Ciro Meneses Santos

Filtragem de Informação com Utilização e Reformulação de Perfil

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Curso de Pós-graduação em Ciência da Computação, Departamento de Ciência da Computação, Instituto de Ciências Exatas, Universidade Federal de Minas Gerais. Orientador: Prof. Newton José Vieira.

Belo Horizonte

2003

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O objetivo desta dissertação é apresentar uma proposta para melhorar a qualidade das respostas em sistemas de recuperação de informação para a Web. Para a recuperação de informação, é utilizado o modelo vetorial. Em seguida, as informações recuperadas são filtradas tomando-se como base o perfil do usuário. Após a classificação obtida através do processo de filtragem, o perfil armazenado é reformulado, a partir de informações capturadas do usuário, utilizando-se o método de contribuição de palavras (*word contribution*) [13][15][14]. Isto introduz no sistema um certo grau de conhecimento sobre o usuário.

Com a filtragem feita a partir da consideração do perfil do usuário, assim como a reformulação deste mesmo perfil com base na experiência, é obtida uma melhoria na qualidade dos resultados retornados pela máquina de busca. Os resultados experimentais indicam que, em ambientes onde são possíveis o controle e interação com o usuário, o método proposto pode ser uma alternativa viável para implementação em um sistema de recuperação de informação.

Abstract

The aim of this paper is to present a proposal to improve the quality of query results in information retrieval systems for the Web. The vetorial model is used for the information retrieval. Then, the retrieved informations are filtered taken as base the user´s profile. After the classification is obtained throught the filtering process, the profile is reformulated taken into account the user´s captured information, by using the method of word contribution. This process introduces a certain level of knowledge about the user into the system.

With the filtering, considering the user´s initial profile as well as the reformulated profile, there is an improvement in the quality of the results returned by the search machine. The experimental results indicate that, in places where the control and interaction with the user are possible, this method can be a feasible alternative for implementation in an information retrieval system.

Agradecimentos

Uma das tarefas mais complicadas é a de fazer justiça e lembrar de todas as pessoas e entidades que contribuíram para a realização deste trabalho. Posso não ter citado nominalmente todos, entretanto, saibam que as pessoas aqui lembradas representam um sentimento: Se você estiver empenhado a aprender, sempre existirá pessoas disponíveis a ajudar e grandes projetos sempre tem a participação de várias pessoas. Em especial, eu gostaria de agradecer.

ao meu orientador, Professor Newton José Vieira, pela dedicação, confiança, incentivo e amizade.

À minha esposa Andréia, meus filhos Bruno e Ciro Matheus, pelo apoio, carinho e compreensão.

Ao Professor Ulisses de Azevedo Leitão, que foi desde do inicio um grande amigo e incentivador e que em alguns momentos difíceis, sempre esteve ali para ajudar.

Aos meus pais Antônio e Zilda, que nunca deixaram de acreditar em mim.

Á FIC, Faculdade de Ciência da Computação de Caratinga - MG, pelo incentivo e apoio financeiro.

Aos professores Nívio Ziviani, Antônio Alfredo, Roberto Bigonha, Henrique Pacca, e Wagner Meira, pela contribuição inestimável à minha formação.

Aos membros da banca examinadora, Nivio Ziviane, Henrique Pacca, e Antônio Braga, por suas valiosas contribuições.

Aos alunos do 6º período do curso de Ciência da Computação da FIC - Faculdades Integradas de Caratinga, pela participação no processo de avaliação do Sistema de Filtragem.

Aos colegas do LATIN, em especial Maria de Lourdes, Pavel, Charles, Wesley e Claudine.

Aos colegas do LINC, em especial Lucília Camarão e Hermann.

Ao Departamento de Informática (DCC) da UFMG, pelo apoio.

Aos colegas da Pós-graduação do DCC que estiveram em todos os momentos tolerando, ajudando, brincando, explicando, aprendendo, trabalhando, convivendo, revivendo, reclamando, enfim vivendo.

Ao famoso cafezinho do DCC que não está explicitamente liberado para os alunos da Pós-graduação, entretanto, espero que nunca seja proibido.

Capítulo 1

Introdução

A World Wide Web vem se tornando, ao longo dos anos, uma fonte importante de recursos bibliográficos. Isto se dá, não apenas pela enorme quantidade de informações colocadas disponíveis, mas também pela facilidade de recuperação destas informações. Tal facilidade é suprida por portais de busca, como Altavista[1], Google[2], Yahoo[3], e muitos outros, os quais lidam com o problema de armazenar e organizar as informações de maneira que sua recuperação seja rápida, eficiente e atenda às necessidades dos pesquisadores.

Um problema importante relacionado às máquinas de busca dos portais acima mencionados é a quantidade de documentos não relevantes retornados para o usuário. Por exemplo, ao se pesquisar sobre determinado autor brasileiro, é comum ter como resultado, além de informações relevantes, grande número de informações sem relevância, como indicações de instituições públicas e privadas, localizações geográficas, agremiações diversas, etc., que possuem coincidentemente o mesmo nome do autor pesquisado. A partir deste ponto, o usuário deve começar um processo de verificação dos resultados através do acesso aos *links* retornados pela consulta. Este processo, em muitos casos, é totalmente frustrante, seja pelo tempo despendido na averiguação das respostas, seja pela confirmação, após um longo tempo de navegação, que o que foi retornado não é relevante para o usuário.

O motivo principal para as máquinas de busca retornarem grandes quantidades de documentos não relevantes está na tecnologia que elas utilizam. São utilizadas, principalmente, indexação de termos e classificação de documentos, no intuito de facilitar a verificação da ocorrência ou não dos termos pesquisados na base de dados.

[1] www.altavista.com
[2] www.google.com
[3] www.yahoo.com

Por exemplo, um dos modelos mais usados para classificação de documentos, o Modelo Vetorial [10], não leva em consideração aspectos semânticos da consulta ou o perfil do usuário. Estas informações poderiam auxiliar no processo de recuperação da informação, fazendo com que os resultados retornados estivessem mais próximos daqueles que o usuário esperasse obter.

Como ressaltado acima, as consultas realizadas na Web por meio das máquinas de busca convencionais fornecem resultados que desconsideram o contexto no qual estão inseridas e as características do usuário em questão. Entretanto, têm surgido ultimamente propostas que estão levando este tipo de informação em consideração. Métodos de reformulação de consultas pesquisados na área de Recuperação de Informação [10][22], expandem automaticamente a consulta formulada pelo usuário, para que sejam recuperados, não apenas documentos que tenham os termos da consulta, mas também documentos relacionados indiretamente com a consulta elaborada. Outro método é a análise de apontadores [1]. Este define a importância do documento em função da quantidade de documentos que o referenciam ou são referenciados por ele.

Os sistemas para recuperação de informações tradicionais são compostos de três processos básicos: a coleta, a indexação e o cálculo de similaridade. O coletor é responsável pela coleta das informações disponíveis na Web e pela atualização de *links*. As informações são processadas através da criação de índice invertido das palavras pertencentes aos documentos nos quais elas ocorrem. O cálculo de similaridade é feito pela máquina de busca a partir de uma análise do conteúdo da consulta e das informações existentes nos arquivos de índice. Este processo retorna um resultado, levando em consideração a ocorrência de determinados termos em um documento. Os resultados são associados e classificados através de cálculos de medidas do cosseno [10]. Entretanto, o contexto em que a consulta está inserida, ou mesmo o perfil do usuário, que poderiam ser usados para melhorar a classificação dos documentos retornados, não são normalmente utilizados pela máquina de busca. Além disso, alguns aspectos das máquinas de busca que podem interferir negativamente no processo de obtenção de informação são:

- cobertura da máquina de busca: sua abrangência de atuação é limitada;
- situação dos *links*: são apresentados *links* desatualizados, relativos a páginas que podem ter mudado de enderêço ou sido removidas;
- disponibilidade de conteúdo: é disponibilizada apenas uma fração da Web de domínio público, e não são disponibilizados documentos com requisitos de

autenticação.

O objetivo deste trabalho é propor a implementação de uma ferramenta de busca utilizando técnicas de filtragem, capaz de auxiliar o usuário na recuperação de informações, de acordo com suas necessidades, de maneira ágil e eficiente. Com isso, espera-se diminuir consideravelmente a quantidade de URL's[4] a serem analisados pelo pesquisador, que, ao final, não tenham o conteúdo desejado por ele.

Foi construído um protótipo de uma ferramenta que recebe os termos da consulta, aplica o modelo vetorial [10] para cálculo e classificação dos documentos e submete tais documentos à filtragem, levando em consideração o perfil do usuário. Após este processo, os perfis são reformulados automaticamente, através do algoritmo de *word contribution* [15] [14], utilizando um *feedback* do usuário.

1.1 Trabalhos relacionados

O problema de filtragem de informação utilizando técnicas de recuperação vem sendo amplamente pesquisado pela comunidade de Ciência da Computação. A seguir, é feita uma breve revisão dos principais trabalhos nesta direção, assim como de outros trabalhos que originaram técnicas importantes que foram consideradas para uso neste projeto.

O trabalho apresentado por T. W. Yan [28] propõe uma ferramenta de filtragem chamada de SIFT (Stanford Information Filtering Tool). Essa ferramenta utiliza o modelo vetorial sobre um índice invertido de perfis para filtragem de documentos, uma vez que essa é a solução natural para se gerenciar uma grande quantidade de perfis sobre um fluxo de chegada de documentos de forma isolada. Nesse projeto, também é usado o modelo vetorial para recuperação dos documentos, sendo que o cálculo de similaridade acontece duas vezes: uma sobre o conjunto de termos da consulta e outra sobre o conjunto de termos do perfil.

R. Wilkinson [27] descreve os documentos através de conjuntos de termos derivados dos documentos. Com isso, é feita a indexação dos documentos. A partir desse ponto, é utilizada uma rede neural artificial para relacionar padrões de consulta sobre um grande número de conjuntos de termos. Através dessa organização é possível selecionar documentos relevantes levando em consideração um perfil estabelecido. Os algoritmos de aprendizagem de máquina no estilo de redes neurais artificiais chegaram a ser cogitados para uso neste projeto, mas não foram utilizados

[4] Uniforme Resource Locator.

porque não são eficientes para o problema em questão, uma vez que necessitariam de um número considerável de documentos para treinamento e aprendizado sobre determinado interesse do usuário, sempre que houvesse reformulação no perfil.

O método de classificação supervisionada *ID3 (Itemized Dichotomizer 3)* desenvolvido por Quinlan [19] [20] usa indução de atributos para geração de regras através da construção de uma árvore de decisão. Sua habilidade para operar dados não numéricos é facilmente aplicável para muitas situações. Segundo Han [9], esse método é o mais utilizado em aplicações de aprendizado indutivo. Essa seria uma outra abordagem para filtrar os documentos sobre os termos do perfil. Esse método chegou a ser experimentado; entretanto, a necessidade de treinamento para construção da árvore de decisão toda a vez que o perfil tivesse alteração tornaria o processo muito lento, necessitando uma grande capacidade de computação.

Em [5] uma técnica foi apresentada para classificar termos de acordo com a capacidade de discriminação entre documentos que fazem parte de uma coleção. Essa técnica chamada de análise de valor de discriminação atribui valor a um termo dependendo do tamanho da separação média entre documentos individuais quando o termo é atribuído a um documento. Os melhores termos são aqueles que conseguem as maiores separações. O sistema desenvolvido nesta dissertação também atribui valores aos termos dos documentos dependendo da contribuição dos termos para a relevância dos documentos.

O método *word contribution* [15] [14] calcula uma medida para expressar a influência de uma palavra na similaridade entre consulta e documentos. Esta medida representa a contribuição da palavra para a relevância do documento. Todas as palavras que têm peso positivo acima de um limiar estabelecido são adicionadas ao perfil do usuário. Palavras com grande contribuição positiva são palavras que ocorrem na consulta e nos documentos. Esse algoritmo foi usado para reformulação do perfil do usuário.

T. F. A. Jesini [12] propõe um sistema para filtragem de notícias disponíveis eletronicamente. O sistema recupera documentos relevantes a um perfil de interesse e o melhora automaticamente a partir de informações vindas dos usuários. O sistema de filtragem faz uso do modelo vetorial para recuperar notícias e utiliza o algoritmo de Rocchio para realimentação do perfil. Essa seria uma outra abordagem para a reformulação do perfil do usuário. Entretanto, após examinar o trabalho publicado por K. Hoashi[14], onde é feita uma comparação de desempenho entre os dois métodos, optou-se por usar o método de contribuição de palavras por se mostrar mais eficiente.

Bruce Krulwich [16] ressalta que um dos aspectos mais importantes na tarefa de filtragem está na modelagem do perfil do usuário. Este processo é geralmente implementado através de técnicas de reformulação de perfil utilizando documentos relevantes. Entretanto, qual informação do texto pode ser utilizada para a reformulação? Esta pergunta é crucial para este processo. A utilização de todo o texto permite que termos irrelevantes sejam usados para compor o novo perfil, prejudicando-o na seleção de documentos efetivamente relevantes. Entretanto, em nosso trabalho a quantidade de palavras inseridas no perfil dependerá de sua importância, conforme resultados obtidos na aplicação do método de contribuição de palavras [15] [14].

1.2 Contribuições

Este trabalho apresenta uma proposta para classificação de documentos relevantes levando em consideração o perfil do usuário. A principal contribuição desta dissertação é justamente a proposta de um modelo para filtragem de informação utilizando perfil do usuário. Neste modelo, aplica-se o método vetorial sobre os termos da consulta e os termos do perfil. Este processo combina duas propostas: a do modelo vetorial tradicional e a do modelo vetorial com filtro proposto por Persin[11]. Essa última, permite uma redução significativa do volume de memória principal necessária durante o cálculo da similaridade, mantendo um resultado satisfatório para as respostas das consultas.

A reformulação do perfil é feita baseada no método de contribuição de palavras, que representa a influência de um termo para a similaridade entre a consulta e um documento através de valores quantitativos. Quanto maior é o valor da contribuição do termo, maior é sua importância para a relevância do documento.

Resultados experimentais realizados para avaliar o sistema de filtragem indicam que, em ambiente onde são possíveis o controle e interação com o usuário, o método proposto pode ser uma alternativa viável para implementação de sistemas para recuperação de informação.

Os resultados experimentais obtidos com a simulação do sistema de filtragem, mostram que a realização da reformulação do perfil melhora consideravelmente a qualidade dos documentos recuperados. A frequente utilização do sistema de filtragem com sucessivas consultas e reformulações, tende a melhorar significativamente seu desempenho.

1.3 Estrutura da dissertação

O próximo capítulo apresenta os principais conceitos relacionados com recuperação de informação e filtragem, necessários para o entendimento do trabalho. O terceiro capítulo apresenta a arquitetura do sistema de filtragem, o algoritmo para reformulação de perfil baseado no método de contribuição de palavra. O quarto capítulo apresenta os resultados obtidos. E, finalmente, o quinto capítulo apresenta as conclusões e propostas de trabalhos futuros.

