DISSERTAÇÃO DE MESTRADO

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Recuperação de Documentos baseada em Informação Semântica no Ambiente AMMO

Orientadora: Profa. Dra. Marina Teresa Pires Vieira

Co-orientadora: Profa. Dra. Marilde Teresinha Prado Santos

Aluna: Adriana Cristina Giusti Corrêa

São Carlos

Agosto/2003

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

C824rd

Corrêa, Adriana Cristina Giusti.
Recuperação de documentos baseados em informação semântica no ambiente AMMO / Adriana Cristina Giusti Corrêa. -- São Carlos : UFSCar, 2005.
92 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2004.

1. Banco de dados. 2. Sistema de recuperação da informação. 3. Data mining (mineração de dados). 4. Ambiente AMMO. I. Título.

CDD: 005.74 (20ª)

Agradecimentos

A Deus,

Que está acima de tudo e de todos,
Que me beneficiou com a vida e a inteligência,
Que ilumina todos os passos de minha caminhada.

Aos meus pais,

Que sempre me acompanharam,
E que me deram a educação, tão importante na formação do meu caráter.

Ao meu marido Marcelo,

Muito obrigada pela paciência, pela compreensão e por me incentivar
quando eu já estava desistindo. Amo você.

Às minhas amigas e orientadoras, Profa. Marina e Profa. Marilde,

Muito obrigada por conduzirem meu trabalho para o melhor caminho,
Todo o conhecimento, que vocês me passaram, será muito importante na minha profissão.

Aos meus colegas do Grupo de Banco de Dados,

Obrigada pela troca de conhecimentos, pelas brincadeiras e acima de tudo,
Pela amizade conquistada.

A todos,
Que de alguma forma contribuíram para a realização deste trabalho.

Sumário

Índice de Figuras

Resumo

Neste trabalho são apresentadas técnicas utilizadas na extração de informação semântica de textos e estratégias para recuperação de documentos baseada em informação semântica, que foram adotadas para o desenvolvimento de um Sistema para Manipulação de Documentos baseada em Informação Semântica. Para a recuperação de documentos são considerados valores de similaridade e relevância estabelecidos. A organização da informação semântica, as interfaces com o usuário, a manipulação das informações imprecisas, o mecanismo de extração e recuperação são discutidos. A imprecisão inerente à informação semântica é tratada através de valores de similaridade para comparar os termos requeridos na consulta e os consultados no banco de dados. Os valores de relevância especificados para os termos envolvidos na consulta são utilizados para classificar os documentos resultantes. Foi desenvolvido um protótipo do sistema, que pode ser incorporado ao ambiente AMMO (***A****uthoring and* ***M****anipulation of* ***M****ultimedia* ***O****bjects*)**,** que tem por objetivo fornecer recursos para a criação, armazenamento e manipulação de aplicações multimídia.

Abstract

This study presents techniques used for extracting semantic information from texts and strategies for semantic information-based document retrieval. These techniques and strategies have been adopted so as to develop a Document Manipulation System based on Semantic Information. Previously established values of similarity and relevance are used in the document retrieval process.. The organization of semantic information, user interfaces, manipulation of imprecise information and the extraction and retrieval mechanism are discussed. The impreciseness in the semantic information is treated through similarity values in order to compare the query´spredicates with the database results. Relevance values specified for the query´s predicates have been used to classify the resulting documents. A prototype of the system has been developed. This prototype can be incorporated into the AMMO environment (*Authoring and Manipulation of Multimedia Objects*), whose aim is to provide resources for creating, storing and manipulating multimedia applications.

Capítulo 1

Introdução

1.1 Considerações Iniciais

Uma ampla variedade de dados está sendo coletada e armazenada de diversas fontes de informação. Pesquisadores, mais especificamente da Área de Banco de Dados, passaram a pesquisar novas aplicações para as informações armazenadas nos Bancos de Dados. Além das informações tradicionais armazenadas, é possível descobrir informações implícitas que são de grande importância, utilizando-se técnicas de Descoberta de Conhecimento em Banco de Dados.

A Descoberta de Conhecimento passou também a ser explorada na análise de informações textuais, onde é denominada de *Descoberta de Conhecimento em Textos* ou *Text Mining*.

Text Mining utiliza abordagens já consagradas das áreas de Recuperação de Informação, Processamento de Linguagem Natural e Descoberta de Conhecimento em Banco de Dados. A Descoberta de conhecimento em textos conta, basicamente, com duas etapas: a primeira etapa considera o tratamento do texto para convertê-lo para uma forma estruturada; a segunda etapa considera a aplicação da mineração para a descoberta do conhecimento.

Na primeira etapa, técnicas de Processamento de Linguagem Natural e Recuperação de Informação são utilizadas para o tratamento de informações textuais, enquanto que, para a segunda etapa, a Descoberta de Conhecimento em Banco de Dados é utilizada para descobrir tendências e padrões.

1.2 Objetivo da Pesquisa

Este trabalho tem por objetivo possibilitar a extração automática de informação semântica de documentos-texto, para possibilitar sua recuperação com base no conteúdo de tais documentos. A recuperação dos documentos utiliza uma estratégia baseada em similaridade entre os termos e a relevância destes para a consulta.

Para avaliar a viabilidade da técnica proposta, para a extração de informação semântica e recuperação de documentos, foi desenvolvido um protótipo de um Sistema para Manipulação de

Documentos baseada em Informação Semântica. Para alcançar tal finalidade, foi realizado um estudo sobre técnicas de Recuperação de Informação (em especial, indexação e recuperação de documentos) com o intuito de utilizar e adaptar técnicas existentes no projeto em questão. Ao final da indexação (obtenção de palavras-chave), uma tarefa de mineração (clusterização) é aplicada sobre as palavras-chave obtidas, criando classes de palavras similares, como uma forma de auxiliar na recuperação dos documentos.

O sistema pode ser acoplado a um ambiente de autoria multimídia (AMMO) que já está em desenvolvimento pelo Grupo de Banco de Dados (DC-UFSCar).

1.3 Organização do Trabalho

O trabalho está organizado da seguinte forma: O Capítulo 2 apresenta aspectos relacionados à Mineração de Informação em Textos, onde são citadas as diferenças entre mineração em bancos de dados tradicionais e bancos de dados textuais. No capítulo 3 são mostrados alguns conceitos sobre Recuperação de Informação, envolvendo indexação de textos e criação de tesauros, e ainda Extração de Informação. Ambas podem ser utilizadas para a obtenção de uma forma estruturada do documento, onde será aplicada a tarefa de mineração. O Capítulo 4 apresenta as características do ambiente AMMO, do qual este trabalho faz parte. No Capítulo 5 são descritos os recursos e técnicas empregados para o desenvolvimento do Sistema para Manipulação de Documentos baseada em Informação Semântica desenvolvido neste trabalho, sendo que os principais pontos abordados nesse capítulo são a arquitetura do sistema (módulos), estrutura de dados, a obtenção dos documentos, o pré-processamento dos documentos, o processo de criação de classes similares de termos (clusterização), a forma de armazenamento no banco de dados e as estratégias utilizadas na recuperação dos documentos.

