se trata de uma média da precisão nos níveis de revocação escolhidos, mas de uma média dos valores de precisão obtidos sempre que um documento relevante é recuperado, quando um documento relevante não é recuperado sua precisão é assumida como zero. Considere, por exemplo, quatro documentos relevantes são recuperados nas posições 1, 2, 4 e 7 da lista de documentos recuperados. A precisão obtida quando cada documento relevante é recuperada é 1 (1/1), 1 (2/2), 0.75 (3/4) e 0.57 (4/7), respectivamente. A precisão média, então, é igual a 0.83.

# Capítulo 5. Estudo de caso

Este capítulo detalha o estudo de caso realizado utilizando bases de dados disponibilizadas pela TREC. A seção 5.2 detalha o formato dos documentos TREC, dá detalhes sobre os tipos de documentos que compõem um conjunto de testes TREC e os campos utilizados em cada um deles para identificação dos documentos relevantes para fins de avaliação. A seção 5.3 mostra os resultados da avaliação para as bases apresentadas.

## 5.1.Introdução

A pesquisa em recuperação da informação tem freqüentemente recebido críticas em duas frentes. Primeiro, falta uma fundamentação sólida. Segundo, faltam coleções de testes consistentes e *benchmarks*. No tocante à primeira crítica, é difícil corrigir o problema completamente em razão do grau de subjetividade envolvido na tarefa de decidir a relevância de um dado documento [Baeza & Ribeiro-Neto, 1999]. No tocante ao segundo, no entanto, existem iniciativas mundiais no sentido de fornecer esses *benchmarks* em documentos de diversos domínios de conhecimento (biologia, jornalismo, medicina, etc), em diversos formatos (texto estruturado, web, texto puro) e tanto para recuperação da informação *ad hoc* quanto para filtragem. Por três décadas a experimentação em recuperação da informação foi baseada em coleções de teste relativamente pequenas que não refletiam os principais aspectos presentes em uma grande base de documentos corporativa e menos ainda com relação aos documentos da internet. Além disso, comparações entre os vários sistemas de recuperação da informação era difícil porque grupos distintos de pesquisadores conduziam suas pesquisas baseada em diferentes aspectos da recuperação da informação (mesmo quando usando a mesma coleção de documento) e não haviam *benchmarks* amplamente utilizados.

No começo da década de 1990, uma reação a essa falta de padronização foi iniciada sob a liderança de Donna Harman no National Institute of Standards and Technology (NIST), em Maryland, EUA. Este esforço consistiu em promover uma conferência anual, que recebeu o nome TREC (do inglês, *Text Retrieval Conference*) [TREC, 1998], dedicada à experimentação com grandes coleções contendo mais de um milhão de documentos. Para cada conferência, um conjunto de experimentos de referência era selecionado, então os grupos de pesquisa que participavam da conferência usavam essas referências para comparar os seus sistemas.

As coleções TREC são distribuídas mediante o pagamento de uma taxa de distribuição, algo em torno de US$ 300,00 a depender da coleção solicitada. E os documentos vêm de uma série de fontes como o Wall Street Journal, a Associated Press, artigos em computação, patentes estadunidenses, Financial Times, Medline (um banco de dados *on-line* sobre artigos médicos), dentre outros. A quantidade de documentos em algumas coleções chega a ser de mais de 300.000 documentos.