Capítulo 2

Filtragem de Informações

Neste capítulo são descritos os modelos de recuperação de informação, assim como o funcionamento de um sistema de filtragem genérico. As diferenças em relação ao sistema proposto nesta dissertação são também abordadas. Em seguida, são especificadas as métricas para avaliação do desempenho do sistema implementado.

2.1 Noções gerais

Podemos descrever um sistema de filtragem de informações como sendo um mecanismo automático com a capacidade de monitorar um fluxo contínuo de documentos e habilidade para selecionar documentos considerados relevantes para um determinado usuário ou grupos de usuários, levando em consideração suas necessidades. Estas necessidades são representadas através de um perfil de interesse associado ao usuário ou ao grupo de usuários. A habilidade para selecionar documentos relevantes está associada a mecanismos de recuperação de informações que calculam o valor da similaridade entre os documentos da coleção e os perfis. Os documentos de grande similaridade com o perfil são considerados relevantes para o usuário ou grupo de usuários. Entretanto, para que o perfil possa acompanhar as mudanças de comportamento de um usuário ou grupo de usuários, deve ser possível fazer atualizações apropriadas no mesmo. Estas atualizações são feitas através de informações enviadas para o sistema sobre a relevância dos documentos recebidos.

O sistema de filtragem proposto nesta dissertação apresenta um aspecto diferente em relação ao sistema de filtragem genérico, uma vez que ele pretende recuperar documentos relevantes para uma determinada consulta. Os termos da consulta darão o escopo de atuação do filtro de informação. Esse filtro pretende melhorar a classificação dos documentos previamente recuperados levando em consideração o

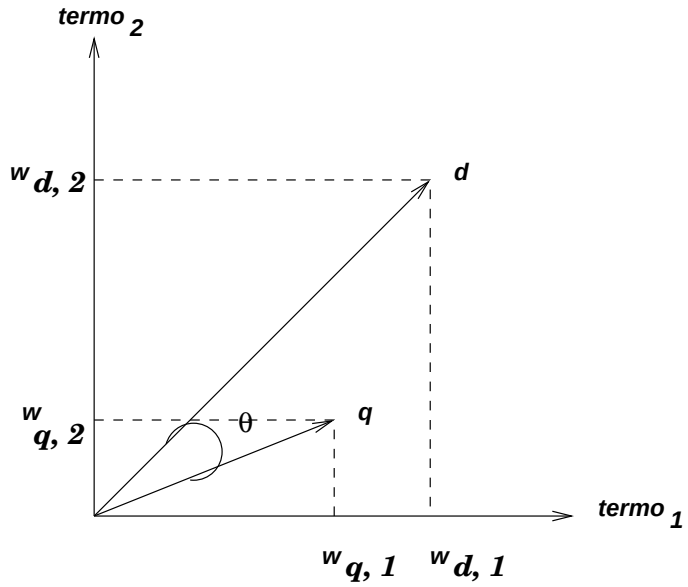


Figura 2.1: Representação de vetores de consulta e documento em um espaço de 2 dimensões.

perfil, e não adicionar novos documentos para satisfazerem as necessidades do perfil.

2.2 Modelo vetorial

O Modelo Vetorial utiliza um vetor de termos para representar a freqüência com a qual cada termo aparece no documento. Esse modelo tem bom desempenho na recuperação de documentos porque leva em consideração o casamento parcial e a proximidade dos documentos em relação aos termos da consulta [22] [10].

No modelo vetorial, documentos e consultas são representados através de vetores de termos dentro de um espaço vetorial, conforme mostrado na Figura 2.1. A dimensão deste espaço é dada pelo número de termos distintos da coleção e as coordenadas dos vetores são determinadas pela importância de cada documento da coleção para cada termo da consulta.

Um documento d_j é representado por um vetor $\vec{d_j} = (w_{j,1}, w_{j,2}, \ldots, w_{j,n})$, onde n representa o número de termos distintos da coleção e $w_{j,t}$ é o peso do termo t no documento d_j. A consulta é representada por um vetor de consulta $\vec{q} = (w_{q,1}, w_{q,2}, \ldots, w_{q,n})$, onde $w_{q,t}$ é o peso do termo t na consulta q. O perfil é representado por um vetor de perfil $\vec{p} = (w_{p,1}, w_{p,2}, \ldots, w_{p,n})$, onde $w_{p,t}$ é o peso do termo t no perfil p. A similaridade entre um documento d_j e uma consulta q ou perfil p é definida como uma correlação entre os vetores $\vec{d_j}$ e $\vec{q}$ ou $\vec{d_j}$ e $\vec{p}$. Essa similaridade é medida através do cálculo do cosseno entre os dois vetores. A equação

abaixo mostra a fórmula para cálculo da similaridade entre um documento d_j e um consulta q, através do cosseno do ângulo entre os vetores $\vec{d_j}$ e $\vec{q}$:

$$sim(\vec{d_j}, \vec{q}) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{t=1}^{n} w_{d_j,t} w_{q,t}}{\sqrt{\sum_{t=1}^{n} w_{d_j,t}^2} \sqrt{\sum_{t=1}^{n} w_{q,t}^2}}.$$

Técnicas de normalização são empregadas com o objetivo de evitar que documentos grandes sejam beneficiados em detrimento de documentos pequenos. Nessa equação,

$$|\vec{d_j}| = \sqrt{\sum_{t=1}^{n} w_{d_j,t}^2} \text{ e } |\vec{q}| = \sqrt{\sum_{t=1}^{n} w_{q,t}^2}$$

são os fatores de normalização do documento d_j e da consulta q, respectivamente. O fator $|\vec{q}|$ não afeta a similaridade, pois é o mesmo para todos os cálculos. Já $|\vec{d_j}|$ deve ser calculado para cada documento d_j; da coleção. Quando os vetores são normalizados, o valor da similaridade se mantém entre 0 e 1: o valor 0 como indicação de nenhuma similaridade, e o valor 1 como uma indicação de similaridade máxima. A normalização dos documentos de uma coleção evita que documentos grandes sejam favorecidos em detrimento de documentos pequenos.

Os pesos dos termos nos documentos, na consulta ou no perfil podem ser calculados de várias maneiras, dependendo das características da coleção e do tipo de recuperação desejada. Duas medidas bastante utilizadas são: aquelas que consideram o número total de ocorrências do termo nos documentos, e as que consideram o número de ocorrências do termo em cada documento.

O número de vezes que um termo ocorre em um documento pode ser considerado como uma importante medida para a importância do termo para aquele documento. Essa informação é conhecida como *tf* (*term frequency*). O *tf* de um termo em um documento é representado pela fórmula abaixo, onde $f_{d_j,t}$ representa o número de vezes que o termo t ocorre no documento d_j:

$$tf_{d_j,t} = 1 + \log f_{d_j,t}.$$

O número de documentos em que um termo ocorre também pode ser utilizada para se quantificar a importância deste termo para a coleção. Normalmente, quanto mais raro for o termo, mais importante é o fato deste termo ter ocorrido em um documento. Em contrapartida, quanto mais um termo ocorre em uma coleção, menos importante ele se torna. O fator que indica a influência de um termo na coleção é chamado de freqüência inversa de um termo *idf* (*inverse document frequency*). O *idf*

1. Criar um vetor de acumuladores A_j.
2. Para cada termo t da consulta:
   recuperar a lista invertida de t no disco;
   para cada $\langle d_j, f_{d_j,t} \rangle$ da lista invertida:
   inicializar A_j, se o mesmo já não tiver sido inicializado;
   $A_j \leftarrow A_j + simparc_{q,d_j,t}$.
3. Dividir cada acumulador $A_j > 0$ pelo comprimento $|\vec{d_j}|$.
4. Identificar os k maiores valores de acumuladores e recuperar os documentos respectivos.

Figura 2.2: Cálculo de medida do cosseno.

de um termo é dado pela fórmula abaixo, onde N é o número total de documentos da coleção e f_t é o número de documentos em que o termo t ocorre:

$$idf_t = log(1 + \frac{N}{f_t}).$$

Conhecidos os valores de tf e idf de um termo t, a fórmula proposta por Salton [5] para obter o peso do termo t em um documento ou consulta x é dada pela equação:

$$w_{x,t} = tf_{x,t} \times idf_t.$$

O algoritmo para processamento de consultas usado na recuperação e classificação dos documentos de acordo com o modelo vetorial, utiliza um acumulador A_j para cada documento d_j da coleção, no qual são armazenados os resultados da similaridade. Para cada termo t, é adicionada a A_j a similaridade parcial $simparc_{q,d_j,t}$, dada por:

$$simparc_{q,d_j,t} = w_{d_j,t} \times w_{q,t}.$$

É importante ressaltar que essas expressões levam em conta uma série de questões: a freqüência do termo dentro do documento ($f_{d_j,t}$), a freqüência do termo na coleção (f_t), o tamanho dos documentos ($|\vec{d_j}|$), e o número de documentos na coleção (N). Dessa forma, se um termo aparece muito em um documento, isso o torna relevante. Termos muitos freqüentes na coleção, como é o caso de *stopwords*, têm menor importância. Além disso, documentos muito extensos, que poderiam ser favorecidos por conter muitos termos, são normalizados através do fator $|\vec{d_j}|$.

O algoritmo para cálculo de medida do cosseno está mostrado na Figura 2.2

2.3 Um modelo vetorial com filtro

A técnica proposta por Michael Persin [11] para a filtragem de documentos permite uma redução significativa do volume de memória principal necessária durante o cálculo do *ranking*, mantendo um resultado satisfatório para as respostas às consultas. Nesta técnica, os acumuladores são inseridos ou apenas atualizados dependendo de dois parâmetros de similaridade: um global, C_{ins}, e outro local, C_{add}. A similaridade global está relacionada com a raridade do termo na coleção, enquanto que a similaridade local está relacionada com a frequência do termo no documento. Desta forma, toda a lista invertida de um termo é analisada, mas uma entrada para um determinado documento só modifica o acumulador se for capaz de melhorar a ordem de classificação do documento.

Inicialmente, é criado um vetor de acumuladores e os termos da consulta são ordenados em ordem decrescente do seu *idf* . Com isso, os termos mais importantes são processados primeiro. Em seguida, são calculados os dois limiares, f_{ins} e f_{add}. O limiar f_{ins} estabelece um critério para a criação de um novo acumulador e o limiar f_{add} estabelece um critério para a inclusão do valor da similaridade parcial em um acumulador. Esses valores são computados por:

$$f_{ins} = \frac{C_{ins} \times S_{max}}{f_{q,t} \times w_t^2} \text{ e } f_{add} = \frac{C_{add} \times S_{max}}{f_{q,t} \times w_t^2}.$$

Esses dois fatores são utilizados na filtragem para rejeitar ou aceitar um documento d_j pertencente a uma lista invertida do termo t, dependendo de: freqüência do termo na coleção (f_t), quantidade de ocorrências do termo no documento ($f_{d_j,t}$), e importância do termo na consulta ($f_{q,t}$).

Quando a lista invertida de um termo t é processada, a similaridade parcial $simparc_{q,d_j,t}$ entre a consulta q e o documento d_j em relação ao termo t é comparada aos fatores de inserção e adição. Se $f_{d_j,t} \geq f_{ins}$ o documento d_j é considerado importante. Se o acumulador deste documento ainda não existir, ele é criado e inicializado com 0. A similaridade parcial para este documento será acumulada ao valor existente. Caso contrário, Se $f_{d_j,t} \geq f_{add}$ o termo t influi pouco para a relevância do documento d_j, mas pode afetar em sua classificação final. Assim, a similaridade parcial $simparc_{q,d_j,t}$ somente é adicionada se d_j já possuir um acumulador. Caso contrário, a lista invertida para este termo é descartada. Em seguida S_{max} recebe o maior valor entre S_{max} e o valor para o acumulador desse documento. Finalmente, os valores dos acumuladores são normalizados através da divisão pela norma dos documentos $|\vec{d_j}|$. O Algoritmo para cálculo de medida do cosseno com filtro está

1. Criar um vetor de acumuladores A_j.
2. Ordenar os termos da consulta por valor de *idf*.
3. Inicializar $S_{max} = 0.0$.
4. Para cada termo t da consulta:
   (a) computar os valores de f_{ins} e f_{add}
   (b) recuperar a lista invertida de t no disco;
   (c) para cada $\langle d_j, f_{d_j,t} \rangle$ da lista invertida:
      i. Se $f_{d_j,t} \geq f_{ins}$ inicializar A_j se necessário
         $A_j \leftarrow A_j + simparc_{q,d_j,t}$.
      ii. Senão, se $f_{d_j,t} \geq f_{add}$ e existe A_j,
         $A_j \leftarrow A_j + simparc_{q,d_j,t}$.
      iii. $S_{max} \leftarrow max(S_{max,A_j})$
5. Dividir cada acumulador $A_j > 0$ pelo comprimento $|\vec{d_j}|$.
6. Identificar os k maiores valores de acumuladores e recuperar os documentos respectivos.

Figura 2.3: Cálculo de medida do cosseno com filtro.

mostrado na Figura 2.3.

As constantes C_{ins} e C_{add} são utilizadas para ajustar o algoritmo. Aumentando-se a constante C_{add}, reduz-se o número de entradas processadas para um termo. Em conseqüência, reduz-se a quantidade de documenntos a ser processada, diminuindo o tempo de processamento. Aumentando-se a constante C_{ins}, reduz-se o número de documentos que podem ser considerados. Conseqüentemente, reduz-se a utilização de memória. As constantes devem ser escolhidas de tal forma que as informações descartadas tenham influência mínima na classificação final.

2.4 Administração do perfil do usuário

A administração do perfil consiste na parte mais sensível do sistema de filtragem, pois é a partir do estado atual do perfil que o sistema determina quais informações são relevantes para determinado usuário. O conteúdo atual do perfil é utilizado pelo sistema no cálculo da similaridade entre o perfil e os documentos retornados pelo sistema de classificação. As atualizações no perfil, com base no comportamento do usuário, têm como objetivo assimilar os interesses típicos do usuário. Sendo essa atualização continuada, ela proporciona uma adaptação gradual e constante ao perfil do usuário. Nesta seção serão apresentados os fundamentos teóricos sobre os quais foram implementados o módulo para administração de perfil.

O processo para criação de um perfil que expresse o interesse de um usuário é

complexo e depende de características que representem um usuário, como por exemplo, preferências pessoais, grau de conhecimento, vocabulário, informações sobre o conteúdo da coleção, conhecimento sobre o tipo de índice utilizado, etc. Essas características conferem ao perfil um contexto subjetivo que pode ser utilizado pelas máquinas de busca. Entretanto, estas informações são normalmente desprezadas pelas máquinas de busca.

O trabalho desenvolvido em [15][14], descreve a técnica de contribuição de palavras que calcula a influência de cada palavra para a similaridade entre a consulta e o documento. Esta técnica provê uma medida que visa capturar a importância de cada palavra para a relevância de um documento relativamente a uma consulta. De acordo com esta técnica, cada palavra apresenta uma contribuição positiva ou negativa. Uma palavra com contribuição positiva alta tem uma maior similaridade com os termos da consulta. Em contrapartida, uma palavra com grande contribuição negativa tem baixa similaridade com os termos da consulta. O administrador de perfil implementado nesta dissertação reformula dinamicamente o perfil do usuário baseado nesta técnica.