O Capítulo 6 apresenta detalhes dos cálculos realizados para determinação da relevância de um documento para a expressão de busca, sendo utilizados para fins de classificação dos documentos. O Capítulo 7 apresenta as conclusões e sugestões de trabalhos futuros. Os principais algoritmos utilizados são apresentados no Apêndice A.

Capítulo 2

Mineração de Informação em Textos

2.1 Introdução

A informatização de diversas áreas trouxe como conseqüência um grande volume de informações sendo armazenadas em bancos de dados. Melhorias têm sido feitas em Sistemas Gerenciadores de Banco de Dados para que esses ofereçam cada vez mais recursos para a realização de consultas às informações armazenadas no Banco de Dados. Entretanto, utilizando-se apenas de consultas é impossível a descoberta do conhecimento, muitas vezes implícito nas informações armazenadas.

Teorias e ferramentas computacionais para auxiliar na extração de informação útil (conhecimento) começaram a surgir, dando origem a uma área chamada Descoberta de Conhecimento em Banco de Dados (*Knowledge Discovery in Databases - KDD*) [Fay96].

Outras ferramentas também surgiram para o tratamento de informações textuais, dando origem à área de Descoberta de Conhecimento em Textos (*Knowledge Discovery in Texts – KDT*) [Fel95].

Pesquisadores de diversas áreas, interessados em Descoberta de Conhecimento, lidam com técnicas de Mineração de Dados que, quando aplicadas a grandes volumes de informações armazenadas, revelam tendências e padrões entre os dados, os quais não são descobertos através de consultas realizadas de forma tradicional.

A descoberta do conhecimento consiste na aplicação de vários processos sobre os dados, sendo que muitas decisões são tomadas pelo usuário. De acordo com Fayyad [Fay96], uma visão prática da aplicação de KDD, enfatizando a interatividade do processo, é mostrada na figura 1.

Conforme mostra a figura, as etapas pertencentes à Descoberta de Conhecimento em Banco de Dados são:

1. Dados: A Descoberta de Conhecimento em Banco de Dados se baseia nos dados armazenados de forma estruturada;

2. Seleção dos Dados: Nesta etapa, deve-se entender o domínio da aplicação e conhecer os objetivos finais a serem atingidos para que se possa selecionar um conjunto de dados, no qual será feita a descoberta de conhecimento;

3. Processamento: Os dados são processados (ou seja, é feita a "limpeza" dos dados) visando adequá-los aos algoritmos. Isso se faz através da integração de dados heterogêneos, eliminação de dados incompletos, eliminação de tuplas repetidas, problemas de definição de tipos, etc;

4. Transformação: A etapa de transformação armazena os dados adequadamente no banco de dados, visando facilitar a aplicação dos algoritmos de mineração.

5. Mineração de Dados: A etapa de mineração começa com a escolha do algoritmo a ser aplicado (classificação, clusterização, regras associativas, sumarização, etc.). Dependendo do objetivo que se deseja alcançar com o processo de mineração, um algoritmo é aplicado nos dados obtendo-se padrões;

6. Interpretação/Avaliação: A etapa final do processo consiste em interpretar os resultados obtidos e elaborar uma apresentação de modo que o usuário possa entender.

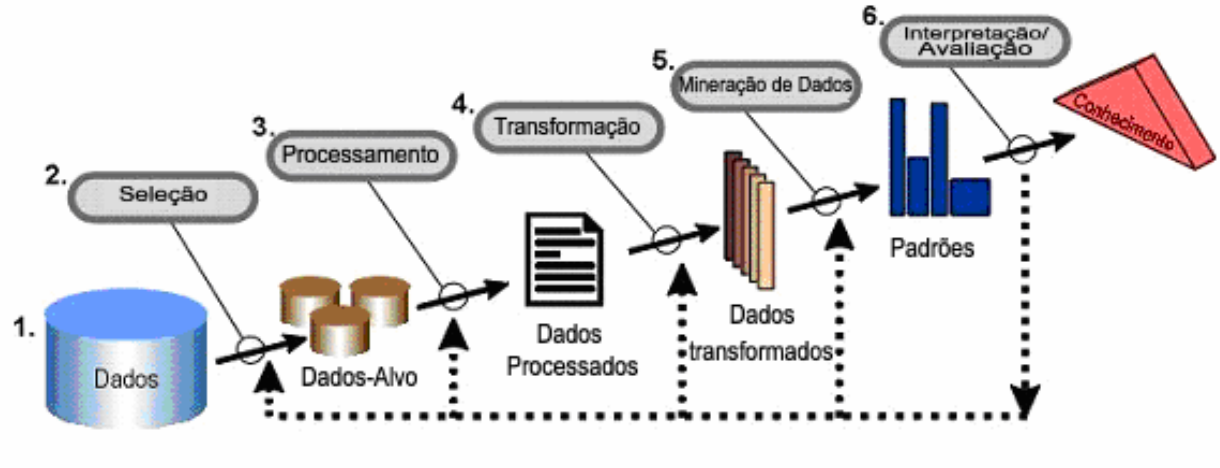


Figura 1 : Etapas do KDD

O processo é itinerante, de forma que os analistas revêem o resultado, formam um novo conjunto de questões para refinar a busca em um dado aspecto das descobertas, e realimentam o sistema com novos parâmetros.

As técnicas de Descoberta de Conhecimento também podem ser aplicadas sobre informações textuais. Entretanto, enquanto os bancos de dados tradicionais armazenam informações na forma de registros estruturados, fornecendo métodos para consultá-los, a maioria dos bancos de dados textuais apresentam-se desestruturados, impossibilitando a aplicação das mesmas técnicas de banco de dados tradicionais [Fel98].

Técnicas específicas para tratamento de textos devem ser utilizadas para possibilitar a descoberta de conhecimento implícito em bancos de dados textuais. Tais técnicas são apresentadas na próxima seção.

2.2 Descoberta de Conhecimento em Textos (KDT)

A Descoberta de Conhecimento em Textos é um campo emergente na intersecção de várias áreas de pesquisa, dentre elas mineração de dados e recuperação de informação.

Também conhecida como Mineração de Textos (*Text Mining*) [Fel95], KDT surgiu da necessidade de análises automáticas em textos, já que a grande quantidade de textos disponível, juntamente com a falta de padronização e organização destes, ocasionou uma sobrecarga de informações, dificultando sua análise manual, localização e acesso.

As técnicas de KDT visam encontrar padrões e tendências em um conjunto de documentos, realizar classificação de documentos, ou ainda comparar documentos.

Através da combinação de técnicas já existentes de Recuperação de Informação, Processamento de Linguagem Natural, Extração de Informação e Mineração de Dados, ferramentas e teorias estão em constante desenvolvimento com o objetivo de auxiliar na extração de informação útil (conhecimento) em textos.

As técnicas tradicionais para extração e busca de informações em textos não são capazes de descobrir informações úteis de coleções de documentos para o usuário, mas muitos conceitos dessas áreas foram reaproveitados para a manipulação de textos [Fel95].