## 5.2.Estrutura dos documentos TREC

As principais estruturas em um documento TREC são identificadas por meio de *tags* SGML para facilitar o processamento. O número do documento, por exemplo, é identificado pela *tag* <DOCNO> e o campo que contém o texto do documento é identificado pela *tag* <TEXT>. As subestruturas podem variar de coleção para coleção de modo a preservar a formatação original do documento. Os termos existentes nas coleções de documentos, na forma em que são distribuídos pela TREC não passaram pela remoção de stop-words e nem pela redução ao radical (*stemming*). Abaixo, um exemplo de um extrato de documento TREC utilizado no presente trabalho:

```
<DOC>
<DOCNO> 74001</DOCNO>
<DOCID> 00001 </DOCID>
<MEDNR> 75051687</MEDNR>
<AUTOR> Hoiby-N.  Jacobsen-L.  Jorgensen-B-A.  Lykkegaard-E.
Weeke-B.
</AUTOR>
<TITULO>Pseudomonas aeruginosa infection in cystic fibrosis.
Occurrence of precipitating antibodies against pseudomonas
aeruginosa in relation  to the concentration of sixteen serum
proteins and the clinical and radiographical status of the lungs.
</TITULO>
<FONTE>
Acta-Paediatr-Scand. 1974 Nov. 63(6). P 843-8.
</FONTE>
<PCHAVES>CYSTIC-FIBROSIS: co.  PSEUDOMONAS-AERUGINOSA: im.
PSEUDOMONAS-INFECTIONS: co.  RESPIRATORY-TRACT-INFECTIONS: co.MN
ADOLESCENCE.  BLOOD-PROTEINS: me.  CHILD.  CHILD-PRESCHOOL.
CYSTIC-FIBROSIS: im, bl.  FEMALE.  HUMAN.  IMMUNOELECTROPHORESIS.
IMMUNOGLOBULINS: me.  LUNG: ra.  MALE.  PRECIPITIN-TESTS.
PRECIPITINS.  PSEUDOMONAS-INFECTIONS: im, bl, ra. RESPIRATORY-
TRACT-INFECTIONS: bl, im, ra.  SERUM-ALBUMIN: me.
</PCHAVES>
<SUMARIO>The significance of Pseudomonas aeruginosa infection in
the respiratory tract of 9 cystic fibrosis patients have been
studied by means of immunoelectrophoretical analysis of patients'
sera for the number of precipitins against Pseudomonas aeruginosa
and the concentrations of 16 serum proteins.  In addition, the
clinical and radiographical status of the lungs have been
evaluated using 2 scoring systems.  Precipitins against
Pseudomonas aeruginosa were demonstrated in all sera, the maximum
number in one serum was 22. The concentrations of 12 of the serum
proteins were significantly changed compared with matched control
persons.  Notably IgG and IgA were elevated and the "acute phase
proteins" were changed, the latter suggesting active tissue
damage.  The concentrations of 3 of the acute phase proteins,
notably haptoglobin, were correlated to the number of precipitins
suggesting that the respiratory tract infection in patients with
many precipitins is accompanied by more tissue damage than the
infection in patients with few precipitins. The results indicate
no protective value of the many precipitins on the tissue of the
respiratory tract.
</SUMARIO>
</DOC>
```

***Figura 5.1: Representação de um documento de uma coleção distribuída pela TREC***

Neste trabalho são utilizadas três coleções de documentos: um subconjunto da base Medline contendo 54710 documentos, CFC e um subconjunto da base CETEN-Folha contendo 3500 documentos.

Medline é uma base de dados da literatura internacional na área médica e biomédica, produzida pela NLM (do inglês, National Library of Medicine) nos EUA. A partir dessa base de dados, a 9ª conferência TREC [TREC-9, 2000] extraiu um conjunto de 348566 referências, contendo os seguintes campos:

- número do documento (utilizado pelo TREC para identificar os documentos relevantes para cada consulta);
- o autor do documento;
- o periódico em que o mesmo foi publicado;
- as palavras chaves;
- o título;
- o resumo.

A base CFC também é um subconjunto da base Medline, com 1239 documentos e está disponível no sítio do livro Modern Information Retrieval na [Universidade de Berkeley].

O CETEN-Folha (Corpus de Extratos de Textos Eletrônicos NILC/Folha de S. Paulo) é um corpus de cerca de 24 milhões de palavras em português brasileiro, criado pelo projeto [Processamento computacional do português](projeto que deu origem à [Linguateca]) com base nos textos do jornal Folha de São Paulo que fazem parte do corpus NILC/São Carlos, compilado pelo [Núcleo Interinstitucional de Lingüística Computacional](NILC). Este corpus está dividido em 340.947 extratos, classificados por semestre e caderno do jornal do qual provêm. Cada extrato está dividido em parágrafos e frases, títulos e autores dos artigos

Todas as bases foram pré-processadas, uma vez que o formato em que são distribuídas não permitem a leitura pelo Terrier, pois não seguem o padrão xml para leitura de documentos TREC pelo Terrier. A Figura 5.1 mostra um extrato de um arquivo da base Medline, após o pré-processamento.

Juntamente com as coleções de documentos Medline e CFC, é disponibilizado um conjunto de consultas, em um arquivo com formato específico. Este arquivo descreve as necessidades reais de informações, onde foi solicitado aos especialistas para descrever o paciente que os motivou a buscar aquela informação e qual informação que necessitavam para o tratamento do referido paciente.

Na Figura 5.2 é mostrado um extrato desse arquivo. A *tag* <TITLE> contém uma descrição do paciente e a *tag* <DESC> contém a informação que o especialista precisa recuperar.

```
<TOP>
<NUM> OHSU1 </NUM>
<TITLE>
60 year old menopausal woman without hormone replacement
therapy
</TITLE>
<DESC>
Are there adverse effects on lipids when progesterone is given
with estrogen replacement therapy
</DESC>
</TOP>
<TOP>
<NUM> OHSU2 </NUM>
<TITLE>
60 yo male with disseminated intravascular coagulation
</TITLE>
<DESC>
pathophysiology and treatment of disseminated intravascular
coagulation
</DESC>
</TOP>
```

***Figura 5.2: Representação de um documento de especificação de consultas de uma coleção distribuída pela TREC***

A *tag* <NUM> serve para identificar o número da consulta, este campo será usado, no arquivo descritivo dos documentos relevantes para cada consulta. Este arquivo está no formato texto puro, e contém uma entrada para cada linha do arquivo. Cada entrada contém o número da consulta, o documento recuperado e o nível de relevância do documento para aquela consulta atribuído por um especialista humano.