2.5 O método de contribuição de palavras

O método de contribuição de palavras (WC) foi utilizado nesse trabalho por se tratar de um método eficiente para a reformulação de perfil e expansão de consultas. No artigo [15] é feita uma comparação do desempenho entre esse método e o algoritmo de Rocchio, onde os resultados apresentados mostram que a utilização do método de contribuição de palavras é mais eficiente para a solução dessa classe de problemas.

O método de contribuição de palavras (WC) procura quantificar a influência de uma palavra na similaridade entre a consulta e um documento. Em termos matemáticos, pode-se descrever a contribuição de uma palavra através da fórmula:

$$Cont(w, q, d_j) = Sim(q, d_j) - Sim(q'(w), d'_j(w))$$

onde $Cont(w, q, d_j)$ é a contribuição da palavra w para a similaridade entre a consulta q e o documento d_j, $Sim(q, d_j)$ é a similaridade entre a consulta q e o documento d_j, e $Sim(q'(w), d'_j(w))$ é a similaridade entre a consulta q e o documento d_j, excluindo a palavra w da consulta q e do documento d_j ($q'(w)$ é a consulta q, excluída a palavra w, e $d'_j(w)$ é o documento d_j, excluída a palavra w). Pode-se

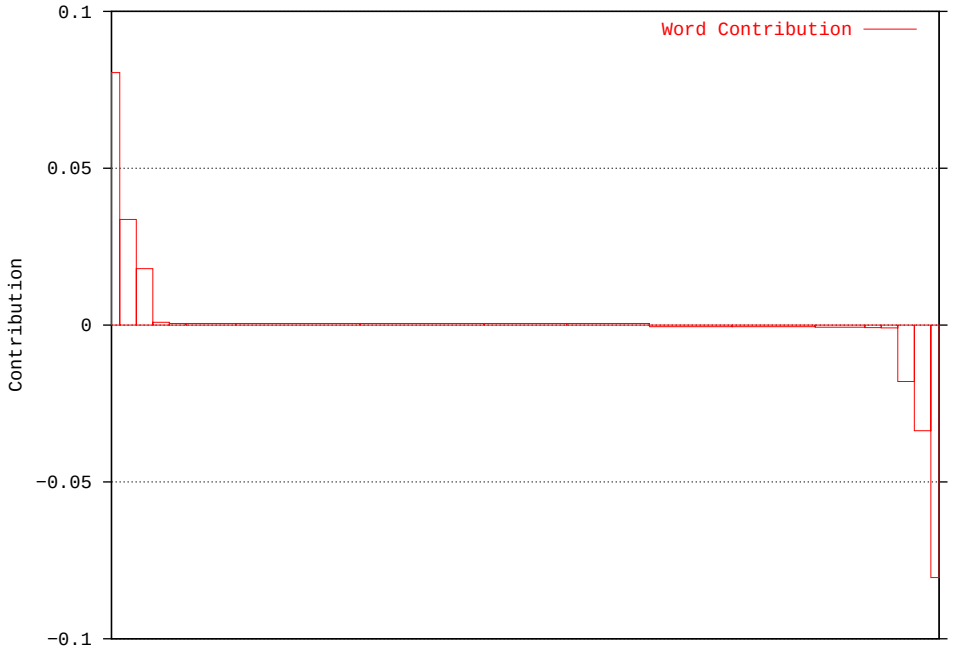


Figura 2.4: Comportamento da contribuição das palavras.

definir a contribuição da palavra w como sendo a diferença entre as similaridades da consulta q e do documento d_j, desconsiderando a palavra w na consulta e no documento.

Conforme mencionado por Hoashi [15] [14] e mostrado na Figura 2.4, a maioria dos termos de um documento tem contribuição próxima de zero para a relevância do documento para a consulta. Uma quantidade pequena tem influência negativa e uma outra pequena quantidade de termos tem contribuição positiva para a relevância dos documentos. Os termos com contribuição positiva, ou parte deles, são utilizados pelo módulo de reformulação para a atualização do perfil.

2.5.1 Calculando a contribuição das palavras da consulta e do documento

Inicialmente é calculada a contribuição de todas as palavras pertencentes à consulta e ao documento através do método WC apresentado anteriormente. As palavras w com menores valores de $Cont(w, q, d_j)$ serão utilizadas para o cálculo do $Score$ através da fórmula:

$$Score(w) = wgt \times \sum_{d_j \in Drel(q)} Cont(w, q, d_j)$$

onde wgt é uma constante cujo ajuste depende das características de cada coleção, e $Drel(q)$ são os documentos considerados relevantes pelo usuário.

Armazenar perfis que representem o interesse de um usuário é considerado uma tarefa complicada, embora de grande importância. Na prática, não existe fórmula para a criação de perfil, mas existem métodos que possibilitam a manutenção de perfil que representam de forma aceitável os interesses de um usuário, baseados na interação do sistema de filtragem com o usuário. Considerando uma consulta q, suponha que sejam recuperados e enviados para o usuário k documentos relevantes para esta consulta, e que, o usuário examina essa lista, ou parte dela, e identifica quais documentos são relevantes. Com base nesta informação e na consulta original, é possível produzir uma atualização no perfil.

2.6 Métricas de avaliação

Nesta seção são apresentadas algumas métricas para avaliação do desempenho de sistemas de filtragem disponíveis na literatura. São descritas as medidas de precisão e revocação e a medida de utilidade.

2.6.1 Precisão e revocação

Documentos relevantes para uma consulta são aqueles semanticamente relacionados com os termos da consulta. As medidas de precisão e revocação representam o desempenho do sistema baseando-se nos números de documentos relevantes que foram recuperados, e como esses documentos foram classificados entre os documentos recuperados pelo sistema. A precisão identifica os documentos relevantes entre os documentos recuperados pelo sistema; enquanto que a revocação identifica a quantidade de documentos relevantes recuperados dentre todos os documentos relevantes existentes na coleção. Essas medidas, em geral, são usadas em sistemas que apresentam suas respostas como um conjunto ordenado de documentos. A avaliação normalmente é feita observando-se a evolução da precisão em função da revocação, representada através de uma curva de precisão-revocação. Como o funcionamento do nosso sistema de filtragem está fundamentado no perfil individual de cada usuário, passando por processos de reformulação, usaremos estas medidas para avaliar o comportamento do sistema de filtragem comparando-se com o desempenho de uma consulta aplicando-se o modelo vetorial.

As expressões para calcular a precisão e a revocação são:

$$Precisão = \frac{num\ de\ docs\ recuperados\ relevantes}{num\ total\ de\ docs\ recuperados}, \text{ e}$$

$$Revocação = \frac{num\ de\ docs\ relevantes\ recuperados}{num\ total\ de\ docs\ relevantes}.$$

2.6.2 Utilidade

A medida de utilidade avalia um sistema de filtragem baseado no fato dele ter classificado um documento em um dos quatros conjuntos a seguir: o conjunto de documentos relevantes recuperados, o conjunto de documentos relevantes não recuperados, o conjunto de documentos não relevantes recuperados e o conjunto de documentos não relevantes não recuperados. A medida de utilidade atribui a cada documento um custo, baseado no fato dele ter sido classificado em uma das quatro categorias mencionadas anteriormente. A equação abaixo representa a fórmula geral de utilidade:

$$Utilidade = A \times R_+ + B \times N_+ + C \times R_- + D \times N_-$$

Nessa equação, as variáveis R_+, R_-, N_+, N_-, se referem aos números de documentos em cada categoria, respectivamente: o conjunto de documentos relevantes recuperados, o conjunto de documentos relevantes não recuperados, o conjunto de documentos não relevantes recuperados e o conjunto de documentos não relevantes não recuperados. Os parâmetros de utilidade (A, B, C, D) determinam os pesos de cada categoria. Um valor positivo para um parâmetro da utilidade indica um valor a favor de um documento que foi classificado naquela categoria, enquanto que um valor negativo indica uma penalização a um documento por ter sido classificado em determinada categoria. Assim, quanto maior for o valor da utilidade, melhor o sistema de filtragem estará desempenhando seu papel para um dado perfil.

Capítulo 3

Implementação do Sistema de Filtragem

Neste capítulo, discutiremos a arquitetura do sistema de filtragem implementado, a qual está esquematizada na Figura 3.1. O sistema de filtragem funciona da seguinte forma. Primeiro, é feito o cadastramento do usuário através do módulo de cadastramento. Durante o cadastramento, o usuário efetua a autenticação através do módulo de autenticação. Após a autenticação, o usuário submete para o sistema uma requisição de consulta através do módulo de consulta. Esse módulo recebe e processa a requisição retornando uma lista ordenada com os documentos relevantes. Com base nesses documentos o módulo de filtragem recupera os termos do perfil do usuário e efetua o cálculo da similaridade entre os termos do perfil e dos documentos retornados pela consulta. O resultado é um lista classificada. Ao receber o resultado do sistema de filtragem o usuário avalia os resultados e envia os documentos relevantes para o módulo de reformulação do perfil. Esse módulo recebe os documentos transmitidos pelo usuário, recupera a consulta original e aplica o algoritmo de contribuição de palavras entre os documentos e a consulta, armazenando no perfil do usuário os termos que mais contribuíram para a relevância dos documentos. A seguir é descrito cada um dos componentes.

3.1 Cadastramento

O usuário acessa o sistema e seleciona a opção cadastramento de novo usuário, conforme mostrado no caso de *Cadastramento* exibido na Figura 3.2. O sistema exibe a tela solicitando as informações do usuário. Estas informações são verificadas e utilizadas para a criação do perfil inicial do usuário. Após a conclusão do

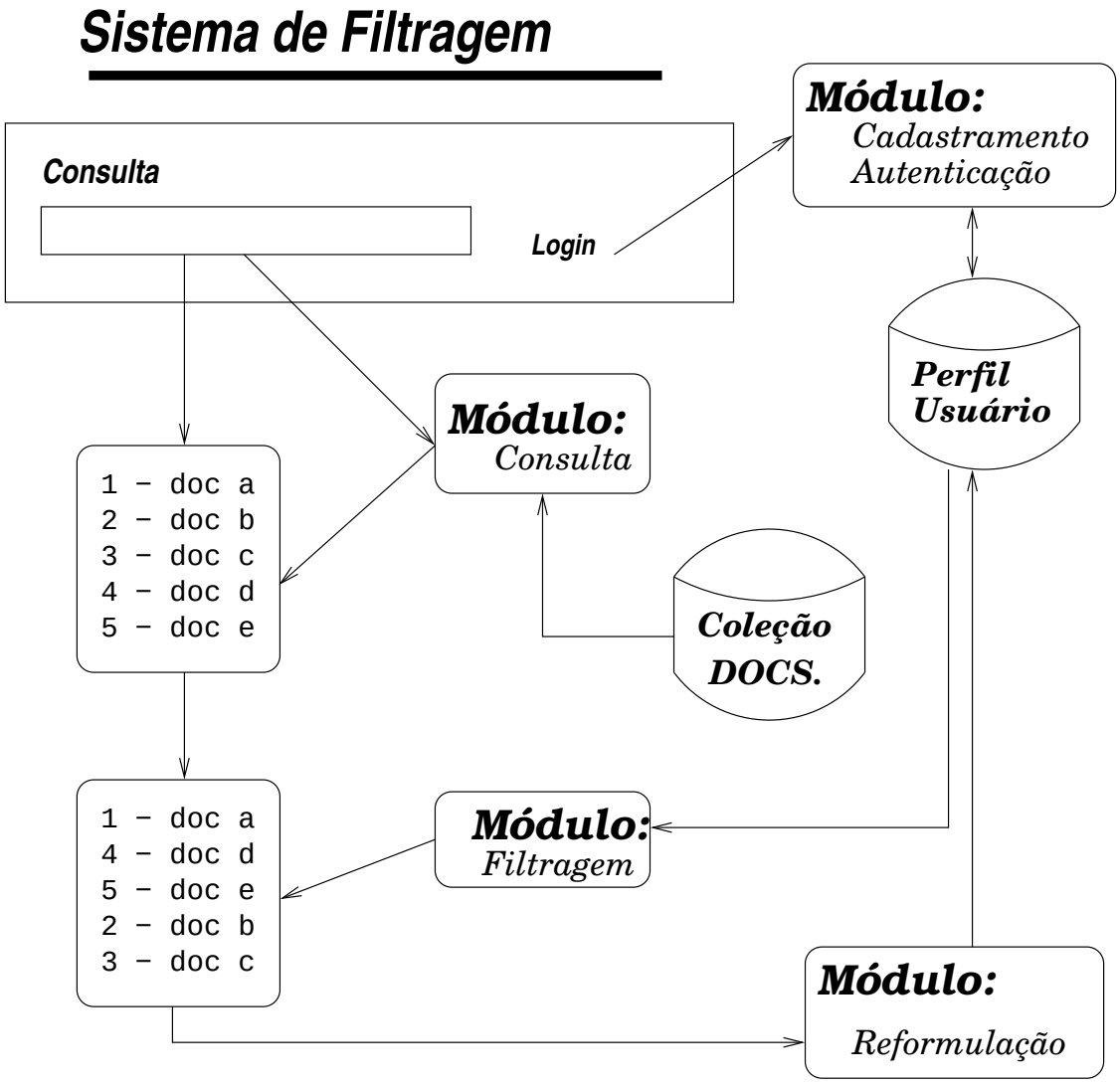


Figura 3.1: Arquitetura para o sistema de filtragem.

Objeto: Cadastramento.
Autores: Usuário.
Pré-Condições: 1. Novo usuário do sistema seleciona a opção Cadastrar.
Eventos: *try* 1. Sistema solicita informações do usuário. 2. Usuário fornece as informações. 3. Sistema verifica as informações. 4. Sistema faz o login automático. 5. Sistema recupera o perfil do usuário. 6. Sistema exibe interface para consulta. *except*: 1. Caso o cadastro esteja incompleto, as informações são solicitadas novamente.
Pós-condições: 1. Cadastramento e login efetuados com sucesso.

Figura 3.2: Caso de uso *Cadastramento*.

cadastramento o usuário tem a opção de utilizar o sistema ou sair.

3.2 Autenticação do usuário

O módulo de autenticação do usuário, mostrado pelo caso de uso *Login* exibido na Figura 3.3, foi desenvolvido com o objetivo de controlar o perfil de cada usuário cadastrado no sistema. Consiste de uma base de dados *MySQL*[25] onde são armazenadas informações que descrevem o perfil de cada usuário. Essas informações são submetidas a um programa que é responsável por processá-las e verificá-las na base de dados. Caso o usuário tenha um perfil registrado no sistema, o mesmo será recuperado para posterior utilização. Caso o usuário não tenha um perfil estabelecido, o sistema pode auxiliá-lo na criação de um perfil inicial chamando a interface para cadastramento.