A Recuperação de Informação objetiva identificar as melhores técnicas para o armazenamento e localização das informações [Lew96]. A área de Processamento de Linguagem Natural preocupa-se com a análise do texto, identificando as classes morfossintáticas existentes dentro dele. Extração de Informação visa extrair informações baseando-se em *slots* pré-definidos (itens de dados formados por pares: atributo-valor). A Mineração de Dados é o campo de pesquisas que soluciona os problemas relacionados à descoberta de informação implícita em banco de dados [Kam01].

Segundo Kamber [Kam01], a principal diferença entre Mineração de Textos e Mineração de Dados é que a primeira utiliza técnicas avançadas para explorar uma grande coleção de dados textuais desestruturados, enquanto a segunda prioriza a descoberta dentro de coleções estruturadas em bancos de dados. A mineração de textos concentra-se no fluxo sempre crescente de dados textuais, objetivando classificar documentos ou textos pertencentes a uma categoria, encontrar relações ou associações entre documentos, agrupar textos ou determinar quem escreveu um determinado texto.

Por se tratar de documentos textuais desestruturados, há a necessidade de transformá-los em uma forma estruturada para que seja possível a aplicação de técnicas de mineração. A figura 2 ilustra o processo para obtenção de conhecimento a partir de textos:

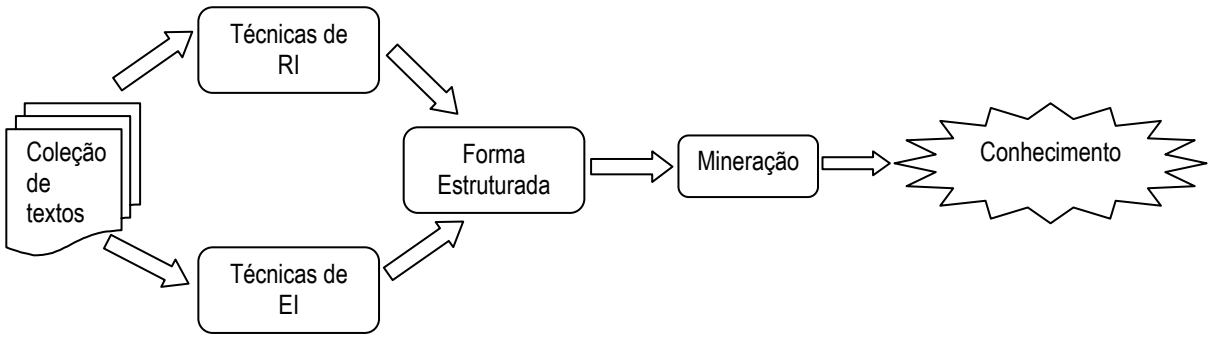


Figura 2: Processo de Mineração de Textos

Como pode ser observado, a partir da Coleção de Textos, uma Forma Estruturada é obtida, e para isso, Técnicas de Recuperação de Informação ou Técnicas de Extração de Informação podem ser utilizadas; a aplicação de técnicas de mineração para a obtenção do Conhecimento é feita sobre a Forma Estruturada.

O processo de mineração envolve desde a obtenção dos documentos relevantes para a análise até a interpretação dos resultados produzidos. O processo de mineração de textos envolve etapas, as quais são descritas detalhadamente na seção seguinte.

2.3 Etapas da Mineração de Textos

A descoberta de conhecimento em textos envolve passos para obtenção de informação útil. Os dados são extraídos dos textos e formatados em bases de dados estruturadas com o auxílio de técnicas de Recuperação de Informação (RI) ou Extração de Informação (EI), onde serão aplicadas técnicas de Mineração de Dados. De acordo com Dixon [Dix97], as etapas que podem ser seguidas para a mineração de textos são:

a) Recuperação de Informação: fase onde são feitas a localização e recuperação de documentos que podem ser considerados relevantes. É necessário um sistema que filtre o conjunto de documentos especificado pelo usuário e indexe as palavras-chave encontradas, as quais identificam o conteúdo dos textos.

b) Extração de Informação: fase de preenchimento de *templates (*documentos com lacunas a serem preenchidas) com a informação desejada da coleção de documentos. Os itens identificados (características, palavras) que são relevantes nos documentos, são extraídos e convertidos em dados (tabelas ou *templates*) que possam ser utilizados pelas técnicas de mineração.

c) Mineração de Informação: estando a informação armazenada de forma estruturada (*templates* ou palavras-chave), a descoberta de padrões nos dados é feita sobre o banco de dados gerado, através de aplicação de uma tarefa de mineração.

d) Interpretação: interpretação dos padrões recuperados na fase de mineração. Os resultados obtidos são interpretados realizando a descoberta do conhecimento.

O processo de descoberta de conhecimento em textos, de um modo geral, necessita de uma forma estruturada para representação dos textos, obtida manualmente ou automaticamente por ferramentas, para que possa utilizar as técnicas de KDD.

Segundo Feldman [Fel98], os documentos devem ser pré-processados, permitindo-se a extração de elementos lingüísticos mais complexos, também chamados de termos (palavras-chave). Isso é feito através do processo de recuperação de informação, o que permite localizar documentos a partir da comparação direta do termo de consulta do usuário e os termos presentes nos documentos [Wiv97]. Nesse sentido, as técnicas de Recuperação de Informação podem ser utilizadas para o processo de indexação, extraindo as palavras-chave que identificam o conteúdo dos textos. A indexação por palavra-chave também é discutida em [Raj98], onde é utilizada associação por palavra-chave.

Uma analogia entre as tecnologias para tratamento de informações textuais é feita por Nasukawa [Nas01], onde é possível observar características como a função desempenhada, tecnologia utilizada e a representação dos dados obtidos.

Segundo ele, a Recuperação de Informação é uma tecnologia utilizada para buscar documentos, focalizando nos dados relacionados a algum tópico específico. Os dados são representados por palavras-chave e a saída de um Sistema de Recuperação de Informação é um conjunto de documentos. Em termos de extração de informação, muitos tipos de informação podem ser extraídos de dados textuais, tais como informação semântica e léxica de domínio específico que pode estar armazenada em banco de dados. Por outro lado, a Extração de Informação (EI) inicialmente era voltada apenas para Conferências de Entendimento de Mensagens (MUCs) visando encontrar classes específicas de eventos, tais como consolidação de empresas e preenchimento de *templates* para cada instância de um evento. Atualmente, a EI é bastante utilizada para descobrir tendências no domínio descrito no banco de dados textual.

A Mineração de Informação pode ser utilizada para classificar documentos dentro de uma categoria já existente ou agrupar documentos que possuam os mesmos conceitos. A Mineração ainda pode ser utilizada com o propósito de extrair relacionamentos entre os documentos. Nesse último caso, a saída de um Sistema de Mineração é um conjunto de informações que indicam

padrões, regras associativas, etc. A próxima seção apresenta algumas tarefas de mineração de informação em textos.

2.4 Tarefas de Mineração em Textos

As fases de Recuperação de Informação ou Extração de Informação objetivam o refinamento do texto, transformando textos desestruturados em uma forma estruturada. Nessa forma estruturada, os dados extraídos são armazenados em banco de dados, para que as técnicas de *Data Mining* (DM) sejam utilizadas, visando a descoberta de conhecimento. A obtenção da forma estruturada é apresentada no Capítulo 3.