Desta forma, a avaliação de desempenho de um sistema de recuperação da informação pode ser feita comparando os documentos que ele recupera com os documentos que era esperado que ele recuperasse, segundo o arquivo descritivo dos documentos relevantes para cada consulta. A Figura 5.3 contém um extrato desse arquivo:

```
OHSU1     87316326  1
OHSU1     87202778  1
OHSU1     87157536  2
OHSU1     87157537  2
OHSU1     87097544  2
OHSU1     87316316  1
OHSU2     87230756  1
OHSU2     87076950  1
OHSU2     87254296  2
OHSU2     87058538  2
OHSU2     87083927  2
OHSU2     87309677  2
OHSU2     87198965  1
OHSU2     87312037  1
OHSU2     87065512  1
OHSU2     87057614  1
```

***Figura 5.3: Representação de um documento de especificação dos níveis de relevância de uma coleção distribuída pela TREC***

Apesar das melhorias apresentadas na seção 4.3 , a definição do procedimento de EM implica a existência de laços aninhados em três níveis para o cálculo da matriz *temp* e *totalContador*. Por isso, nos experimentos realizados com a base Medline, contendo 54710 documentos, um léxico de mais de 60000 termos distintos, um conjunto de palavras da

consulta da ordem de 25000 termos distintos, a criação dos índices de similaridade demorava cerca de 4 dias em um computador com processador de 1.8 Ghz de núcleo único, pois o cômputo da matriz *temp*, por exemplo, tem o seguinte número de operações:

$$Nr\ termos\ consulta \times Nr\ de\ documentos\ na\ base \times Tamanho\ médio\ dos\ documentos$$

Em razão desse número de operações, a base Medline foi indexada com apenas uma iteração do algoritmo de EM. Para a base CFC, foram executadas cinco iterações do algoritmo de EM, até que a diferença entre os graus de similaridade entre uma iteração e a anterior não ficasse superior a 0.001. Os resultados obtidos em ambos os casos são apresentados na próxima seção.

## 5.3.Avaliação dos resultados

Foram executados experimentos com a base CETEN-Folha, Medline e com a base CFC. A base CETEN-Folha não dispõe de um conjunto de consultas e de julgamentos de relevância dos documentos e, portanto, não há como avaliar o desempenho desse modelo em língua portuguesa, segundo a metodologia de avaliação da TREC, pois não há base de testes nessa língua. No entanto, os experimentos realizados em língua portuguesa mostraram que o modelo agrupava palavras próximas segundo a co-ocorrência das palavras, como pode ser verificado na Figura 5.4:

| Torneio | |
|---|---|
| Torneio | 0,12 |
| Vencedor | 0,09 |
| Matt | 0,06 |
| Promovido | 0,04 |
| sena | 0,04 |
| Derrota | 0,03 |
| Enfrentar | 0,03 |
| Santos | 0,03 |
| América | 0,03 |
| Corinthians | 0,02 |
| Conmebol | 0,02 |
| Conseguir | 0,02 |
| Mogi-Mirim | 0,02 |
| Paulista | 0,02 |
| Novorizontino | 0,02 |

| Polícia | |
|---|---|
| Polícia | 0,09 |
| PF | 0,08 |
| Treinamento | 0,08 |
| Carioca | 0,08 |
| Lacrar | 0,06 |
| Exército | 0,03 |
| Recusar | 0,03 |
| Ganhar | 0,03 |
| Rodney King | 0,03 |
| Federal | 0,02 |
| Roubar | 0,02 |
| Prender | 0,02 |
| Advogado | 0,02 |
| Tentativa | 0,02 |
| Acusado | 0,02 |

| Droga | |
|---|---|
| Baixa | 0,07 |
| Carcamano | 0,07 |
| Esquecer | 0,07 |
| Droga | 0,07 |
| Inglês | 0,05 |
| PF | 0,05 |
| Combate | 0,05 |
| Asma | 0,05 |
| Treinamento | 0,03 |
| Carioca | 0,03 |
| Roubo | 0,03 |
| PM | 0,03 |
| Tentar | 0,03 |
| Homens | 0,02 |
| Prender | 0,02 |