3.3 Módulo de consulta

O módulo de consulta do sistema de filtragem, mostrado pelo caso de uso *Consulta* exibido na Figura 3.4, foi desenvolvido mantendo-se o *layout* tradicional das máquinas de busca existentes na *Web*, com uma interface simples e de fácil utilização. A interface consiste de uma caixa de texto, onde o usuário pode fornecer a

Objeto: Login.
Autores: Usuário.
Pré-Condições: Usuário do sistema.
Eventos: *try* 1. Sistema solicita login e senha. 2. Usuário fornece login e senha. 3. Sistema verifica as informações. 4. Sistema faz o login automático. 5. Sistema recupera o perfil do usuário. 6. Sistema exibe interface para consulta. *except*: 1. Login e senha inválidas. 2. Caso o cadastro esteja incompleto, as informações são solicitadas novamente.
Pós-condições: 1. Login efetuado com sucesso.

Figura 3.3: Caso de uso *Login*.

expressão para a consulta, a qual pode conter qualquer quantidade de termos. A interpretação da consulta é realizada utilizando-se o método de disjunção dos termos, como interpretados tradicionalmente pelas máquinas de busca. O sistema transmite os termos da consulta para os mecanismos de busca que calcula a similaridade parcial entre os termos da consulta e os termos dos documentos da coleção, retornando uma lista de documentos em ordem decrescente de relevância.

3.4 Módulo de filtragem

O sistema de filtragem funciona da seguinte forma. Primeiro, é feito o cadastramento do usuário através do módulo de cadastramento. A partir do cadastramento, o usuário efetua a autenticação através do módulo de autenticação. Após a autenticação, o usuário submete para o sistema uma requisição de consulta através do módulo de consulta. Esse módulo recebe e processa a requisição utilizando o modelo vetorial, retornando uma lista ordenada com os documentos relevantes. Com base nesses documentos, o módulo de filtragem recupera os termos do perfil do usuário e efetua a filtragem. No processo de filtragem é feito o cálculo da similaridade entre os termos do perfil e dos documentos retornados pela consulta. O resultado é uma lista classificada. Ao receber o resultado do sistema, o usuário avalia os resultados e envia os documentos relevantes para o módulo de reformulação do perfil. Esse

Objeto: Consulta.
Autores: Usuário cadastrado no sistema.
Pré-Condições: 1. Usuário do sistema que acessa a URL do sistema via Web. 2. Usuário efetua o login no sistema.
Eventos: *try* 1. Sistema solicita consulta através de termos. 2. Usuário fornece os termos da consulta. 3. Sistema faz a leitura dos termos e transmite para o sistema. 4. Sistema executa a consulta vetorial na base de dados do sistema e retorna uma lista. 5. Sistema executa a consulta dos documentos retornados utilizando o perfil do usuário. 6. Sistema classifica os dois resultados da consulta. 7. Sistema exibe o resultado da consulta. *except*: 1. Caso o usuário não tenha um perfil cadastrado no sistema o processo de filtragem não é executado.
Pós-condições 1. Usuário visualiza os resultados da consulta.
Feedback 1. Usuário transmite informações sobre documentos relevantes ou não relevantes.

Figura 3.4: Caso de uso *Consulta*.

módulo recebe os documentos transmitidos pelo usuário, recupera a consulta original e aplica o método de contribuição de palavras entre os documentos e a consulta, armazenando no perfil do usuário os termos que mais contribuíram para a relevância dos documentos.

3.5 Mecanismo para reformulação do perfil

O processo de reformulação do perfil está baseado na contribuição das palavras da consulta e nos documentos avaliados como relevantes pelo usuário. Os documentos relevantes relativos a uma consulta q são representados através de um vetor de documentos:

$$Drel(q) =< d_0, d_1, ..., d_{N-1} > .$$

Inicialmente, calcula-se a contribuição de todas as palavras da consulta e dos documentos relevantes utilizando-se a fórmula:

$$Cont(w, q, d_j) = sim(q, d_j) - sim(q'(w), d'_j(w)).$$

Para cada documento relevante d_j, k palavras com baixa contribuição são extraídas. Hoashi em seu paper [15] descreve que a quantidade k depende da natureza da coleção.

Para cada palavra w extraída do documento d_j é calculado um peso através da fórmula:

$$score(w) = wgt \times \sum_{d_j \in Drel(q)} Cont(w, q, d_j)$$

O parâmetro wgt é utilizado para transformar termos importantes que tem contribuição negativa em contribuição positiva, para facilitar os cálculos de similaridade.

Conforme observado em [15], a quantidade de termos com menor contribuição extraída do documento relevante, depende do tamanho médio e das características dos documentos da coleção. No artigo escrito por [14], foi desenvolvido vários testes para se achar os melhores valores para wgt. Os que tiveram os melhores desempenhos foram -200, -400 e -800. Apresentaremos no capítulo 4, testes com estes valores, no cálculo da contribuição de palavra para a nossa coleção.

3.5.1 Processo de reformulação

Conforme Krulwich [16], para que o sistema de filtragem "aprenda" sobre o in-

teresse do usuário, é necessário que o sistema extraia sentenças semanticamente significativas. Para isto, são analisados os documentos avaliados como relevantes pelo usuário. Nesta análise são interpretados os termos do documento que possam ter contribuído para que o documento seja relevante. Entre eles destacamos os termos com alto valor para o *idf*, termos escritos em negrito, itálico ou termos escritos em letras maiúsculas.

O processo de realimentação é executado no recebimento do *feedback* do usuário. Esse, informa o código do perfil, a consulta e os documentos relevantes.

Essas informações são então processadas, recuperando o perfil do usuário, os termos da consulta e os documentos relevantes. As contribuições dos termos da consulta e dos documentos relevantes são calculadas através da análise dos termos conforme [15] [14]. Os termos com menor valor WC são considerados para avaliação. Para esses termos são calculados *Scores* e definido um *limiar* conforme as características da coleção. Todos os termos que tiverem *Score* acima do *limiar* são utilizados para a reformulação do perfil.

O perfil do usuário é armazenado em um banco de dados relacional através do MySQL [24], representado por três tabelas conforme diagrama $E - R$ mostrado na Figura 3.5. A tabela Perfil, armazena as informações gerais do usuário, como nome, login, email etc. A tabela Termos-Reformulados, armazena os termos inseridos através do processo de reformulação do perfil. A tabela Termos-Consulta, armazena os números e termos das consultas executadas pelo usuário. Essa última tabela é utilizada pelo Sistema de Filtragem na reformulação do perfil.

3.6 Desenvolvimento do projeto

A recuperação de informação está fundamentada no processo de coleta de informação através de agentes, indexação de coleções através de listas de arquivos invertidos, consulta e classificação através de Máquinas de Busca. Na Figura 3.1 é mostrada a arquitetura do sistema de filtragem proposta nessa dissertação.

Tradicionalmente, sistemas de recuperação de informação utilizam termos de indexação para a recuperação de informação. Termos de indexação são palavras ou grupos de palavras que tenham significado próprio. De uma forma geral, um termo de indexação é simplesmente uma palavra que aparece no texto dos documentos da coleção.

O processamento da consulta no sistema de filtragem é executado em duas fases, descritas a seguir.

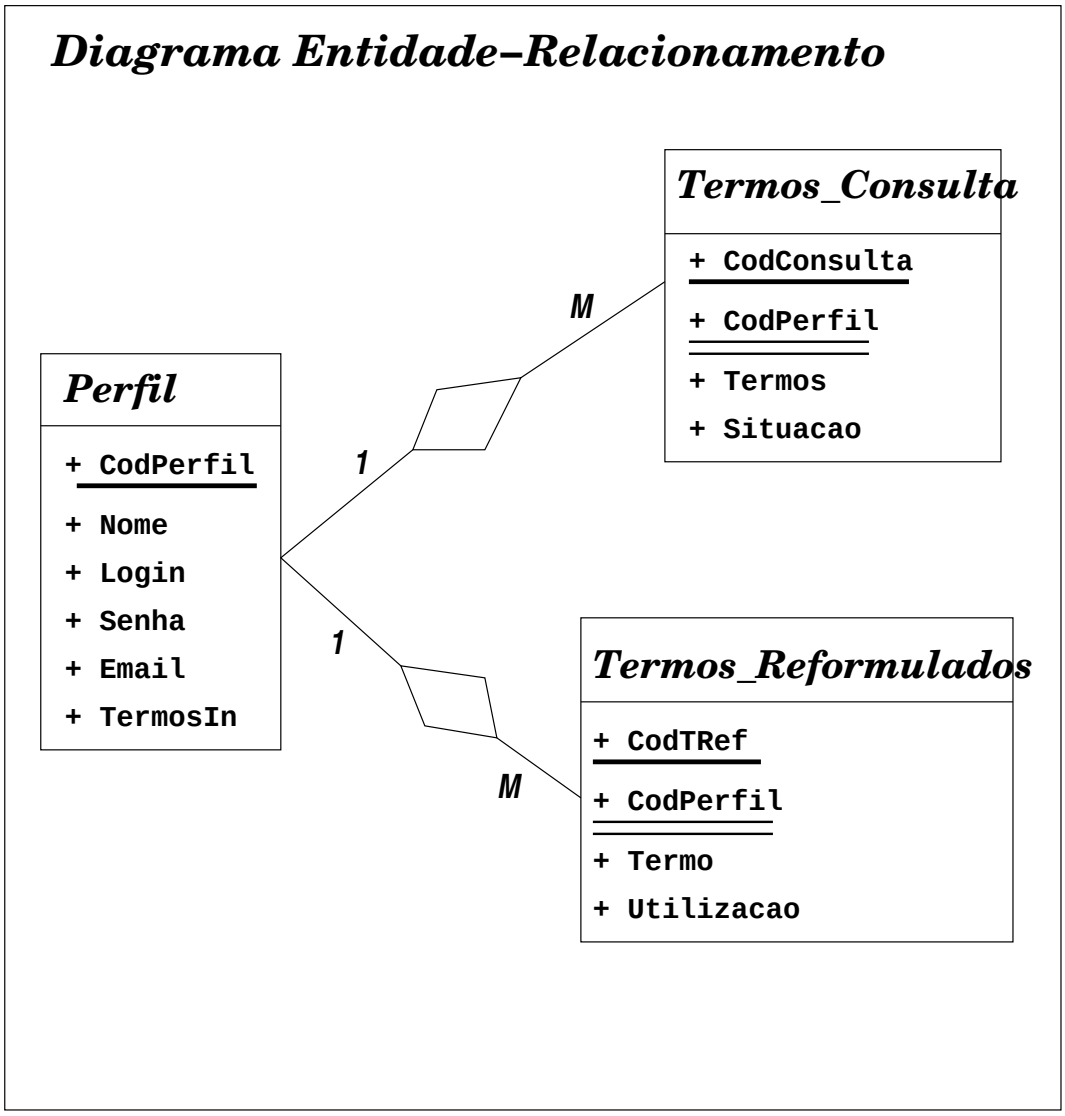


Figura 3.5: Diagrama E-R para representação do perfil

A primeira fase consiste na aplicação do modelo vetorial para calcular a similaridade dos documentos da coleção com os termos da consulta. O resultado obtido é uma lista de documentos classificados de forma decrescente pelo valor da similaridade. Serão utilizados no processo de filtragem os k primeiros documentos, formando uma subcoleção de documentos. O valor de k depende das características da coleção.

Os valores das similaridades dos documentos utilizados pelo módulo de filtragem são guardados para serem acumulados com os resultados da segunda parte do processo de filtragem.

Na segunda fase, é utilizado o modelo vetorial para cálculo da similaridade entre os termos do perfil do usuário sobre os documentos retornados na primeira fase da filtragem. A classificação dos documentos é feita através da soma das similaridades obtidas nas duas fases do processo de filtragem.

O resultado final retornado pelo sistema de filtragem é uma lista de documentos decrescentemente ordenada com base nos valores de similaridades dos termos da consulta, do perfil e os documentos da coleção.

Ao analisar os documentos transmitidos pelo sistema, o usuário retorna um *feedback* que será utilizado para a reformulação do perfil, aplicando-se o algoritmo de contribuição de palavras conforme [13][15][14].

3.6.1 Exemplo de funcionamento do sistema de filtragem

O exemplo abaixo descreve, sucintamente, o funcionamento do sistema de filtragem através do processo de indexação para criação de uma lista invertida, consulta e filtragem utilizando o modelo vetorial e o processo de reformulação do perfil.

O arquivo de entrada para o qual se deseja construir o índice é um arquivo típico da coleção *CACM*, conforme exemplificado na Figura 3.6. Esse arquivo é dividido em documentos lógicos, contendo informações sobre informática e delimitados pelas marcações *SGML* $<DOC>$ *e* $</DOC>$.

Como exemplo, apresentamos na Figura 3.7 uma coleção simplificada formada por 26 termos distintos (representados por números de 0 a 25), distribuidos em 16 documentos. Por exemplo, o documento D_7 contém duas ocorrências do termo 1 e duas do termo 4.

<DOC>
<DOCNO> 1 </DOCNO>
<TEXT>
Preliminary Report-International Algebraic
Language
Perlis, A. J.
Samelson,K.
</TEXT>
</DOC>
<DOC>
<DOCNO> 2 </DOCNO>
<TEXT>
Accelerating Convergence of Iterative Processes
A technique is discussed which, when applied
to an iterative procedure for the solution of
an equation, accelerates the rate of convergence if
the iteration converges and induces convergence if
the iteration diverges. An illustrative example
is given.
Wegstein, J. H.
</TEXT>
</DOC>
<DOC>
<DOCNO> 3 </DOCNO>
<TEXT>
Techniques Department on Matrix Program Schemes
Friedman, M. D.
</TEXT>
</DOC>
<DOC>
<DOCNO> 4 </DOCNO>
<TEXT>
Glossary of Computer Engineering and Programming
Terminology
</TEXT>
</DOC>

Figura 3.6: Exemplo da estrutura da coleção $CACM$.