Fayyad [Fay96] define *Data Mining* como sendo a aplicação de algoritmos específicos sobre o conjunto de dados escolhidos, sob algumas limitações aceitáveis de eficiência computacional, produzindo certa quantidade de padrões.

Os algoritmos utilizados nas tarefas de *Data Mining* são baseados em conceitos que já eram conhecidos como aprendizagem de máquina e estatística. As três áreas estão inter-relacionadas: Aprendizagem de máquina, Estatística e *Data Mining*.

Segundo Hand [Han99], *Data Mining* tem o objetivo de descobrir estruturas nos dados. *Data Mining* envolve análises "retrospectivas" dos dados e seus usuários estão geralmente mais interessados na compreensibilidade do que na precisão e previsibilidade [Gly97].

Aprendizagem de máquina (*Machine Learning*) tenta fazer com que os programas de computador "aprendam" com os dados que eles estudam, tal que esses programas tomem decisões diferentes baseadas nas características dos dados estudados. Utiliza conceitos fundamentais da estatística e heurística avançada de inteligência artificial para alcançar seus objetivos.

Aprendizagem de máquina é um subcampo de Inteligência Artificial, pois lida com formulação e modificação de teorias como resultado de observações. O aprendizado de máquina incorpora também muitas idéias da área de estatística, onde um sistema de indução em árvore de decisão é bastante útil e foi desenvolvido por estatísticos [Qui93].

Data Mining tem como objetivos primários: a predição e a descrição. A predição envolve o uso de algumas variáveis ou campos no banco de dados para predizer os valores futuros ou desconhecidos de outras variáveis de interesse. A descrição enfoca a busca de padrões interpretáveis que descrevem os dados [Fay96b].

Segundo Tan [Tan99], a descoberta de conhecimento em textos deduz padrões e relacionamentos (por exemplo, clusterização e categorização) ou deriva padrões entre os conceitos em um domínio (por exemplo, associações).

As seções seguintes descrevem características das tarefas de sumarização, clusterização, classificação, associação, dentre outras.

2.4.1 Sumarização

A sumarização envolve métodos para encontrar uma descrição compacta para um subconjunto de dados. É bastante utilizada na descoberta de conhecimento em textos, visando identificar palavras ou frases mais importantes do documento ou conjunto de documentos que sumarizam o conceito dos documentos. É útil para reduzir a quantidade de material em um documento, embora mantenha a mesma informação.

A sumarização procura produzir uma lista de sentenças que estão presentes dentro do documento origem para resumir o conteúdo do documento [Dix97]. Alguns exemplos da utilização de sumarização em textos podem ser encontrados em [Net00] [Meg00] [Cop01].

Um exemplo de sumarização de textos pode ser visto abaixo, onde as frases que pertencem ao sumário estão sublinhadas.

Frequent item sets form the basis of association rule mining. Exploiting the monotonicity property of frequent item sets (each subset of a frequent item set is also frequent) and using data structures supporting the support counting, the set of all frequent item sets can be efficiently determined even for large databases. Many different algorithms have been developed for that task, including Apriori [5]. See [8] for an overview on association rule mining. Frequent item sets can also be used for the task of classification. [9] introduces a general method of building an effective classifier from the frequent item sets of a database. [10] presents a modification of this approach for the purpose of text classification. A frequent item-based approach of clustering is promising because it provides a natural way of reducing the large dimensionality of the document vector space. Since we are dealing not with transactions but with documents, we will use the notion of *term* sets instead of *item* sets. A term is any preprocessed word within a document, and a document can be considered as a set of terms occurring in that document at least once. The key idea is not to cluster the high-dimensional vector space, but to consider only the low-dimensional frequent term sets as cluster candidates. A well-selected subset of the set of all frequent term sets can be considered as a clustering. Strictly speaking, a frequent term set is not a cluster (candidate) but only the description of a cluster (candidate). The corresponding cluster itself consists of the set of documents containing all terms of the frequent term set. Unlike in the case of classification, there are no class labels to guide the selection of such a subset from the set of all frequent term sets. Instead, we propose to use the mutual overlap of the frequent term sets with respect to their sets of supporting documents (the clusters) to determine a clustering. The rationale behind this approach is that a small overlap of the clusters will result in a small classification error, when the clustering is later used for classifying new documents. In this section, we introduce the necessary definitions and present two algorithms for frequent term-based text clustering.

Figura 3: Frases sublinhadas representando o sumário

2.4.2 Regras de Associação

As regras de associação, quando aplicadas a um conjunto de dados, possibilitam encontrar regras do tipo X=>Y, ou seja, transações do banco de dados que contêm X tendem a conter Y. Parâmetros de suporte e confiança devem ser fornecidos para que as regras que satisfaçam esses parâmetros sejam encontradas.

As regras de associação são bastante utilizadas em mineração de textos, visando descobrir associações entre termos e categorias de documentos. Nesse sentido, destaca-se a proposta de Zaiane [Zai01] para elaboração de um método rápido para construção de um classificador de texto usando mineração de regras de associação. O classificador é baseado em regras de associação extraídas do conjunto de treinamento. Após pré-processar o documento texto, eliminando *stop-words*, os documentos são representados por conjuntos de palavras limpas e a categoria a que pertencem. Trabalhos envolvendo regras de associação podem ser encontrados em [Lan98] [Fel96] [Fel98a] [Nah00].

Para realizar a tarefa de Associação, algoritmos específicos podem ser utilizados individualmente ou em combinação. Na próxima seção, o algoritmo APRIORI é apresentado visto que é o mais conhecido e de fácil utilização na tarefa de encontrar regras associativas.

2.4.2.1 Algoritmo APriori

O algoritmo APriori é utilizado para encontrar associações relevantes entre itens de dados.

São definidos parâmetros que determinam quais associações são interessantes ou não para o usuário. Para eliminar regras que não aparecem freqüentemente no banco de dados é usado o suporte, que indica a porcentagem de registros onde aparecem juntos X e Y, sobre o total de registros. Outra medida, que indica o grau de acerto da regra, é a confiança, a qual indica a porcentagem de registros que contém X e Y, sobre o total de registros que possuem X.

O algoritmo visa encontrar todas as regras de associação que satisfaçam as restrições de valores de suporte e confiança definidos pelo usuário [Agr93] [Agr94].

Quando aplicado a textos, o algoritmo APriori encontra conjuntos freqüentes de palavras nos documentos do conjunto de treinamento. As regras interessantes são do tipo X=>Y, onde X é um conjunto de palavras e Y é uma categoria. Pode-se considerar cada categoria separadamente (regras diferentes são extraídas dos documentos de cada categoria) e, portanto, o classificador obtido é um conjunto de regras de cada categoria; ou considerar documentos de todas as

categorias, obtendo-se assim um classificador para todo o conjunto de dados, levando em conta termos que ocorrem com freqüência em diferentes categorias.

2.4.3 Classificação / Categorização

Classificação ou Categorização é a classificação automática de documentos baseando-se em um conjunto treinado pré-classificado. O conjunto de treinamento deve ser definido por pessoas experientes e o algoritmo então, é aplicado para classificar documentos novos em uma classe.