***Figura 5.4: Similaridades obtidas para três termos do índice criado para a base CETEN-Folha***

Os resultados obtidos para a base Medline estão apresentados na Figura 5.5.

```
_____________________________________
Number of queries  = 63
Retrieved          = 63000
Relevant           = 670
Relevant retrieved = 570

_____________________________________
Average Precision: 0.0980
R Precision      : 0.1070

_____________________________________
Precision at    1: 0.1270
Precision at    2: 0.1190
Precision at    3: 0.1111
Precision at    4: 0.1071
Precision at    5: 0.1048
Precision at   10: 0.1032
Precision at   15: 0.0889
Precision at   20: 0.0825
Precision at   30: 0.0762
Precision at   50: 0.0670
Precision at  100: 0.0460
Precision at  200: 0.0314
Precision at  500: 0.0158
Precision at 1000: 0.0090

_____________________________________
Precision at   0%: 0.2675
Precision at  10%: 0.2030
Precision at  20%: 0.1452
Precision at  30%: 0.1151
Precision at  40%: 0.1051
Precision at  50%: 0.0983
Precision at  60%: 0.0863
Precision at  70%: 0.0753
Precision at  80%: 0.0592
Precision at  90%: 0.0316
Precision at 100%: 0.0185

_____________________________________
Average Precision: 0.0980
```

***Figura 5.5: Avaliação da precisão e da revocação do modelo proposto para a base Medline, com 54710 documentos e 63 consultas***

A mesma coleção, indexada e recuperada segundo o modelo DFR, apresentou o seguinte desempenho na avaliação realizada:

```
_____________________________________
Number of queries  = 63
Retrieved          = 489944
Relevant           = 670
Relevant retrieved = 656

_____________________________________
Average Precision: 0.4009
R Precision      : 0.3936

_____________________________________
Precision at    1: 0.5714
Precision at    2: 0.5714
Precision at    3: 0.5503
Precision at    4: 0.5000
Precision at    5: 0.4667
Precision at   10: 0.3698
Precision at   15: 0.3164
Precision at   20: 0.2667
Precision at   30: 0.2090
Precision at   50: 0.1489
Precision at  100: 0.0852
Precision at  200: 0.0462
Precision at  500: 0.0197
Precision at 1000: 0.0104

_____________________________________
Precision at   0%: 0.7275
Precision at  10%: 0.6969
Precision at  20%: 0.6022
Precision at  30%: 0.5132
Precision at  40%: 0.4598
Precision at  50%: 0.4249
Precision at  60%: 0.3501
Precision at  70%: 0.3057
Precision at  80%: 0.2386
Precision at  90%: 0.1606
Precision at 100%: 0.1230

_____________________________________
Average Precision: 0.4009
```

***Figura 5.6: Avaliação da precisão e da revocação do modelo DFR para a base Medline, com 54710 documentos e 63 consultas***

Pela análise das Figura 5.5 e da Figura 5.6 fica claro que o desempenho do modelo proposto foi bem inferior àquele que obteve o modelo DFR. Enquanto no modelo DFR a precisão média foi de 0.4009, no modelo proposto a precisão média foi 0.0980. Uma diferença de mais de 300% a favor do modelo DFR. Certamente, o modelo não foi ajustado devido à falta de iterações do algoritmo de EM. A alternativa a essa constatação foi executar o algoritmo em uma base menor, a base CFC, que poderia dar resultados mais rápidos e provavelmente mais precisos devido a possibilidade de ser executado um maior número de iterações de EM.

Para a base CFC, o algoritmo de EM executou cinco iterações. Os resultados dessa base, para o modelo proposto e para o modelo DFR estão apresentados nas Figuras 5.7 e 5.8, respectivamente:

```
_____________________________________
Number of queries  = 100
Retrieved          = 123799
Relevant           = 2232
Relevant retrieved = 2230
_____________________________________
Average Precision: 0.2451
R Precision      : 0.2475
_____________________________________
Precision at    1: 0.3400
Precision at    2: 0.3200
Precision at    3: 0.3333
Precision at    4: 0.3150
Precision at    5: 0.2980
Precision at   10: 0.2700
Precision at   15: 0.2420
Precision at   20: 0.2255
Precision at   30: 0.1953
Precision at   50: 0.1602
Precision at  100: 0.1182
Precision at  200: 0.0777
Precision at  500: 0.0396
Precision at 1000: 0.0223
_____________________________________
Precision at   0%: 0.5300
Precision at  10%: 0.4514
Precision at  20%: 0.3743
Precision at  30%: 0.3083
Precision at  40%: 0.2749
Precision at  50%: 0.2508
Precision at  60%: 0.2039
Precision at  70%: 0.1739
Precision at  80%: 0.1483
Precision at  90%: 0.0970
Precision at 100%: 0.0614
_____________________________________
Average Precision: 0.2451
```

***Figura 5.7: Avaliação da precisão e da revocação do modelo proposto para a base CFC, com 1239 documentos e 100 consultas***