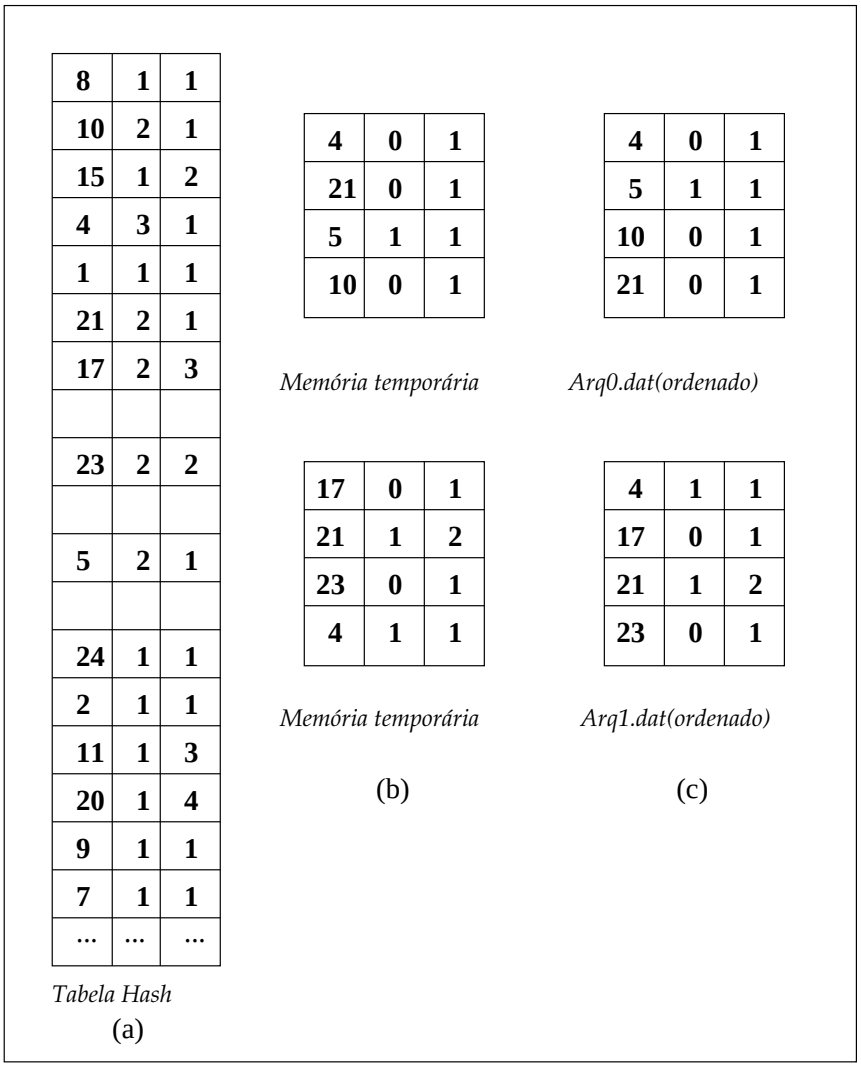
$d_0 = \{4, 10, 17, 21, 23\}$
$d_1 = \{1, 2, 4, 5, 7, 8, 9, 11, 11, 11, 15, 15, 20, 20, 20, 20, 21, 21, 24\}$
$d_2 = \{0, 3, 3, 3, 5, 6, 10, 17, 17, 17, 21, 23, 23, 25\}$
$d_3 = \{4, 5, 10, 17, 17, 23\}$
$d_4 = \{0, 3, 3, 7, 7, 9, 11, 11, 16, 16, 16, 16, 19, 19, 19, 19, 19\}$
$d_5 = \{1, 4\}$
$d_6 = \{7, 7, 7, 8, 9, 10, 11\}$
$d_7 = \{1, 1, 4, 4\}$
$d_8 = \{15, 15, 18, 18, 20, 21\}$
$d_9 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 9, 10, 12, 14, 15, 16, 17, 18, 18, 19, 21, 22, 23, 25\}$
$d_{10} = \{7, 8, 9, 10\}$
$d_{11} = \{11, 11, 17, 17, 17, 19, 19\}$
$d_{12} = \{4, 4, 10, 10, 17, 17, 21, 21, 23, 23, 23\}$
$d_{13} = \{5\}$
$d_{14} = \{10, 10, 11, 12, 13, 15\}$
$d_{15} = \{9, 9, 9, 15, 15, 18, 18, 20, 21\}$

Figura 3.7: Coleção simplificada.

3.6.2 Indexação da coleção

A indexação é feita através da leitura dos documentos da coleção. Para cada termo é gerada uma tupla $\langle t, d_j, f_{d_j,t} \rangle$, que identifica o termo, o documento e o número de ocorrências do termo no documento. Essa tupla é armazenada em uma tabela *hash*. A tabela *hash* foi adotada por apresentar eficiência no custo de pesquisas e simplicidade de implementação. (A Figura 3.8(a) mostra partes de uma tabela *hash*, após a leitura dos documentos d_0 ao d_2 desconsiderando alguns termos. Por exemplo, a entrada $< 8, 1, 1 >$ significa que o termo 8 ocorreu 1 vez no documento 1). Quando um termo de um novo documento é encontrado, os valores armazenados anteriormente na tabela *hash* são transferidos para uma memória temporária, dando lugar a esse novo termo. Em nosso exemplo, usamos a memória temporária para armazenar quatro tuplas, conforme Figura 3.8(b). Quando o espaço de memória temporária se esgota, as informações são ordenadas e armazenadas em arquivos temporários com a mesma estrutura da memória temporária, conforme arq0.dat e arq1.dat, mostrado na Figura 3.8(c).

Ao final da leitura de todos os documentos da coleção é construído um arquivo invertido aplicando o *Multiway Merging* [10] através do algoritmo de ordenação *heapsort*. Esse algoritmo funciona da seguinte forma: são lidas para o *heap* as primeira tuplas de cada arquivo temporário. Essas tuplas são ordenadas, sendo



...

Figura 3.8: Construção do arquivo invertido.

armazenado no arquivo invertido a tupla com o menor termo e documento e a próxima tupla é lida para o *heap*. Esse processo se repete até que todos as tuplas tenham sido lidas e ordenadas. A Tabela 3.1 está representando o conteúdo de um arquivo invertido para alguns termos da coleção simplificada, onde as tuplas são ordenadas primeiramente por termos, em seguida por documentos.

3.6.3 Cálculo do idf

Como exemplo, é mostrado o cálculo do *idf* dos termos 0, 1 e 2, utilizando a

t	1	2	4	4	4	5	5	7	8	9	10	10	11	15	17	17	20	...
j	1	1	0	1	3	1	2	1	1	1	0	2	1	1	0	2	1	...
$f_{d_j,t}$	1	1	1	1	1	1	1	1	3	1	1	1	3	2	1	3	4	...

Tabela 3.1: Conteúdo do arquivo invertido.

t	0	1	2	3	4	5	6	7	8	9	10	11	12
ft	3	5	2	3	7	5	2	5	4	6	8	6	3
idf	1,8	1,6	2,2	1,8	1,2	1,4	2,2	1,4	1,6	1,3	1,1	1,3	1,8
t	13	14	15	16	17	18	19	20	21	22	23	24	25
ft	1	2	5	2	6	2	4	3	6	1	5	1	2
idf	2,8	2,2	1,44	2,2	1,3	2,2	1,61	1,84	1,3	2,83	1,44	2,83	2,2

Tabela 3.2: Valor IDF para cada termo da coleção.

Fórmula 3.1 mostrada abaixo. O termo 0 ocorre 3 vezes na coleção, o termo 1 ocorre 5 vezes e o termo 2 ocorre 2 vezes para um total de 16 documentos. Os valores dos idf são mostrados na Tabela 3.2.

$$idf_t = \log(1 + \frac{N}{f_t}). \tag{3.1}$$

$$idf_0 = \log(1 + \frac{16}{3}) = 1,84.$$

$$idf_1 = \log(1 + \frac{16}{5}) = 1,61.$$

$$idf_2 = \log(1 + \frac{16}{2}) = 2,2.$$

A seguir é mostrado como utilizar a Fórmula 3.2 para calcular a norma dos documentos 0 e 3 para a coleção simplificada. A Tabela 3.3 mostra os valores das Normas para todos os documentos do exemplo.

$$|\vec{d_j}| = \sqrt{\sum_{t=1}^{n}((1 + \log f_{d_j,t}) \times w_t)^2}. \tag{3.2}$$

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\|\vec{d_j}\|$	2,8	7,7	6,8	3,4	8,4	2	4	3,4	5,1	9,1	2,8	4,4	5,1	1,4	4,4	5,3

Tabela 3.3: Valor da norma para cada documento da coleção.

$$|\vec{d_0}| = \sqrt{(1,19)^2 + (1,1)^2 + (1,3)^2 + (1,3)^2 + (1,44)^2} = \sqrt{8,46} = 2,84.$$

$$|\vec{d_3}| = \sqrt{(1,19)^2 + (1,44)^2 + (1,1)^2 + ((1,69) \times (1,3))^2 + (1,44)^2} = \sqrt{11,6} = 3,4.$$

3.6.4 Cálculo da similaridade

É mostrado a seguir, o cálculo da similaridade entre os termos da consulta e os documentos da coleção utilizando a Fórmula 3.3 mostrada abaixo. O resultado da similaridade é um conjunto de documentos, classificados em ordem decrescente de acordo com sua relevância. Esses documentos são utilizados no cálculo da similaridade com os termos do perfil.

$$simparc_{q,d_j,t} = w_{d_j,t} \times w_{q,t}. \tag{3.3}$$

Tomaremos como exemplo a consulta com os termos 1,4 e 13 na coleção descrita acima, para um usuário que tenha um perfil representado através dos termos 5, 8, 12 e 14.

Cálculo da similaridade parcial para o **termo 1**, que ocorre nos documentos 1, 5, 7 e 9:
$simparc_{q,d_1,1} = 1,60/7,7 = 0,208.$
$simparc_{q,d_5,1} = 1,60/2,0 = 0,804.$
$simparc_{q,d_7,1} = (1,693 \times 1,60)/3,4 = 0,804.$
$simparc_{q,d_9,1} = 1,60/9,1 = 0,175.$

Cálculo da similaridade parcial para o **termo 4**, que ocorre nos documentos 0, 1, 3, 5, 7, 9 e 12:
$simparc_{q,d_0,4} = 1,19/2,8 = 0,418.$
$simparc_{q,d_1,4} = 1,19/7,7 = 0,154 + 0,208 = 0,362.$
$simparc_{q,d_3,4} = 1,19/3,4 = 0,349.$
$simparc_{q,d_5,4} = 1,19/2,0 = 0,594 + 0,804 = 1,398.$

Classificação	*Doc*	Similaridade Consulta
1	d_5	1,398
2	d_7	1,398
3	d_{14}	0,647
4	d_0	0,418
5	d_{12}	0,390
6	d_1	0,362
7	d_3	0,349
8	d_9	0,305

Tabela 3.4: Documentos retornados em ordem de classificação.

$simparc_{q,d_7,4} = (1,693 \times 1,19)/3,4 = 0,594 + 0.804 = 1,398.$
$simparc_{q,d_9,4} = 1,19/9,1 = 0,175 + 0,175 = 0,305.$
$simparc_{q,d_{12},4} = (1,693 \times 1,19)/5,1 = 0,39.$

Cálculo da similaridade parcial para o **termo 13**, que ocorre no documento 14:
$simpar_{q,d_{14},13} = 2,83/4,4 = 0,647.$

A Tabela 3.4 mostra os documentos relevantes retornados para esta consulta, na ordem de classificação.

3.6.5 Cálculo da similaridade do perfil para os documentos relevantes

Cálculo da similaridade parcial para os termos do perfil com termos 5, 8, 12 e 14, que ocorrem nos documentos 0, 1, 3, 5, 7, 9, 12, 14 retornados pela consulta, conforme Tabela 3.5.

Cálculo da similaridade parcial para o **termo 5**, que ocorre nos documentos 9 e 13:
$simpar_{q,d_1,5} = 1,44/7,7 = 0,19.$
$simpar_{q,d_3,5} = 1,44/3,4 = 0,423.$
$simpar_{q,d_9,5} = 1,44/9,1 = 0,16.$

Cálculo da similaridade parcial para o **termo 8**, que ocorre nos documentos 1 e 9:
$simpar_{q,d_1,8} = 1,61/7,7 = 0,39 + 0,19 = 0,58.$
$simpar_{q,d_9,8} = (1,693 \times 1,61)/9,1 = 0,30 + 0,16 = 0,46.$

Classificação	*Doc*	Similaridade consulta	Similaridade perfil	Similaridade
1	d_5	1,398		1,398
2	d_7	1,398		1,398
3	d_{14}	0,647	0,42	1.07
4	d_0	0,418		0,418
5	d_{12}	0,390		0,390
6	d_1	0,362	0,58	0,94
7	d_3	0,349	0,423	0,77
8	d_9	0,305	0,90	1,21

Tabela 3.5: Tabela com os resultados parciais.

Classificação	*Doc*	Similaridade
1	d_5	1,398
2	d_7	1,398
3	d_9	1,21
4	d_{14}	1,07
5	d_1	0,94
6	d_3	0,77
7	d_0	0,418
8	d_{12}	0,39

Tabela 3.6: Documentos retornados após a filtragem.

Cálculo da similaridade parcial para o **termo 12**, que ocorre nos documentos 9 e 14:
$simpar_{q,d_9,12} = 1,84/9,1 = 0,20 + 0,46 = 0,66.$
$simpar_{q,d_{14},12} = 1,841/4,4 = 0,42.$

Cálculo da similaridade parcial para o **termo 14**, que ocorre no documento 9:
$simpar_{q,d_9,14} = 2,2/9,1 = 0,24 + 0,66 = 0,90.$

A Tabela 3.6 mostra os documentos relevantes retornados para esta consulta após o filtro, na ordem de classificação.

Capítulo 4

Análise de Resultados Experimentais

Neste capítulo, discutiremos os experimentos realizados com o sistema de filtragem implementado neste trabalho. Com a finalidade de realizarmos testes, foi desenvolvido um protótipo com o qual executamos simulações de indexações, consultas e reformulações de perfis. Para a realização dos experimentos usamos duas coleções conforme mostra a Tabela 4.1: a *FCCCMG*, elaborada com a participação dos alunos da *Faculdade de Ciência da Computação de Caratinga*, e a coleção padrão *CACM*, disponível no laboratório *LATIN - Laboratório para Tratamento da Informação do DCC/UFMG*. Simulamos as interações dos usuários com perfis individuais, nas quais os mesmos forneceram termos para consultas, receberam documentos relevantes, analisaram e retornaram os documentos mais importantes para a reformulação do perfil. Com os resultados obtidos para uma consulta foi feita uma avaliação utilizando as medidas de *utilidade* e *precisão-revocação*.

Nas seções seguintes, são descritas as coleções usadas para validar o sistema de filtragem, os experimentos que foram conduzidos, os resultados obtidos e as conclusões a que chegamos com base nesses resultados.

Coleção	FCCCMG	CACM
Documentos	1.750	3.204
Consultas	197	12
Termos	8729	7.121
Média Tam. Docs	48,45	24,26
Perfis	35	12
Tamanho Coleção	940 Kbytes	1,8 Mbytes

Tabela 4.1: Coleções usadas para experimentos.

4.1 Coleta de documentos para a formação de uma coleção

Existem várias formas de construção de uma coleção de documentos para ser usada por sistemas de filtragem em geral. Ela pode ser construída dinamicamente ou utilizando fragmentos de grandes coleções estáticas de documentos. Em um ambiente de produção dinâmica de informação, como é o que ocorre na *Web*, uma coleção pode ser criada através de coleta ativa, com agentes autônomos percorrendo a *Web*, ou de forma passiva e através do recebimento de informações por serviços de distribuição, ou alguma combinação dessas duas formas. A maioria dos sistemas de filtragem foram desenvolvidos para filtrar informação proveniente de serviços de distribuição de informação [18][26]. Outros trabalhos avaliam seus métodos de filtragem realizando experimentos sobre grandes coleções, como a coleção TREC [13] e a REUTERS [21]. Como a quantidade de informação disponível na *Web* tem aumentado consideravelmente, a coleta ativa tem se tornado gradativamente mais importante. Essa técnica tem ganhado popularidade porque vem sendo usada no desenvolvimento de máquinas de busca. As empresas que mantêm portais de busca comerciais estão investindo muito nessa tecnologia, uma vez que a coleta automática de documentos é fundamental para diversas aplicações na Web. A coleta ativa é implementada a partir da construção de um coletor automático de documentos, responsável por coletar um subconjunto de documentos de uma coleção que satisfaçam a uma determinada especificação, armazenando cópias locais destes documentos. As cópias locais dos documentos coletados deverão ser utilizadas para a geração de índices que agilizem a recuperação quando solicitados.