Um Treinador, algoritmo capaz de extrair conhecimento, analisa todos os exemplos de documentos, aprende as regras e as armazena em uma Base de Conhecimento. Os documentos a serem classificados passam por um Categorizador, o qual, baseado nas regras previamente inseridas na Base de Conhecimento, estabelece a classe à qual pertence cada documento. Maiores detalhes sobre o processo de classificação podem ser encontrados em [Sal83].

A figura 4 mostra o processo de classificação:

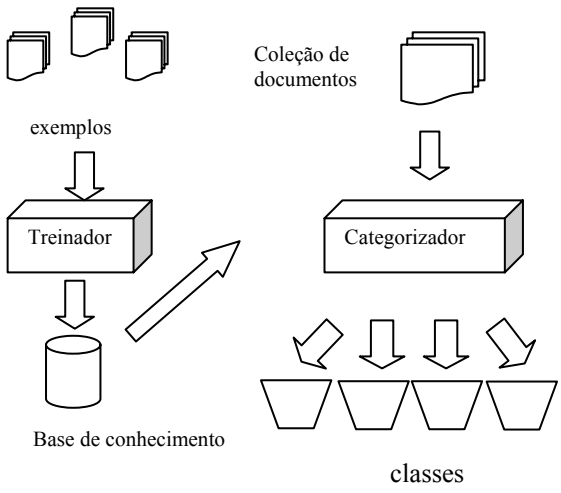


Figura 4: Processo de Classificação

2.4.4 Clusterização

A clusterização consiste em agrupar documentos similares, baseado nos termos usados dentro dos documentos. O agrupamento também pode ser feito em nível de termos, onde termos similares são colocados em uma mesma classe. O processo de clusterização não requer um conjunto treinado, como o utilizado em Classificação.

A figura 5 mostra o processo de clusterização:

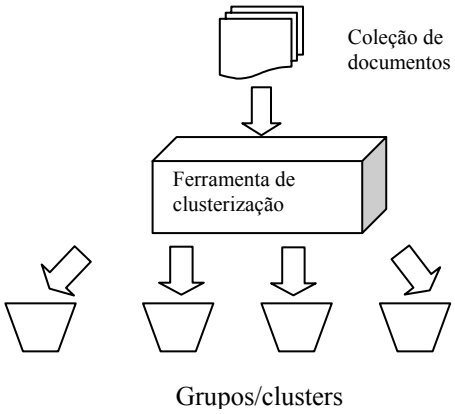


Figura 5: Processo de Clusterização

A clusterização é usada para particionar um conjunto de dados em *clusters* ou grupos, onde os objetos no mesmo "*cluster"* são mais similares se comparados aos objetos pertencentes a outros "*clusters*". No campo de recuperação de informação, a clusterização de documentos é usada para automaticamente organizar grande coleção de resultados recuperados, agrupando os documentos que pertencem ao mesmo tópico para facilitar a navegação pelos documentos resultantes [Hea96].

A maioria dos algoritmos de clusterização usa o modelo de espaço vetorial de Recuperação de Informação [Sal83], no qual os documentos-texto são representados como um conjunto de pontos em um espaço vetorial. A similaridade entre dois textos é tradicionalmente medida pelo cosseno do ângulo entre os vetores. Os documentos considerados similares, com base nessa medida, são agrupados pelo algoritmo de clusterização.

A construção de classes de termos similares (equivalente ao Tesauro) também é obtida a partir de uma matriz de similaridade, onde são coletados em uma classe comum todos os termos cuja similaridade ultrapassa um limite estabelecido.

Trabalhos envolvendo clusterização podem ser encontrados em [Kar00] [Net00].

No Capítulo 3 são fornecidos detalhes sobre o processo de clusterização de documentos, por ser utilizado neste trabalho.

2.5 Ferramentas

As ferramentas para descoberta de conhecimento em textos são voltadas para visualização de documentos e grupos de documentos similares (utilizando clusterização) ou são ferramentas voltadas para o entendimento e análise do texto (utilizando categorização de texto, extração de informação e sumarização).

As ferramentas descritas nas seções seguintes utilizam técnicas de Recuperação de Informação ou técnicas de Extração de Informação para obtenção da forma estruturada. Tais técnicas são apresentadas no Capítulo 3.

2.5.1 Text Analyst (TA)

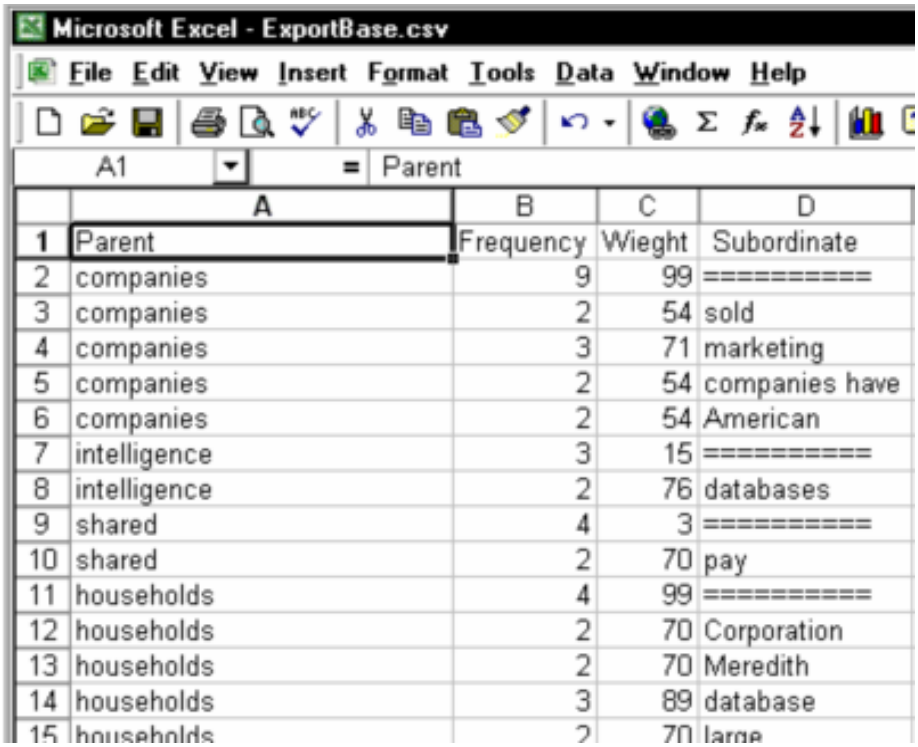
A ferramenta Text Analyst é distribuída por Megaputer Intelligence [Meg00]. A entrada para o Text Analyst é um texto, do qual as informações relevantes são extraídas. As opções iniciais do TA estão exibidas na figura 6, onde o usuário pode escolher entre analisar novos textos e criar a base de conhecimento, abrir uma base de conhecimento existente ou ainda utilizar o tutorial.

Se a escolha for "Analisar novos textos", a ferramenta determinará conceitos, ou seja, palavras e combinações de palavras que são mais importantes no contexto. Algoritmos matemáticos dentro do TA determinam a importância relativa de um conceito, analisando suas conexões a outros conceitos no texto.