Para a base CFC, os índices de precisão do modelo DFR e do modelo proposto ficaram muito mais próximos. Enquanto a precisão média no modelo DFR foi de 0.3375, no modelo proposto a precisão média foi de 0.2451. Uma diferença de cerca de 37% a favor do modelo DFR.

```
_____________________________________
Number of queries  = 100
Retrieved          = 9956
Relevant           = 2232
Relevant retrieved = 1172

_____________________________________
Average Precision: 0.3185
R Precision      : 0.3571

_____________________________________
Precision at    1: 0.5600
Precision at    2: 0.5350
Precision at    3: 0.5000
Precision at    4: 0.4725
Precision at    5: 0.4620
Precision at   10: 0.3850
Precision at   15: 0.3327
Precision at   20: 0.2940
Precision at   30: 0.2360
Precision at   50: 0.1774
Precision at  100: 0.1172
Precision at  200: 0.0586
Precision at  500: 0.0234
Precision at 1000: 0.0117

_____________________________________
Precision at   0%: 0.7328
Precision at  10%: 0.6210
Precision at  20%: 0.5511
Precision at  30%: 0.4867
Precision at  40%: 0.4127
Precision at  50%: 0.3445
Precision at  60%: 0.2466
Precision at  70%: 0.1519
Precision at  80%: 0.1012
Precision at  90%: 0.0467
Precision at 100%: 0.0298

_____________________________________
Average Precision: 0.3185
```

***Figura 5.8: Avaliação da precisão e da revocação do modelo DFR para a base CFC, com 1239 documentos e 100 consultas***

Pela a análise dos resultados apresentados na figura 5.8, constatamos que, apesar da capacidade do algoritmo de agrupar palavras semelhantes segundo o domínio em que são empregadas (como apresentado na figura 5.4), de alguma sorte, a identificação de palavras similares não foi utilizada de forma a melhorar a precisão média quando comparada com um modelo probabilístico como o DFR. Ora, se o pressuposto de que a ocorrência de palavras próximas à palavra utilizada na consulta é válido, então fica claro que uma potencial melhoria ao desempenho do algoritmo proposto consiste na forma em que esses índices de similaridades são usados para estimar a relevância de um documento. No algoritmo 2, o grau de proximidade entre a palavra da consulta e a palavra do documento é multiplicada pela freqüência desta no documento a fim de obter a sua contribuição no grau de relevância do documento para aquela palavra da consulta considerada. No entanto, essa abordagem tem o

efeito indesejado de considerar todas as palavras do documento, independentemente do grau de capacidade daquela palavra em representar o documento.

```
_____________________________________
Number of queries  = 100
Retrieved          = 41757
Relevant           = 2232
Relevant retrieved = 1799

_____________________________________
Average Precision: 0.3152
R Precision      : 0.3295

_____________________________________
Precision at    1: 0.4800
Precision at    2: 0.5000
Precision at    3: 0.4767
Precision at    4: 0.4575
Precision at    5: 0.4380
Precision at   10: 0.3570
Precision at   15: 0.3207
Precision at   20: 0.2785
Precision at   30: 0.2237
Precision at   50: 0.1712
Precision at  100: 0.1131
Precision at  200: 0.0715
Precision at  500: 0.0352
Precision at 1000: 0.0180

_____________________________________
Precision at   0%: 0.7028
Precision at  10%: 0.5913
Precision at  20%: 0.5216
Precision at  30%: 0.4609
Precision at  40%: 0.3980
Precision at  50%: 0.3333
Precision at  60%: 0.2547
Precision at  70%: 0.1751
Precision at  80%: 0.1337
Precision at  90%: 0.0733
Precision at 100%: 0.0396

_____________________________________
Average Precision: 0.3152
```

***Figura 5.9: Avaliação da precisão e da revocação do modelo TF_IDF para a base CFC, com 1239 documentos e 100 consultas***

Esta abordagem pressupõe que uma palavra do documento terá tanto mais peso na obtenção do nível de relevância do documento segundo a palavra da consulta considerada, quanto maiores forem dois indicadores:

1. O grau de similaridade entre a referida palavra do documento e a palavra da consulta considerada

2. A frequência da ocorrência da palavra no documento.

Enquanto que o indicador (1) está fortemente vinculado às considerações feitas até o momento com relação à busca por documentos que contenham palavras similares, o indicador (2) pode não identificar, necessariamente, a importância da palavra no documento. A introdução do indicador (2) é uma forma de identificar a capacidade da palavra considerada representar o documento, uma vez que não basta encontrar palavras similares no documento considerado se esta palavra não representar o conteúdo do documento.