4.2 Coleções usadas pelo sistema

A coleção $FCCCMG$ foi criada com a participação de 35 alunos da Faculdade de Ciência da Computação de Caratinga. É composta por documentos extraídos dos provedores de busca Todobr[1], Google[2], e Cadê[3] sobre os assuntos cinema, culinária, educação, esporte, informática, política, saúde, segurança e turismo. A escolha desses provedores e dos tópicos foi feita pelos alunos da Faculdade, durante 60 dias. As regras para a criação da coleção foram:

[1] www.todobr.com.br
[2] www.google.com
[3] www.cade.com.br

- cada aluno tinha um assunto para pesquisa;
- as consultas só podiam ocorrer nos provedores descritos acima;
- cada aluno recuperou em média 50 documentos;
- cada aluno classificou os 10 documentos mais relevantes de acordo com a sua preferência.

Este processo de coleta e classificação individual de documentos se deu para atender à exigência do problema, uma vez que o filtro tem por objetivo melhorar a classificação do resultado do modelo vetorial, considerando o perfil do usuário.

Como a coleta das informações ocorreu em um ambiente onde os documentos não tinham uma organização estrutural, uma vez que se tratavam de documentos *HTML*, foi necessária a criação de um *script* em *PHP* que fizesse uma leitura de todos os arquivos e os transformassem em documentos com marcações *SGML*. Exemplo de marcação *SGML* <*DOC*> </*DOC*>.

A coleção é composta por *1750* documentos, os quais fazem referência a *8729* termos distintos. Cada documento tem em média *48,45* termos. O espaço ocupado é de *940 Kbytes* de armazenamento em disco, conforme mostrado na Tabela 4.1.

Para uma melhor avaliação do desempenho do sistema proposto nesse trabalho, utilizamos a coleção padrão *CACM*, disponível no laboratório *LATIN - Laboratório para Tratamento da Informação.*

Como os documentos dessa coleção são divididos em documentos lógicos, delimitados pelas marcações *SGML*, não foi necessário fazer nenhuma transformação. Outra vantagem é que existe para esta coleção um conjunto de consultas e seus respectivos documentos relevantes.

Conforme mostrado na Tabela 4.1, essa coleção é composta por *3204* documentos, que fazem referência a *7121* termos distintos. Cada documento tem em média *24,26* termos. O espaço ocupado é de *1,8 Mbytes* de armazenamento em disco.

Depois das informações terem sido coletadas e classificadas, elas foram indexadas através do módulo de indexação, gerando um índice dos documentos dessa coleção. Este índice foi utilizado nos experimentos para o sistema de filtragem.

4.3 Métodos de Avaliação

Usamos as métricas de precisão-revocação, conforme descrito no capítulo 2, para avaliar os resultados das consultas utilizando a coleção $CACM$. Alguns trabalhos na

área de filtragem de informação como Hoashi [15], não acham adequada a utilização dessa medida de avaliação, como forma de validação, por falta de conhecimento prévio dos documentos relevantes. Entretanto, para este trabalho, essas métricas são importantes, uma vez que temos os resultados esperados para a coleção CACM classificados pelos especialistas que montaram a coleção. Essa avaliação é feita levando em consideração a curva de precisão dos 11 pontos de revocação. Esta curva mostra a evolução em 11 pontos padrões de revocação entre 0 e 100.

A medida de utilidade, como definida no capítulo 2, foi utilizada para avaliar os resultados das consultas utilizando a coleção $FCCCMG$. Essa medida fornece um valor associado a quatro categorias importantes. Entretanto, só temos conhecimento de duas categorias:

- o conjunto dos documentos relevantes recuperados;
- o conjunto dos documentos não relevantes recuperados.

As coleções TREC-7 e TREC-8 [3] [7] [8] apresentam resultados da medida de utilidade, utilizando vários valores como parâmetros de utilidade, entre eles, 3 e 2 apresentaram resultados muito satisfatórios; onde 3 é utilizado para o conjunto de documentos relevantes recuperados e 2 é utilizado para o conjunto de documentos relevantes não recuperados, conforme Fórmula 4.1. Testamos também os parâmetos com valores absolutos iguais para os dois conjuntos, conforme Fórmula 4.2, com o objetivo de avaliar o comportamento dos dois resultados.

$$Utilidade = (3) \times R_+ - (2) \times N_+ \tag{4.1}$$

$$Utilidade = (3) \times R_+ - (3) \times N_+ \tag{4.2}$$

Por exemplo, atribuindo para o parâmetro A o custo 3, e para o parâmetro B o custo -2. Significa dizer que, na avaliação, um usuário ganha três pontos para cada documento relevante recuperado e perde dois pontos para cada documento relevante não recuperado.

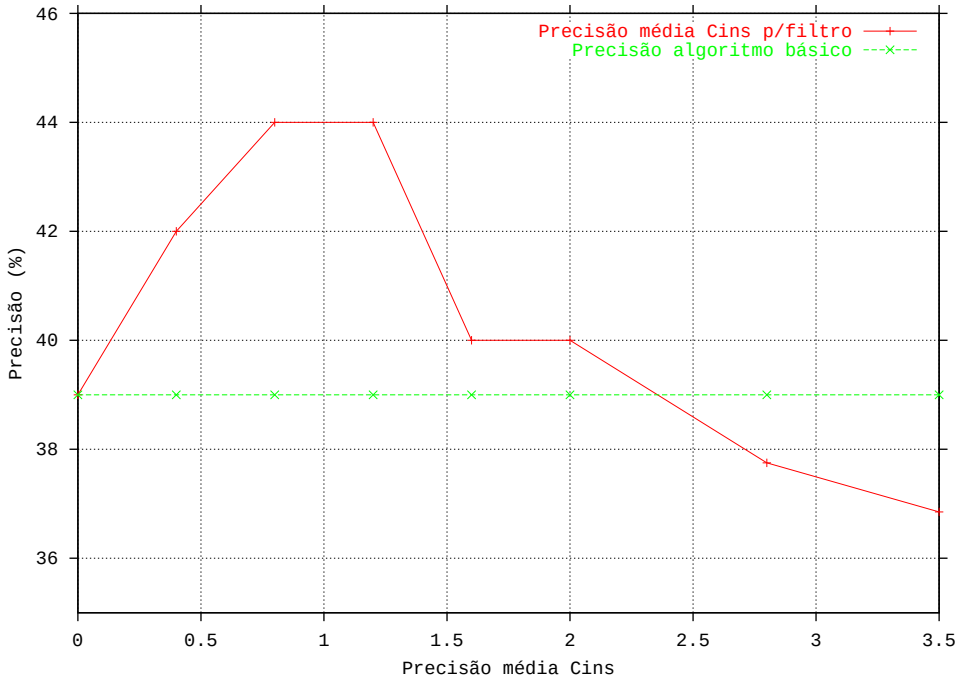


Figura 4.1: Precisão média para diferentes valores de C_{ins} e $C_{add} = 0$.

4.4 Ajustes das constantes utilizadas no módulo vetorial com poda e no módulo de contribuição de palavras

Os testes feitos inicialmente foram realizados para ajustar os valores das constantes C_{ins} e C_{add}, responsáveis pelo controle de inserção e adição de acumuladores, utilizados pelo modelo vetorial para cálculo da similaridade entre consulta e documentos da coleção. Primeiramente, os valores de ambas as constantes foram iniciados com *zero* e o valor de precisão foi 39. Esse valor foi tomado como base para o ajuste das constantes C_{ins} e C_{add}. Em seguida, a cada execução, o valor de C_{ins} foi incrementado de $0,4$. A precisão começava a subir um pouco e, a partir de um certo valor começava a cair. A constrante C_{ins} teve seu valor fixado no ponto anterior ao qual a precisão começou a ficar menor do que o valor da precisão tomado como base. A Figura 4.1 mostra o gráfico da precisão em função de C_{ins}. O valor fixado para essa constante foi de $2,8$.

Com o fim do processo de especificação da constante C_{ins}, o mesmo procedimento foi feito para ajustar o valor de C_{add}. A constante de adição teve seu valor incrementado de $0,1$ a cada execução. A precisão crescia um pouco e então iniciava a queda, conforme mostrado na Figura 4.2. O valor fixado para essa constante foi

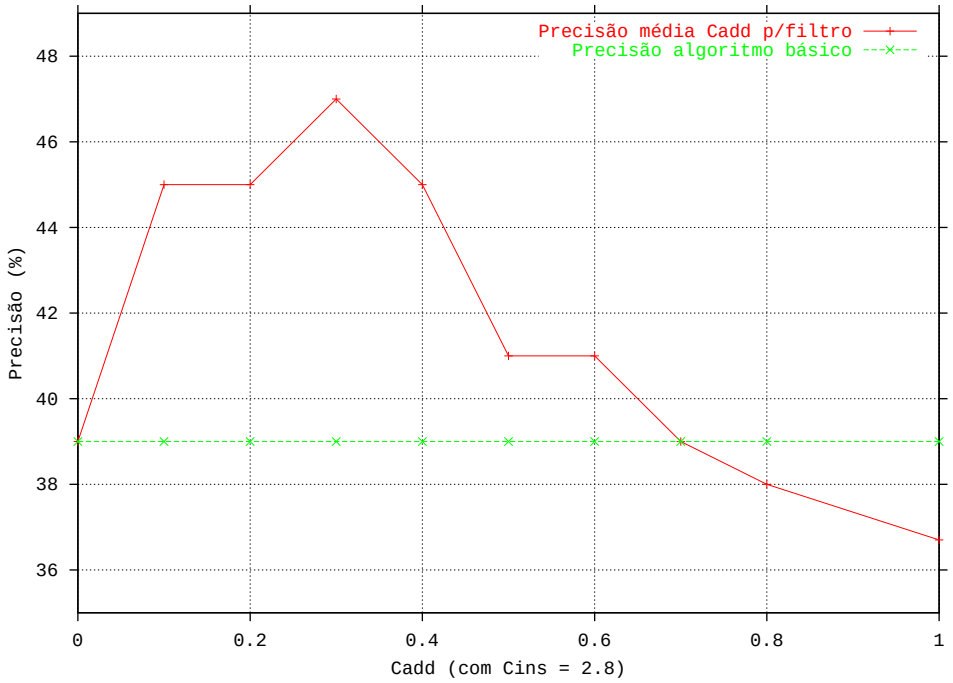


Figura 4.2: Precisão média para diferentes valores de C_{ins} e $C_{add} = 2.8$.

de $0,7$.

Foram realizados vários testes para definir os valores das constantes wgt e $WCpoda$, envolvidas na extração dos termos de menor contribuição dos documentos retornados pelo usuário e no aproveitamento dos termos para a reformulação do perfil, respectivamente. Conforme observado em [14], a quantidade de termos com menor contribuição extraídos do documento relevante depende do tamanho médio dos documentos da coleção. Como os testes foram feitos em coleções pequenas, extraímos 5, 10 e 20 termos com menores contribuições, dos documentos relevantes. Conforme sugerido por [15], os valores utilizados para wgt foram -200, -400 e -800. Após avaliação das constribuições dos termos da coleção, os valores utilizados para $WCpoda$ foram 2.435, 4.876 e 9.961. Na Tabela 4.2 é mostrada a porcentagem média de termos com contribuições superiores ao limiar $WCpoda$. Levando em considerações os resultados descritos na Tabela 4.2, foram extraídos 10 termos com menores contribuições de cada documento relevante. Foram utilizados para a reformulação do pefil os valores -400 e 4.876, para wgt e $WCpoda$, respectivamente. Esses valores foram utilizados por mostrarem os melhores desempenhos entre os valores testados.

wgt	*WCpoda*	5 docs.	10 docs.	20 docs.
-200	2,435	28%	32%	29%
-400	4,876	31%	34%	30%
-800	9,611	29%	31%	28%

Tabela 4.2: Precisão média para contribuições de palavras.

4.5 Metodologia

Para os experimentos, utilizamos duas coleções de documentos: as coleções $FCCCMG$ e $CACM$, conforme Tabela 4.1. Para a coleção $FCCCMG$, os testes foram feitos via comunicação eletrônica *(e-mail)*, pois nosso sistema não foi desenvolvido para a plataforma *Web*. Os alunos submetiam as consultas através do *e-mail*. Essas consultas eram inseridas no sistema o qual retornava os documentos relevantes. O resultado era enviado para o usuário via *e-mail*. Para efeito de simplificação, só foram considerados os dez primeiros documentos. Ao receber a mensagem com os documentos relevantes, o aluno verificava a qualidade das respostas de acordo com seu interesse pessoal. Essa avaliação era feita quantificando os documentos relevantes recebidos e os documentos não relevantes recebidos. Após a avaliação, o usuário enviava uma mensagem informando a quantidade de documentos recebidos que eram relevantes e a quantidade de documentos recebidos que não eram relevantes. Os documentos considerados relevantes pelo usuário foram utilizados pelo sistema para a reformulação do perfil. Os testes foram conduzidos durante 60 dias, onde foram submetidas 197 consultas, com média de $5,6$ consultas por usuário. Foram executadas 591 reformulações nos perfis dos usuários, com média de $16,88$ reformulações por usuário. Os resultados obtidos são mostrados nas Tabelas 4.3 e 4.4. Na Tabela 4.3 estão mostrados a média do número de documentos relevantes e não relevantes recuperados em um conjunto de 10 documentos avaliados pelo usuário, onde DR1 $= (2,48)$ e DN1 $= 3,42$, por exemplo, indicam que, ao serem analisados 10% de todas as 198 consultas, $2,48$ documentos foram avaliados como relevantes e $3,42$ foram avaliados como não relevantes (e, portanto, $10 - (2,48 + 3,42) = 4,1$ não foram avaliados). Segue o significado de cada campo da Tabela 4.3:

- DR1: quantidade de documentos relevantes com perfis vazios.
- DN1: quantidade de documentos não relevantes com perfis vazios.
- DR2: quantidade de documentos relevantes, após a primeira reformulação dos

perfis, repetindo-se a mesma consulta.

- DN2: quantidade de documentos não relevantes, após a primeira reformulação dos perfis, repetindo-se a mesma consulta.
- DR3: quantidade de documentos relevantes, após várias reformulações dos perfis, repetindo-se a mesma consulta.
- DN3: quantidade de documentos não relevantes, após várias reformulações dos perfis, repetindo-se a mesma consulta.
- DR: quantidade de documentos relevantes, após várias reformulações dos perfis, com novas consultas.
- DN: quantidade de documentos não relevantes, após várias reformulações dos perfis, com novas consultas.