O TA cria uma Rede Semântica sem utilizar um conhecimento prévio do assunto. Entretanto, o tema pode ser adicionado pelo usuário através de um dicionário externo, quando desejado.

Figura 6: Caixa de Diálogo Inicial do Text Analyst

Para cada conceito são obtidos dois valores que representam os pesos semânticos dos conceitos em relação ao conceito-pai e os pesos semânticos dos conceitos em relação ao documento. O TA também cria sumários dos documentos de entrada baseando-se nos pesos semânticos. TA pode exportar os resultados para um arquivo no formato Web (HTML), ou ainda exportar para um formato compatível com a Planilha Eletrônica Excel. A figura 7 ilustra o resultado exportado para o aplicativo Microsoft Excel:



Microsoft Excel - ExportBase.csv

File Edit View Insert Format Tools Data Window Help

A1 = Parent

	A	B	C	D
1	Parent	Frequency	Wieght	Subordinate
2	companies	9	99	==========
3	companies	2	54	sold
4	companies	3	71	marketing
5	companies	2	54	companies have
6	companies	2	54	American
7	intelligence	3	15	==========
8	intelligence	2	76	databases
9	shared	4	3	==========
10	shared	2	70	pay
11	households	4	99	==========
12	households	2	70	Corporation
13	households	2	70	Meredith
14	households	3	89	database
15	households	2	70	large

Figura 7: Resultados exportados para o Excel

Como pode ser observado na figura 7, cada conceito é colocado em uma linha do Excel, juntamente com sua freqüência e peso semântico em relação ao conceito-filho (subordinado).

2.5.2 TextVis [Lan98]

O TextVis é um sistema visual de *Data Mining* para coleções de documentos. Tal coleção representa um domínio de aplicação e o objetivo primário é derivar padrões que forneçam conhecimento sobre esse domínio. TextVis possui uma abordagem multi-estratégica para *Text Mining* e possibilita esquemas de análise complexa dos componentes. O sistema fornece grande coleção de ferramentas de análise, como conjuntos freqüentes, associações, distribuições de conceitos e correlações de conceitos. Os padrões descobertos são apresentados em uma interface visual permitindo que o usuário interfira nos resultados para acessar os documentos associados.

2.5.3 FACT [Fel96]

Trata-se de uma ferramenta para descobrir associações em coleções de documentos textuais, dado um conhecimento prévio sobre os tópicos dos documentos na coleção.

As várias tarefas de descoberta são especificadas em uma interface gráfica, fornecendo semânticas bem definidas para as ações de descoberta executadas pelo usuário.

Uma das entradas do sistema é o documento, que deve estar rotulado com palavras-chave representando os tópicos do documento.

Outra entrada é o conhecimento prévio para o processo de descoberta, o qual deve definir predicados sobre as palavras-chave rotulando os documentos e representando os relacionamentos entre elas. Esse conhecimento prévio pode necessitar passar por outras ferramentas para converter no formato específico para o FACT.

A terceira entrada é a escolha feita pelo usuário da tarefa de descoberta de conhecimento. O resultado do processo de descoberta é passado para uma ferramenta que apresenta os resultados e permite que o usuário navegue por eles.

2.5.4 Document Explorer [Fel98a]

A ferramenta Document Explorer primeiramente faz a conversão de documentos para o formato SGML; os documentos resultantes fornecem informações lingüísticas sobre o conteúdo de cada documento. A seguir os documentos são rotulados com os termos extraídos baseados em análise sintática e padrões de ocorrência na coleção de documentos. Os termos são colocados em uma taxonomia através da interação com o usuário e finalmente as operações KDD são executadas nos documentos rotulados. Essa taxonomia captura relacionamentos entre grupos de termos ao invés de termos individuais, além de permitir que o usuário especifique as tarefas de mineração.

Uma abordagem em nível de termos, estudada por Feldman [Fel98a], faz a descoberta do conhecimento a partir de palavras e frases que são extraídas de documentos rotulados. Em trabalhos anteriores do autor foi assumido que cada documento possuía um conjunto de rótulos e as operações de descoberta de conhecimento eram feitas sobre esses rótulos e, portanto, era necessário rotular manualmente os documentos ou utilizar técnicas automáticas para isso, onde regras eram aprendidas de documentos já rotulados.

Outra abordagem é a extração de termos em cada documento para encontrar seqüências de palavras que parecem ter o mesmo significado e então executar a mineração nos termos extraídos, rotulando cada documento. Os termos representam conceitos significativos no domínio do documento. O método de extração elimina a rotulação de documentos.

2.5.5 TextMiner [Kar00]

Extrai eventos e termos (seqüências significativas de palavras) em cada documento para encontrar características com significado no domínio, e então aplica mineração nas características, rotulando cada documento.

Consiste de dois componentes: Text Analysis e Data Mining. O primeiro módulo converte dados semi-estruturados, tais como documentos-texto em dados estruturados armazenados em banco de dados. O segundo módulo aplica técnicas de Data Mining na saída do primeiro módulo. É utilizado um algoritmo de clusterização e depois técnicas de classificação para validar os resultados.

Primeiramente, é feito um pré-processamento lingüístico; os termos dentro do domínio e os eventos de interesse são extraídos, os quais preencherão uma tabela. Essa tabela será usada como entrada para o algoritmo de clusterização. Após a clusterização são usados algoritmos de árvores de decisão para recuperar hierarquias de conceitos e testar a validade das descrições descobertas.

2.5.6 DiscoTEX [Nah00]

O sistema DiscoTEX combina métodos de Extração de Informação (EI) e KDD para executar tarefas de Text Mining, descobrindo regras de predição a partir de texto em linguagem natural.

O módulo de EI localiza partes específicas de dados no texto, gerando um banco de dados, o qual é fornecido para o módulo de KDD para a mineração. A figura 8 exibe o processo de mineração de texto. A entrada para a ferramenta são textos onde o módulo de extração de informação os processa retirando as informações desejadas; os documentos irrelevantes são eliminados; para isso é utilizado um categorizador treinado. A informação a ser extraída está contida em slots. A informação é então armazenada no banco de dados, onde o módulo de KDD executa as tarefas de mineração e gera, como saída, as regras descobertas.

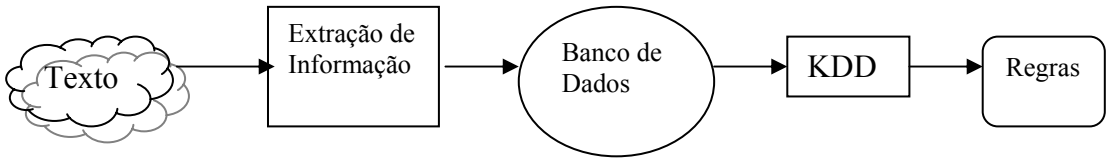


Figura 8: Mineração de Textos baseada em Extração de Informação

2.5.7 Copernic Summarizer [Cop01]

O Copernic Summarizer é um programa que cria sumários de páginas Web e documentos-texto em várias linguagens, localizando elementos tais como conceitos-chave e sentenças.