Uma possível solução para esse problema consiste em identificar qual subconjunto dos termos do documento que deve ser utilizado para o cálculo da similaridade deste documento, ou seja, escolher as palavras que representam o documento e então verificar a sua similaridade com as palavras da consulta, por meio do índice de similaridades criado. Isso porque um documento que contenha uma grande quantidade de palavras, das quais somente uma parte delas tem muita similaridade com o termo da consulta considerado, iria produzir uma similaridade pequena entre documento e termo da consulta, uma vez que o total de similaridades é normalizado. Ou seja, a similaridade alta de alguns dos termos do documento seria distribuída nos outros termos do documento.

O modelo DFR é uma medida que considera que quanto maior a divergência entre a freqüência do termo no documento e na coleção mais relevante é o termo para o documento, além disso são feitas duas normalizações: a primeira considera a probabilidade de o termo pertencer a um conjunto de elite do documento (conjunto que identifica os termos que mais representam o documento), e a segunda considera a razão entre o tamanho do documento considerado e o tamanho médio dos documentos. Este modelo esteve entre aqueles com melhor desempenho na TREC-10 [Amati & Van Rijsbergen, 2002] e é considerado no presente trabalho como uma medida para identificar a capacidade de uma palavra representar um documento.

Desta forma, para calcular o grau de similaridade de um documento com uma palavra da consulta não consideramos todos os termos do documento, mas apenas os termos que pertencem ao conjunto de elite, definido pela medida DFR. Assim, a normalização seria feita apenas entre termos relevantes e não entre todos os termos, evitando dessa forma a perda da precisão decorrente da normalização de documentos grandes ou de documentos que, embora possuam muitos termos similares, possuam ainda outros termos de baixa similaridade que contribuem para deteriorar a medida de relevância do documento.

Os testes realizados segundo essa abordagem, que passaremos de chamar de “busca de similaridades a partir do conjunto de elite”, resultaram em uma precisão média de 0.2844, o que reduziu a diferença a favor do modelo DFR puro para 18%.

A investigação da utilização do modelo DFR como um meio para definir um conjunto de elite e os resultados obtidos a partir desses experimentos levam à experimentação de um modelo de mistura (do inglês, *mixture model*) onde a contribuição de cada modelo no índice de relevância do documento é obtido a partir de um conjunto de coeficientes de soma 1.

| Coeficiente da Abordagem Proposta no cálculo da relevância de um documento | Precisão Média |
|---|---|
| 1 | 0.2844 |
| 0.25 | 0.3467 |
| 0.2 | 0.3538 |
| 0.15 | 0.3558 |
| 0.1 | 0.3555 |
| 0.05 | 0.3536 |
| 0 | 0.3375 |

***Tabela 1: Pesos utilizados na composição da abordagem proposta com o modelo DFR (mixture model) e a precisão média obtido em cada uma dessas composições***

Os resultados obtidos acima mostram que uma combinação do modelo de linguagem proposto com um modelo probabilístico DFR resulta em resultados melhores que aqueles obtidos pela aplicação dos modelo de maneira isolada.

# Capítulo 6. Conclusão

Este capítulo apresenta um resumo das atividades realizadas, os resultados obtidos, as contribuições científicas e tecnológicas decorrentes, as limitações da arquitetura proposta e indicações de possíveis trabalhos futuros

O objetivo da presente pesquisa é propor uma arquitetura de recuperação de informação que permitam ampliar uma consulta, com potencial melhoria no índice de precisão, mantido o índice de revocação, sem requerer a construção de estruturas auxiliares tais como tesauros ou ontologias. Para tanto, foi escolhida a abordagem de tratamento estatístico da linguagem visando o cálculo de índices de similaridade entre termos do corpus. Para cálculo das similaridades foi utilizada a abordagem de estimação de parâmetros via EM.

Os principais resultados obtidos foram:

1. proposição de uma nova arquitetura de sistema de RI levando em conta índices de similaridades (Seção 4.3).

2. detalhamento do algoritmo para cálculo automático de índices de similaridades entre documentos, a partir da abordagem EM (Seção 4.3), proposto em [Berger & Lafferty, 1999]. Por se tratar de um algoritmo iterativo e lidar com matrizes de muito alta dimensionalidade, os requisitos de memória podem ser tão elevados que os torne inviáveis. Para lidar com esse problema, o algoritmo detalhado foi alterado visando permitir gerência de armazenamento em memória virtual (Seção 4.3).

3. implementação de um protótipo de um sistema de recuperação de informação segundo a metodologia proposta (Capítulo 5). Esse protótipo foi construído a partir do software aberto para RI denominado Terrier (Seção 3.1). Para tanto, o módulo de indexação (Seção 3.2) foi alterado para permitir a construção de índices de similaridade através do algoritmo EM detalhado (Seção 4.3). Em função do grande requisito de memória para representar a matriz de similaridade, esse módulo também foi alterado para utilizar a técnica de matrizes esparsas. O módulo de recuperação (classe *matching*) foi estendido para ampliar a consulta levando em conta os índices de similaridade (gerando a classe S*imilaridadesMatching*) (Seção 3.3). O protótipo implementado constitui um exemplo de como configurar o Terrier para o teste de algoritmos específicos de indexação e recuperação de informação desenvolvidos por um pesquisador em RI. Esse fato contribui para maior divulgação do Terrier na comunidade de pesquisa nesta área no Brasil.