Na Tabela 4.4 estão mostrados:

- Util DRN1: utilidade entre documentos relevantes e não relevantes, dos 10 documentos avaliados, para as porcentagens descritas na tabela 4.4 com perfis vazios.
- Util DRN2: utilidade entre documentos relevantes e não relevantes, dos 10 documentos avaliados, para as porcentagens descritas na tabela 4.4 após a primeira reformulação dos perfis, repetindo a mesma consulta.
- Util DRN3: utilidade entre documentos relevantes e não relevantes, dos 10 documentos avaliados, para as porcentagens descritas na tabela 4.4 após várias reformulações dos perfis, repetindo a mesma consulta.
- Util DRN: utilidade entre documentos relevantes e não relevantes, dos 10 documentos avaliados, para as porcentagens descritas na tabela 4.4 após várias reformulações dos perfis, utilizando uma nova consulta.

	Mesma consulta						**Nova Consulta**	
Porcento (%)	DR1	DN1	DR2	DN2	DR3	DN3	DR	DN
10	2,48	3,42	3,17	3,87	3,43	3,70	2,79	3,41
20	2,79	3,30	3,57	3,49	3,89	3,08	3,14	3,07
30	3,58	3,02	4,44	3,37	4,66	3,15	3,91	2,97
40	3,96	2,64	4,91	3,01	5,09	2,90	4,32	2,65
50	3,86	2,72	4,80	3,19	5,02	3,12	4,22	2,81
60	3,88	2,65	4,86	3,14	5,07	3,18	4,28	2,77
70	3,90	2,83	4,76	3,41	5,02	3,35	4,19	3,01
80	3,95	2,73	4,88	3,32	5,15	3,29	4,29	2,92
90	4,05	2,60	5,01	3,16	5,29	3,17	4,41	2,79
100	3,89	2,58	4,82	3,16	5,13	3,16	4,24	2,78

Tabela 4.3: Análise da relevância dos documentos pelo usuário.

Porcento (%)	Util DRN1	Util DRN2	Util DRN3	Util DRN
10	0,66	1,77	2,89	1,55
20	1,77	3,73	5,51	3,28
30	4,71	6,58	7,68	5,79
40	6,65	8,71	9,47	7,66
50	6,14	8,02	8,82	7,04
60	6,34	8,33	8,85	7,34
70	6,04	7,46	8,36	6,55
80	6,39	8,00	8,87	7,03
90	6,95	8,71	9,53	7,65
100	6,51	8,14	9,07	7,16

Tabela 4.4: Resultado de utilidade.

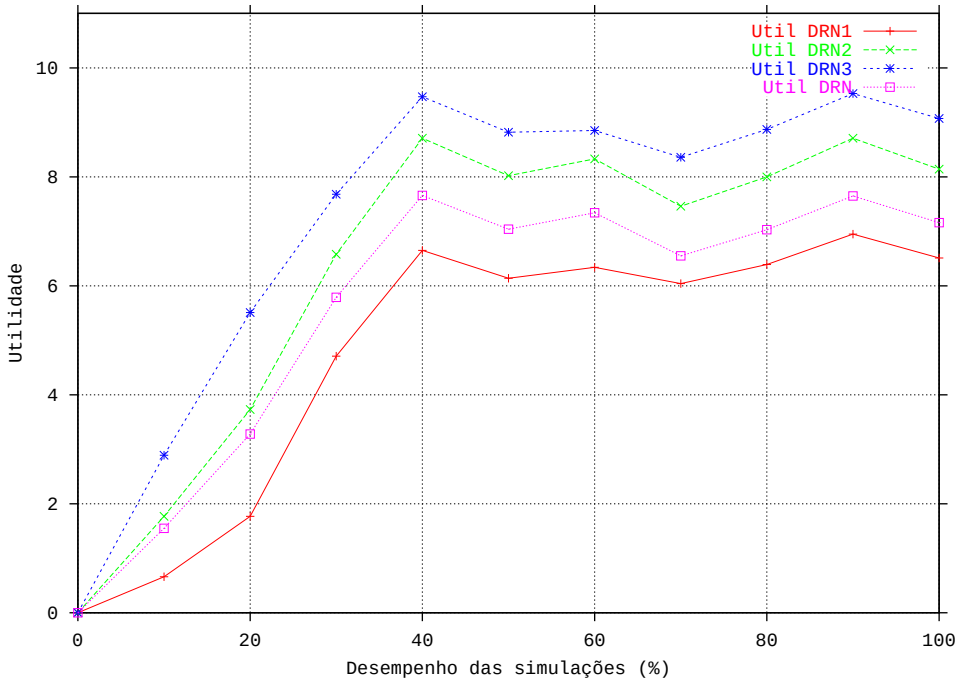


Figura 4.3: Representação da medida de utilidade.

4.6 Resultados obtidos com a coleção $FCCCMG$

Os gráficos a seguir apresentam os resultados obtidos na execução das consultas fornecidas pelos usuários, levando em consideração a reformulação ou não do perfil do usuário.

Na Figura 4.3, temos o comportamento das respostas das consultas, através das informações submetidas pelos alunos envolvidos na avaliação do sistema de filtragem, conforme dados apresentados na Tabela 4.4.

Verificou-se que uma consulta com reformulação do perfil obteve um desempenho $25,3\%$ superior, se comparado com a mesma consulta sem reformulação dos perfis conforme Tabela 4.4 colunas DRN1 e DRN2. Entretanto, numa segunda reformulação utilizando a mesma consulta, o desempenho foi inferior ao obtido após a primeira reformulação, apresentando um desempenho 11% superior conforme Tabela 4.4 colunas DRN2 e DRN3.

Todavia, verificamos através dos resultados obtidos, que sucessivas reformulações não melhoram substancialmente os resultados para a mesma consulta, apresentando ganhos desprezíveis, inferiores a 1%. Os resultados não foram inseridos na tabela por serem irrelevantes.

O desempenho de uma consulta com novos termos sobre um perfil reformulado, é 9% inferior ao obtido através da execução da mesma consulta, conforme Tabela

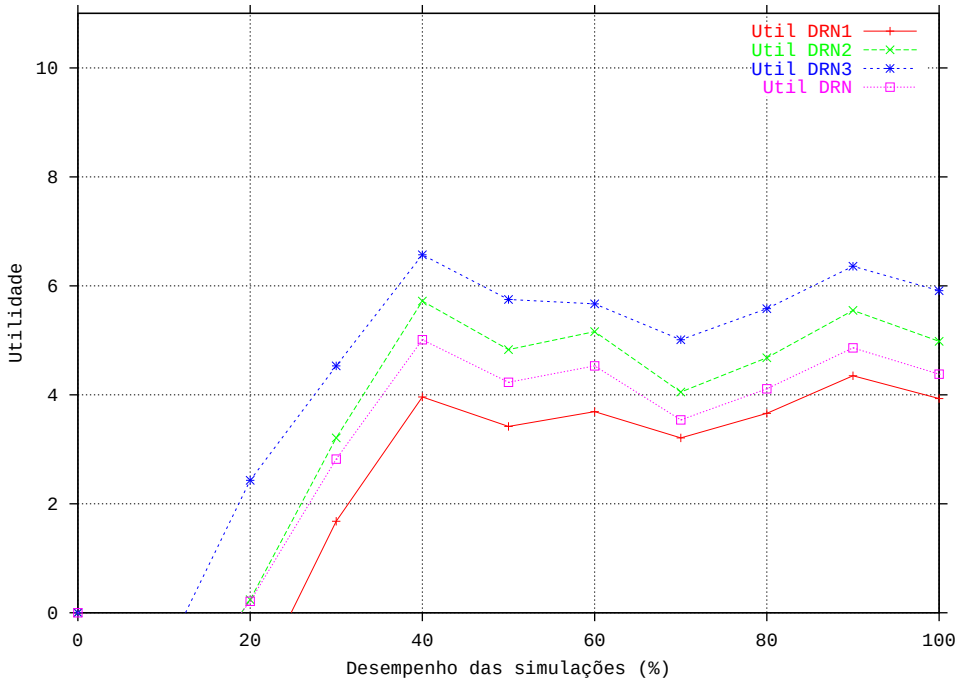


Figura 4.4: Representação da medida de utilidade pesos iguais.

4.4 colunas DRN1 e DRN.

Esses resultados são computados através da utilização da medida de utilidade, considerando que o usuário ganha 3 pontos para cada documento relevante e perde 2 pontos para cada documento não relevante recuperado, conforme Fórmula 4.1.

Na Figura 4.4 são mostrados resultados da Utilidade, considerando que o usuário ganha ou perde os mesmos valores, tanto para documentos relevantes como para documentos não relevantes. Neste caso, o peso considerado foi 3, conforme Fórmula 4.2. Entretanto, os resultados mantiveram a mesma tendência.

Na Figura 4.5 é mostrada a evolução na quantidade dos documentos relevantes para a consulta, à medida que as reformulações eram executadas. Dessa figura podemos concluir que existe uma tendência de crescimento da quantidade dos documentos relevantes recuperados com as reformulações do perfis, até a metade dos experimentos, permanecendo, esse valor estável no restante dos experimentos, com uma pequena queda de desempenho no final dos experimentos.

Na Figura 4.6 são analisados os documentos classificados como não relevantes para a consulta. Verificou-se que enquanto a quantidade de documentos relevantes aumenta à medida que os experimentos são executados, a quantiade de documentos não relevantes permanece estável.

A Figura 4.7 mostra o desempenho médio dos documentos relevantes e não rele-

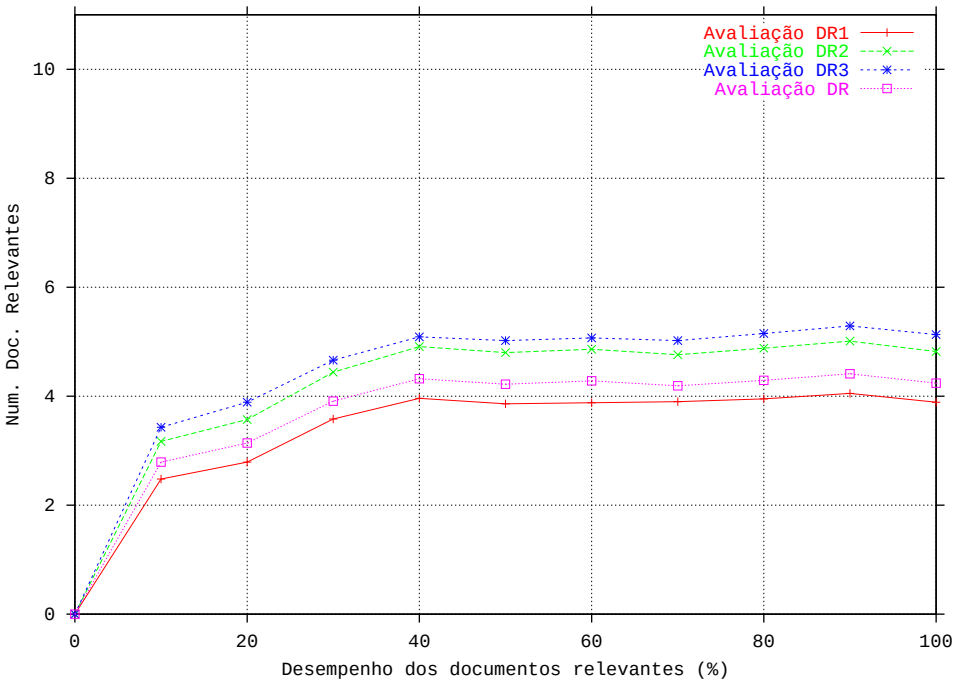


Figura 4.5: Quantidade de documentos relevantes recuperados.

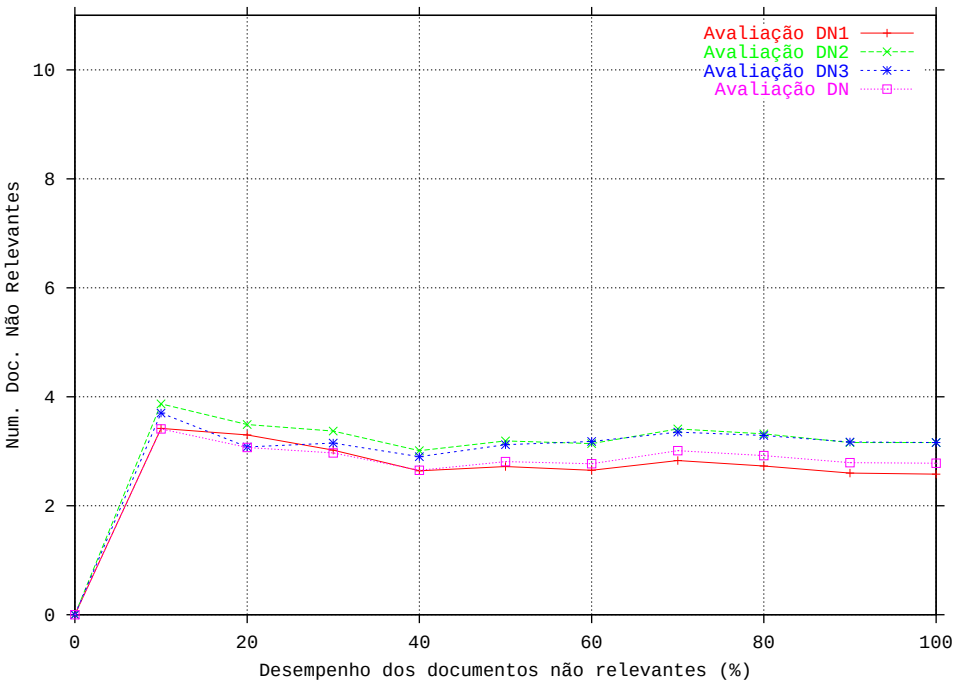


Figura 4.6: Quantidade de documentos não relevantes recuperados.