A tecnologia de inteligência artificial integrada ao produto permite "entender" o conteúdo do documento e extrair conceitos e sentenças-chave. A ferramenta incorpora dois componentes: modelos estatísticos e processos de conhecimento intensivo.

O modelo estatístico pode ser aplicado para vários idiomas para aproximar do vocabulário específico. Ele inclui estimativas Bayesianas e sistemas de regras derivadas da análise de milhares de documentos.

O processo de conhecimento intensivo leva em conta o modo como os humanos sumarizam textos. A efetividade do processo em gerar sumários de qualidade é dependente de como as tecnologias de sumarização integram e aplicam conhecimento, combinado com uma variedade de informações de especialistas (não necessariamente da área de computação).

As tecnologias de sumarização do Copernic empregam algoritmos lingüísticos que são específicos para processar a linguagem escrita. Esse processo envolve dados estatísticos sobre o uso da palavra através de milhares de documentos.

Todas as sentenças são medidas de acordo com a importância relativa das informações que elas contêm.

As características principais da ferramenta "Copernic Summarizer" são:

- produção de relatórios de sumário para conceitos de texto pelo processamento de documentos;

- sumarização de documentos em inglês, francês, alemão e espanhol;

- exportação dos sumários em vários formatos (HTML, XML, Rich Text e arquivo texto).

2.6 Considerações Finais

Este capítulo apresentou as técnicas utilizadas para Mineração de Informação em Textos. A escolha da técnica a ser utilizada depende do objetivo da aplicação. Dentre as técnicas para descoberta de conhecimento em textos destacam-se a sumarização, onde as frases, que identificam o conteúdo do documento, são extraídas para formar o sumário; a clusterização, que agrupa em um grupo (*cluster)* documentos similares; classificação, que utiliza um treinador para

aprender as regras para dividir os documentos em classes pré-definidas; e a associação, para a descoberta de associatividade entre termos que ocorrem nos documentos.

As ferramentas apresentadas não foram utilizadas neste trabalho, pois elas processam o texto e apresentam somente o resultado final, por exemplo, um texto sumarizado, regras de associação, etc. Neste trabalho, é necessário saber a freqüência do termo dentro de um texto e dentro da coleção de textos, relacionamento entre termos de um texto, para que essas informações sejam armazenadas no banco de dados, possibilitando a utilização do mecanismo de busca desenvolvido.

Para o tratamento dos textos, há a necessidade da utilização de algumas técnicas já existentes das áreas de Recuperação de Informação e Extração de Informação a fim de obter uma forma estruturada, para que se possam aplicar as técnicas de mineração.

O capítulo seguinte apresenta os recursos que podem ser utilizados para obtenção da forma estruturada de um texto.

Capítulo 3

Estruturação do Texto para a Mineração

3.1 Introdução

A aplicação de mineração de informação em dados textuais exige que as informações estejam armazenadas de forma estruturada. Os documentos-texto devem ser pré-processados para que se obtenha uma forma estruturada, sobre a qual as tarefas de mineração podem ser aplicadas.

Abordagens das áreas de Recuperação de Informação e Extração de Informação podem ser utilizadas como pré-processamento de documentos, os quais são capazes de obter tal forma estruturada.

Sistemas de Recuperação de Informação visam obter palavras-chave dos documentos, as quais representam o conteúdo de tais documentos, enquanto que os Sistemas de Extração de Informação visam extrair informações baseando-se em *slots* pré-definidos (itens de dados formados por pares: atributo-valor).

As seções seguintes descrevem as técnicas das áreas de Recuperação de Informação e Extração de Informação, para a obtenção da forma estruturada no processo de Mineração de Textos.

3.2 Recuperação de Informação

Diferentes sistemas utilizam técnicas de Recuperação de Informação, tais como Sistemas de Gerenciamento de Informação, Sistemas de Gerenciamento de Banco de Dados, Sistemas para Tomada de Decisão e Sistemas de Recuperação de Informação. Nesse capítulo são analisados os Sistemas de Recuperação de Informação (SRI's) que fazem parte da Descoberta de Conhecimento em Textos.

Os Sistemas de Recuperação de Informação (SRI's) surgiram da necessidade de se extrair informações em bases de dados não estruturadas, tais como grandes coleções de documentos textuais e bibliográficos.

Segundo Salton [Sal83], os SRI's necessitam de técnicas que agilizam o armazenamento e acesso aos dados. Tais técnicas envolvem a atribuição de termos apropriados e identificadores para representar o conteúdo dos documentos na coleção. Esta tarefa, conhecida como indexação, pode ser feita manualmente ou automaticamente.

A recuperação de informação é feita a partir de uma entrada do usuário, ou seja, uma consulta para que os documentos relevantes sejam encontrados. Os SRI's geralmente se baseiam em Busca por Palavra-Chave ou Busca por Similaridade [Kam01].

Na Busca por Palavra-Chave, um documento é representado como um conjunto de termos e pode ser identificado por palavras-chave. Dada uma consulta do usuário que contém a palavra-chave desejada, o sistema recupera os documentos que apresentam essa palavra-chave.

A Busca por Similaridade pode ser representada pelo modelo vetorial, criado por Salton [Sal83]. O modelo vetorial representa documentos e consultas como vetores de termos. O vetor resultado para uma consulta é montado através de um cálculo de similaridade entre o vetor de documentos e o vetor que representa a consulta. Pesos relacionados a cada termo do vetor quantificam a relevância dos termos para as consultas e para os documentos. As estratégias referentes ao processo de busca são apresentadas na seção 3.2.3.

O processo de indexação de textos, assim como os cálculos utilizados para a indicação de relevância dos termos para os textos, são apresentados nas seções seguintes.

3.2.1 Processo de Indexação

Segundo Salton [Sal83], a indexação é o processo pelo qual as palavras contidas nos textos são armazenadas em uma estrutura de índice para viabilizar a pesquisa de documentos através das palavras que eles contêm.

Índices invertidos são criados para possibilitar melhoras significativas no desempenho e na funcionalidade da busca. As buscas usam os índices extraídos dos documentos-texto para comparações com a consulta do usuário. O arquivo de índices invertidos possui como identificadores as palavras-chave do texto e para cada palavra-chave, há um conjunto de referências que apontam para os documentos onde essas palavras-chave ocorrem. O processo de obtenção de palavras-chave que identificam o conteúdo do documento é chamado de Indexação Automática [Sal68].

A estrutura do arquivo invertido é composta por dois elementos: o vocabulário e as ocorrências. O vocabulário é o conjunto de palavras retiradas do texto. As ocorrências são os documentos onde essas palavras aparecem [Bae99].

A figura 9 mostra a utilização de arquivos invertidos para o armazenamento dos termos que identificam os documentos. Os termos, ou palavras-chave são extraídos dos textos e ficam armazenados juntamente com as referências para os respectivos documentos. Através dessas referências é possível recuperar os documentos desejados.