Destaca-se entre as contribuições científicas:

1. Proposição de um algoritmo de EM com utilização de matrizes esparsas manipuladas em disco combinado com gerência de memória (Seção 4.3)

2. Obtenção automática de índice de similaridades entre palavras da língua portuguesa por meio da modelagem estatística da linguagem baseada no modelo proposto por [Brown et al.,1993] para tradução estatística da linguagem.

3. Criação de modelos de linguagem da consulta por meio de análise de co-ocorrência das palavras do título do documento e das palavras-chave.

Destaca-se entre as contribuições tecnológicas:

1. Detalhamento do algoritmo de EM [Berger & Lafferty, 1999] para obtenção de similaridades entre termos da consulta e termos do documento.

2. Extensão da plataforma de desenvolvimento de aplicações em recuperação da informação Terrier para criação e utilização de estruturas de identificação das similaridades entre os termos da consulta e do documento.

A identificação de índices de similaridade entre palavras na recuperação da informação, vem contribuir com a melhoria nos índices de precisão de sistemas de RI onde não se dispõe de um tesauro voltado para o domínio da aplicação. Assim, a identificação de agrupamentos entre as palavras tende a apresentar melhores resultados em domínios específicos, já que é capaz de identificar, indiretamente, o sentido em que uma determinada palavra é usada naquele domínio. Essa identificação da semântica das palavras ocorre justamente pelo agrupamento de termos semelhantes dentro de um domínio específico.

O desempenho do modelo proposto deve ser objeto de uma análise minuciosa, a fim de que sejam feitos os devidos refinamentos, já que não foi empregada até aqui nenhuma técnica de suavização (smoothing) para atenuar o peso das palavras da consulta que têm similaridade com os termos do documento próximos a zero, o que é bastante possível, já que em um corpus pequeno há menor probabilidade de co-ocorrência das palavras.

Além disso, trabalhos futuros devem tratar da capacidade de atualização incremental da matriz de similaridades, a fim de tornar desnecessária a reconstrução de todas as entradas da matriz o que implica em um grande tempo de processamento.

Devem ainda serem feitos testes a fim de definir como o grau de similaridade de uma palavra com ela própria pode contribuir para a melhoria dos resultados de precisão média, uma vez que o modelo proposto não apresenta nenhuma garantia de que o grau de similaridade entre uma palavra e ela própria será maior que o seu grau de similaridade com outras palavras, o que pode degradar os resultados em casos muito específicos onde seja recorrente a escolha de uma palavra para o título ou palavras-chaves que ocorra com baixa freqüência no corpo do texto.

A presente pesquisa pode ainda ser ampliada com a consideração do contexto onde as palavras ocorrem. Desta forma, a importância de uma palavra do documento para identificação do grau de similaridade dependerá não só do grau de similaridade desta com a palavra da consulta considerada, mas também da probabilidade de ocorrência daquela palavra

do documento dada a ocorrência de outras palavras da consulta. A técnica de n-gramas têm sido uma abordagem estatística amplamente utilizada quando se deseja considerar também o contexto no qual as palavras ocorrem [Manning and Schütze, 1999].

# Bibliografia


[Amati & Van Rijsbergen, 2002] Amati, G. and Van Rijsbergen, C. J.. **Probabilistic models of Information Retrieval based on measuring divergence from randomness**. *ACM Transactions on Information Systems*, Vol. 40, No. 4, pp. 1—33, 2002.

[Baeza & Ribeiro-Neto, 1999]: Baeza-Yates and B. Ribeiro-Neto, **Modern Information Retrieval.**, 1999

[Bengio, 1999]: Bengio, Y, **Markovian models for sequential data**, 1999, Neural ComputingSurveys 2,129–162

[Berger & Lafferty, 1999]: A. Berger and J. Laffety, **Information retrieval asstatistical translation**, 1999, Proceedings of SIGIR '99,222-229

[Brown et al.,1993]: P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer., **The mathematics of statisticalmachine translation: Parameter estimation**, 1993

[Cao et al., 2005]: Cao, Guihong, Jian-Yun Nie and Jing Bai, **Integrating word relationshipsinto language models**, 2005, Proceedings of the Twenty-Eight Annual International Conference on Research and Development in Information Retrieval, 298-305

[Croft & Lafferty, 2003]: Croft, W. Bruce and John Lafferty, **Language Modeling for InformationRetrieval**, 2003