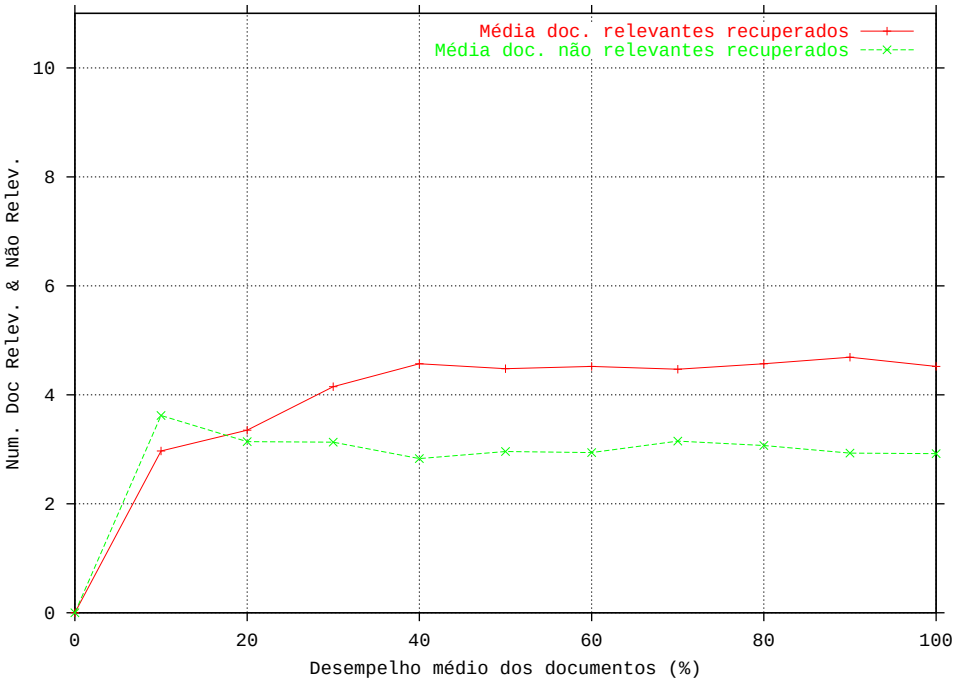


Figura 4.7: Média de documentos relevantes e não relevantes recuperados.

vantes, recuperados após sucessivas submissões de consultas e reformulações de perfis. Desse gráfico, podemos concluir que, à medida que os experimentos vão sendo executados, existe uma tendência inicial de aumento da quantidade de documentos classificados como relevantes, havendo uma inversão na quantidade de documentos não relevantes, que começa com uma taxa alta e diminui enquanto a execução dos experimentos vai acontecendo.

4.7 Resultados obtidos com a coleção *CACM*

Os experimentos com a coleção *CACM* foram conduzidos da seguinte forma: os dados da consulta foram submetidos ao sistema. O sistema então retornava os documentos relevantes, para que os resultados fossem analisados e avaliados. O primeiro documento relevante recuperado era utilizado pelo sistema para a reformulação. Essa interação acontecia por diversas vezes, incrementando-se a cada repetição o número de documentos relevantes utilizados para a reformulação. Para esta coleção foram submetidas 12 consultas e 36 reformulações dos perfis.

A Tabela 4.5 mostra resultados de experimentos que fizemos com a coleção CACM, descritos conforme rótulos abaixo:

- Caso 1: precisão média dos 11 pontos de revocação $0,288$, após a execução de

Revocação (%)	Caso 1	Caso 2	Caso 3	Caso 4
0	0,495	0,510	0,493	0,560
10	0,437	0,443	0,439	0,477
20	0,391	0,391	0,397	0,389
30	0,339	0,339	0,340	0,341
40	0,279	0,299	0,260	0,285
50	0,219	0,219	0,218	0,223
60	0,175	0,175	0,162	0,174
70	0,142	0,142	0,134	0,148
80	0,065	0,065	0,068	0,095
90	0,049	0,049	0,045	0,071
100	0,005	0,005	0,000	0,006

Tabela 4.5: Precisão média para vários níveis de revocação.

12 consultas.

- Caso 2: precisão média dos 11 pontos de revocação $0,293$, após a primeira reformulação de perfil.
- Caso 3: precisão média dos 11 pontos de revocação $0,284$, após várias reformulações, submetendo uma nova consulta.
- Caso 4: precisão média dos 11 pontos de revocação $0,311$, após várias reformulações e reutilização aleatória de uma consulta previamente submetida.

A Figura 4.8 mostra a precisão média para vários níveis de revocação dos vetores-resultado obtidos pelo sistema de filtragem, considerando uma nova consulta sem a utilização do perfil e a mesma consulta após a reformulação. Como pode ser observado, os resultados obtidos nesses dois casos são praticamente os mesmos, levando a crer que, como os documentos relevantes são julgados sem parcialidade, a aplicação da reformulação por si só não melhora o desempenho do sistema de filtragem, uma vez que os resultados obtidos pelo modelo vetorial nesse tipo de situação já é muito bom.

A Figura 4.9 mostra a precisão média, considerando o resultado obtido pelo modelo vetorial, e a submissão de uma nova consulta após a reformulação do perfil, ou a reutilização aleatória de uma consulta previamente realizada. Como pode ser observado, o desempenho de uma consulta com novos termos sobre um perfil reformulado não sofre alteração significativa. Entretanto, quando comparada a uma

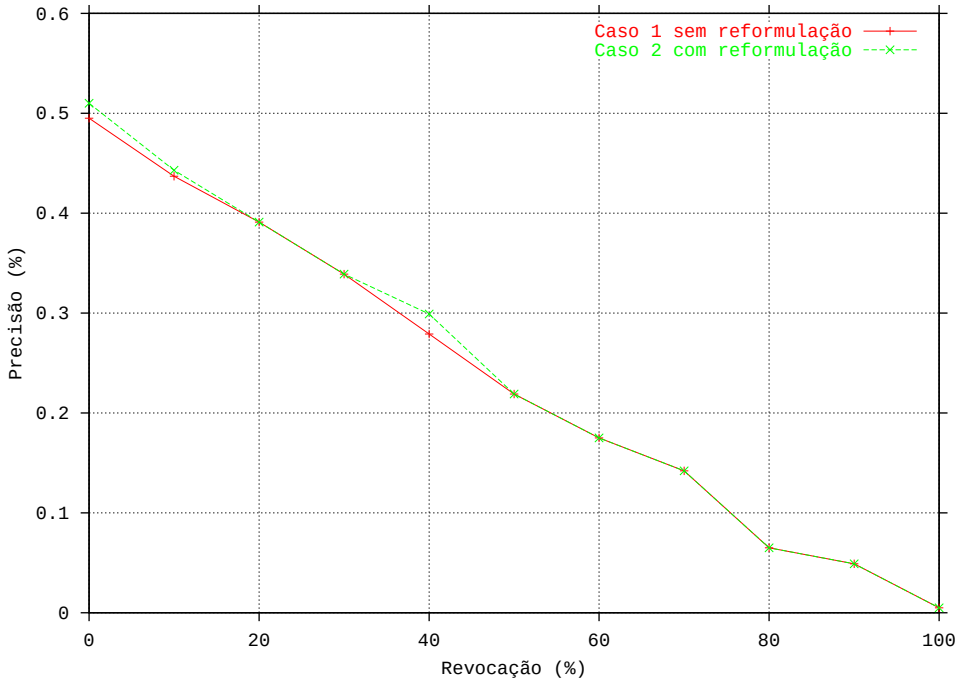


Figura 4.8: Curva de precisão-revocação.

consulta executada anteriormente, a melhora na precisão é significativa para os níveis mais baixos de revocações.

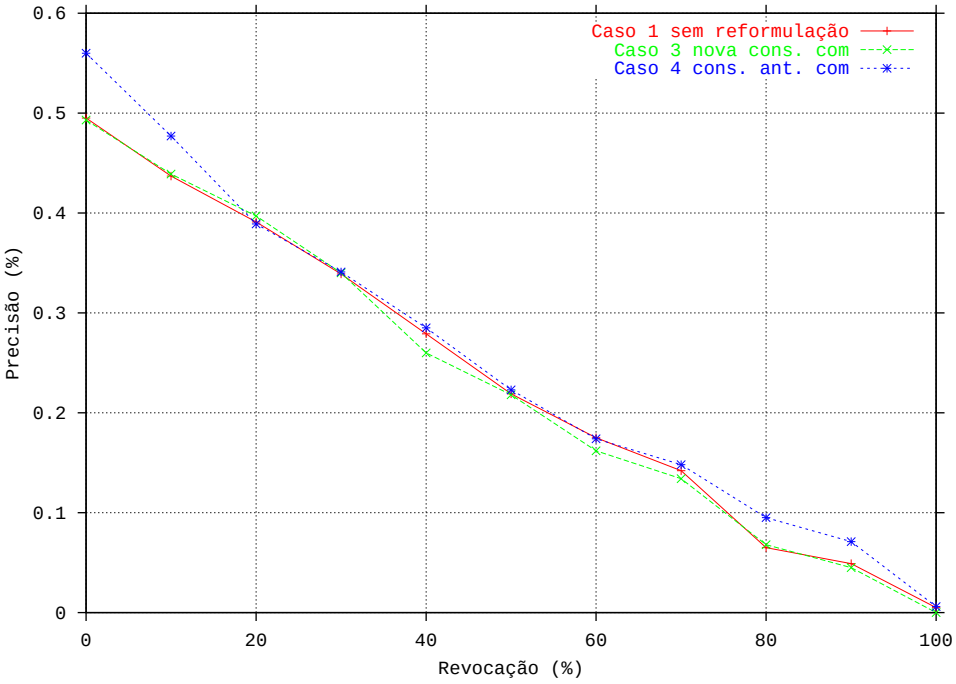


Figura 4.9: Curva de precisão-revocação.

Capítulo 5

Conclusão e Trabalhos Futuros

Nesse trabalho foi implementado um sistema de filtragem baseado no modelo vetorial com reformulação de perfil, aplicando o algoritmo de *contribuição de palavras* [15] [14]. A utilização do modelo vetorial para o processamento das consultas e a filtragem do resultado da consulta considerando o perfil do usuário justificam-se porque o modelo é eficiente e de simples implementação. Os resultados apresentados nessa dissertação demonstram que, em ambientes onde são possíveis o controle e a interação com o usuário, a utilização de perfis torna os resultados das máquinas de busca mais eficientes, aumentando o grau de satisfação dos usuários, sendo o método proposto uma alternativa viável para implementação de sistemas para recuperação de informação.

Os resultados experimentais obtidos com a simulação do sistema de filtragem mostram que a realização da reformulação do perfil melhora em média $25,3\%$ a qualidade do conjunto resposta recuperado quando comparado com a mesma consulta sem reformulação de perfil, e em 9% comparado com a submissão de uma nova consulta.

As áreas para futuras pesquisas incluem o estudo de aplicações de técnicas de *aprendizagem de máquina* [17] em filtragem de documentos, tais como *redes bayesianas* [2] e *árvores de decisão* [23], uma vez que os mesmos mostraram-se eficientes para o tratamento da incerteza.

O sistema de filtragem não foi implementado para execução em plataforma *Web*. Uma proposta de trabalho futuro seria, implementar e disponibilizar esse sistema na *Internet*. Com isso, teríamos oportunidade de testar sua eficiência em situações reais de utilização.

Um possível trabalho, seria um sistema de clipping de notícias de jornais online. Neste sistema, a tarefa principal é pesquisar em jornais as notícias relevantes a um

determinado perfil de interesse.

Outro possível trabalho, seria a utilização de *ontologias* [4][6]. A aplicação de *ontologia* na reformulação do perfil foi uma proposta considerada originalmente, que entretanto não foi investigada, devido a aplicabilidade de uma *ontologia* ser específica para uma determinada área de conhecimento.

Referências Bibliográficas

[1] A. Zheng A. Y. Ng and M.I. Jordan. Stable algorithms for link analysis. In *ACM-SIGIR*, 2001.

[2] I. Silva B. Ribeiro-Neto and R. Muntz. *Bayesian Network Models For Information Retrieval.* Springer-Verlag, 2000.

[3] J. O. Pedersen D. A. Hull and H. Schutze. Method combination for document filtering. In *In Proceedings Of The 19tht Annual International ACM SIGIR Conference On Rearch And Development In Information Retrieval*, pages 279–288, 1996.

[4] D. Fensel. Ontologies: A silver bullet for knowledge management and electronic commerce. In *Springer-Verlag*, 2001.

[5] C.S. Yang G. Salton and C.T. Yu. Term weighting in information retrieval using the term precision model. *Journal Of The ACM Journal Of The Association For Computing Machinery, Vol 24*, pages 152–170, 1982.

[6] N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. In *Communications of the ACM*, 2002.

[7] D. A. Hull. Trec-7 filtering track: Description and analysis. In *Proceedings Of The Seventh Text Retrieval Conference (TREC-7)*, 1998.

[8] D. A. Hull. Trec-8 filtering track: Description and analysis. In *Proceedings Of The Seventh Text Retrieval Conference (TREC-8)*, 1999.

[9] T. Liang I. Han, J.S. Chandler. The impact of meansurement scale and correlation structure on classification performance of inductive learning and statistical methods. In *Expert Systems With Applications, Vol 10/2*, pages 209–221, 1996.

[10] A. Moffat I. Witten and T.C. Bell. *Managing Gigabytes - Compressing And Indexing Documents And Images.* Morgan Kaufmann, 1999.

[11] M. Persin J. Zobel and R. Sacks-Davis. Filtered document retrieval with frequency-sorted indexes. *Journal Of The American Society For Information Science*, 1996.

[12] T. F. A. Jesini. *Um Sistema Para Filtragem de Notícias Disponíveis Eletronicamente*. Dissertação de Mestrato, UFMG, 2001.

[13] N. Inoue K. Hoashi, K. Matsumoto and K. Hashimoto. Experiments on the trec-8 filtering track. In *TRACK*, 1999.

[14] N. Inoue K. Hoashi, K. Matsumoto and K. Hashimoto. Query expansion method on word contribution. In *ACM SIGIR Conference on Research and Devolopment in Information Retrieval*, pages 303–304, Berckeley, CA, USA, 1999.

[15] N. Inoue K. Hoashi, K. Matsumoto and K. Hashimoto. Document filtering method using non-relevant information profile. In *ACM SIGIR Conference on Research and Devolopment in Information Retrieval*, pages 176–183, Athens, Greece, 2000.

[16] B. Krulwich and C. Burkey. The infofinder agent: Learning user interests through heuristic phrase extraction. *IEEE Expert*, pages 22–27, 1997.

[17] T. M. Mitchell. *Machine Learning*. Mc Graw-Hill, 1997.

[18] J. Mostafa and L.M. Quiroga. Filtering medical documents using automated and human classification methods. *Journal of the ASIS*, 1998.

[19] J. R. Quinlan. *Induction Of Decision Trees*. Machine Learning, 1986.

[20] J.R. Quinlan. *Programs For Machine Learning*. Morgan Kaufmann, 1993.

[21] Y. Singer R. E. Schapire and A. Singhal. Boosting and rocchio applied to text. In *Proceedings of the 21st Annual International ACM SIGIR CRDIR Conference On Rearch And Development In Information Retrieval*, pages 215–223, Melbourne, Australia, 1998.

[22] B. Ribeiro-Neto and R. Baeza-Yates. *Modern Information Retrieval*. Addison Wesley, 1998.

[23] K.H. Fasman S. Salzberg, A. L. Delcher and J. Henderson. A decision tree system for finding genes in dna. In *Journal of Computational Biology*, 1997.

[24] C. Seidman. *MySQL And mSQL.* O Reilly & Assoc, 1999.

[25] S. Suehring. *MySQL a Bíblia.* Campus, 2002.

[26] J. Mostafa W. Lam, S. Mukhopadhyay and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proceedings Of The 19th Annual International ACM SIGIR CRDIR*, 1996.

[27] R. Wilkinson and P. Hingston. Using the cosine measure in a neural network for document retrieval. In *ACM-SIGIR*, pages 202–210, 1991.

[28] T. W. Yan and H. Garcia-Molina. The sift information dissemination system. *ACM TODS. Vol 24/4*, pages 529–565, 1999.