[Croft et al., 1995]: Croft, W. B.; Cook, R. & Wilder, D. , **Providing Government Information on The Internet: Experiences with THOMAS**, 1995,Digital Libraries Conference ,19-24

[Dempster, Laird & Rubin, 1977]: A. Dempster, N. Laird, and D. Rubin., **Maximum likelihood from incomplete data via the EM algorithm.**, 1977,Journal of the Royal Statistical Society

[Dragon Toolkit]: , http://www.dragontoolkit.org/api/index.html

[Garfield, Eugene, 1997]: Eugene Garfield. A Tribute to Calvin N. Mooers, A Pioneer of Information Retrieval. **The Scientist**, v.11, n.6, p.9, March 17, 1997

[Hiemstra & Kraaij, 1999]: D. Hiemstra and W. Kraaij, **Twenty-One at TREC-7:ad-hoc and cross-language track**, 1999,Proceedings of the seventhText Retrieval Conference TREC-7,,227-238

[Hofmann, 1999]: Hofmann, T, **Probabilistic latent semantic indexing**, 1999,Proceedings of the 22nd Annual International ACM SIGIR Conference

[Hull, 1993]: Hull, D. , **Using statistical testing in the evaluation of retrieval experiments**, 1993,In Proceedings of the 16th ACM Conference on Research andDevelopment in Information Retrieval

[Jelinek, 1997]: Jelinek, F. , **Statistical Methods for Speech Recognition**, 1997

[Jordan, 1998]: Jordan, M. I. , **Learning in Graphical Models**, 1998

[Lawrie, 2003]: Lawrie, D., **Language Models for Hierarchical Summarization**, 2003

[Lewis and Jones, 1996]: Lewis, David D. and Karen Sparck Jones, **Natural language processing for information retrieval**, 1996,Communications of the ACM,92-101

[Linguateca]: , http://acdc.linguateca.pt/cetenfolha/

[Manning and Schütze, 1999]: Christopher D. Manning, Hinrich Schütze, **Foundations of Statistical Natural Language Processing**, MIT Press: 1999. ISBN 0-262-13360-1.

[Miller et al.. 1999]: D. Miller, T. Leek and R. M. Schwartz, **A hiddenMarkov model information retrieval system**, ,Proceedings ofSIGIR'1999,214-222

[Miller, 1990]: Miller, George A. , **WordNet: An on-line lexical database**, 1990, International Journal of Lexicography

[Moens, 2006]: Moens, Marie-Francine, **Information Extraction: Algorithms and Prospects in a Retrieval Context**, 2006, The Information Retrieval Series

[Núcleo Interinstitucional de Lingüística Computacional]: http://www.nilc.icmsc.sc.usp.br/

[Ounis et al. 2006]: Lioma, C. and Macdonald, C. and Plachouras, V. and Peng, J. and He, B. and Ounis I., **University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier**, 2006,Proceeddings of the 15th Text REtrieval Conference (TREC 2006),

[Ponte & Croft, 1998]: J. Ponte and W. B. Croft, **A language modelingapproach to information retrieval**, 1998

[Processamento computacional do português]: http://www.linguateca.pt/proc_comp_port.html

[Rabiner, 1990]: Rabiner, L. R., **A tutorial on hidden Markov models and selected applications in speech recognition**, 1990. In A. Waibel and K. F. Lee (Eds.),Readings in speech recognition,267–296

[Rasmussen, 1999]: Rasmussen, E. M, **Libraries and bibliographical systems**, 1999,R. A. Baeza-Yates and B. Ribeiro-Neto (Eds.), Modern Information Retrieval, 397–413

[Rosenfeld, 2000]: Rosenfeld, R., **Two decades of statistical language modeling: where do we go from here?** , 2000,In Proceedings of the IEEE

[Salton, 1971]: Salton, G., **The SMART retrieval system: Experiments in automatic document processing.**, 1971

[Salton, 1989]: Salton, Gerard, **Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer**, 1989

[Song & Croft, 1999]: Song, F., & Croft, W. B. , **A general language model for information retrieval**, 1999, Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 279-280

[Tague-Sutcliffe, 1996]: Tague-Sutcliffe, J. M., **Some perspectives on the evaluation of information retrieval systems**, 1996, Journal of the American Society for InformationScience

[TREC-9, 2000]: , http://trec.nist.gov/data/t9_filtering.html

[TREC, 1998]: , http://trec.nist.gov

[Universidade de Berkeley]: http://people.ischool.berkeley.edu/~hearst/irbook/

[Witten, Moffat and Beel, 1994]: Witten I. H., A. Moffat and T.C. Beel, **Managing Gigabytes:**

**Compressingand Indexing Documents and Images**, 1994

[Zhai & Lafferty, 2001]: C. Zhai and J. Lafferty, **A study of smoothing methods for language models applied to ad hoc information retrieval**, 2001, Proceeding of SIGIR'01,334-342