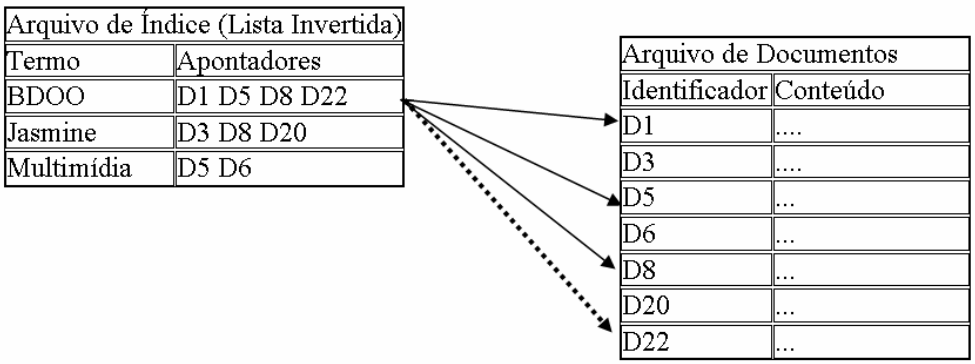


Figura 9: Estrutura de Arquivo Invertido

As etapas que constituem o processo de indexação são:

a) Análise léxica: etapa para converter uma cadeia de caracteres em uma cadeia de palavras; um analisador léxico pode ser utilizado para identificar as seqüências de caracteres no texto. O analisador separa o alfabeto de entrada em caracteres de palavra (letras de a-z) e separadores de palavras (espaço, nova linha, etc);

b) Remoção de *Stop-Words*: esta fase tem por objetivo filtrar e retirar as palavras que ocorrem na maioria dos documentos. Uma lista contendo as palavras que são consideradas irrelevantes deve ser criada para o idioma em que se está trabalhando. A lista é chamada de *stop-list* e o conjunto de palavras que a compõem são chamadas de *stop-words*. As *stop-words* são palavras que aparecem com muita freqüência, como por exemplo, considerando-se textos em inglês, tem-se: artigos (a, an, the,...), preposições (in, on, of,...), conjunções (and, or, but, if,...), pronomes (I, you, them, it...), alguns verbos, nomes, advérbios e adjetivos (make, thing, similar...) e, portanto, devem ser eliminadas do texto a ser indexado. A eliminação de *stop-words* melhora o tamanho das estruturas de indexação. Na figura 10, pode-se observar as *stop-words,* que estão sublinhadas.

Controls are placed on both the type of offers that can be extended to these people and the use of these lists by competitors. Restrictions are placed on the number of times the names can be accessed and when the names can be sold. And there is usually not much data available about these businesses and individuals because the owner does not want to sell anything considered proprietary or sensitive.

Figura 10: Stop-Words sublinhadas que serão removidas

c) *Stemming*: o objetivo desta fase é remover todas as variações de uma palavra, permanecendo apenas a raiz da palavra. As variações incluem plurais, gerúndios (ing), sufixos de terceira pessoa, sufixos de tempo passado, etc. Uma lista de sufixos *(suffixlist*) é pré-definida, dependente do idioma, contendo os sufixos das palavras. Por exemplo: a palavra *connect* pode conter as variações *connects*, *connected*, *connecting*, *connection*…; todas têm semânticas similares e relacionam-se a um mesmo conceito. Os prefixos e sufixos são eliminados das palavras, melhorando o armazenamento, pois menos termos são armazenados. Com o processo de *stemming*, uma consulta, por exemplo, sobre "*connect*" encontraria documentos que tenham "*connects*", "*connected*", "*connecting*", etc. A figura 11 ilustra um documento com os sufixos em tachado.

Control~~s~~ plac~~ed~~ type offer~~s~~ be extend~~ed~~ people use list~~s~~ competit~~ors~~. Restrict~~ions~~ plac~~ed~~ number time~~s~~ name~~s~~ be access~~ed~~ name~~s~~ be sold. usually data available busines~~ses~~ individual~~s~~ own~~er~~ want sell consider~~ed~~ proprietary sensitive.

Figura 11: Processo de Stemming

d) Determinação de pesos: São utilizadas medidas de freqüências relativas, capazes de identificar termos que ocorrem com substancial freqüência em alguns documentos de uma coleção, mas com baixa freqüência na coleção toda. Dentre as medidas de freqüência relativa, destaca-se o peso do termo. O peso do termo consiste em assumir que a importância do termo é proporcional à freqüência de ocorrência de cada termo k em cada documento i ($FREQ_{ik}$) e inversamente proporcional ao número de documentos para os quais o termo é encontrado ($DOCFREQ_k$). A freqüência inversa do termo k na coleção é expressa por [Sal83]:

$$IDOCFREQ_k = \log_2 \frac{n}{DOCFREQ_k} + 1$$

E o peso do termo pode ser representado por:

$$WEIGHT_{ik} = FREQ_{ik} * IDOCFREQ_k$$

Esta função atribui um alto grau de importância para termos que ocorrem em poucos documentos, levando-se em consideração a coleção de documentos.

e) Seleção dos termos-índice: serve para determinar quais palavras ou radicais serão usados como elementos de indexação. A decisão se uma palavra será usada como termo-índice está relacionada à natureza sintática da palavra. Os substantivos freqüentemente transmitem mais significados semânticos que adjetivos, advérbios, e verbos. Uma vez que é comum combinar dois ou mais substantivos em um componente único (por exemplo: *computer science*), faz sentido agrupar substantivos que aparecem próximos no texto em um componente de indexação único (conceito). Esse procedimento objetiva solucionar o problema de termos abrangentes, pois

os termos compostos categorizam melhor os assuntos. Individualmente, os termos geram uma ambigüidade maior do que a combinação dos dois termos, contribuindo para buscas mais específicas. Ao invés de simplesmente utilizar substantivos como termos-índice, pode-se utilizar um grupo de substantivos. Tal procedimento pode ser encontrado em [Bae99] e [Rib99].

Na figura 12 são exibidas as palavras-chave extraídas do texto utilizado como exemplo:

Figura 12: Palavras-chave extraídas

As palavras-chave selecionadas são as que possuem os maiores valores para o peso.

f) Criação de Tesauro (Dicionário Léxico): a construção de estruturas para categorizar os termos, tais como tesauros, permite expandir a consulta original com os termos relacionados. O processo de criação de Tesauro será discutido na seção seguinte.

3.2.2 Tesauro

Em Salton [Sal68], alguns refinamentos são sugeridos além do processo de indexação, consistindo de associações entre termos, conhecidas como classes Tesauro. Segundo ele, Tesauro é um agrupamento de palavras, ou radicais de palavras em certas categorias de assunto chamadas de “classes conceituais”.

Os objetivos da criação do Tesauro são: a) a padronização das palavras-chave que foram encontradas, ou seja, fornecer um vocabulário padrão (ou sistema de referências) para indexação e pesquisa; b) ajudar os usuários a localizar termos para a formulação de consultas; c) fornecer hierarquias de classes que permitem ampliar e limitar as consultas de acordo com as necessidades dos usuários. A vantagem de se usar um Tesauro é que se tem um vocabulário controlado para pesquisa e indexação.

A forma mais simples de um Tesauro consiste em uma lista de palavras (conceitos) importantes em um domínio de conhecimento e, para cada palavra na lista, um conjunto de palavras relacionadas. As palavras relacionadas são variações derivadas de um relacionamento sinônimo. A similaridade entre itens diferentes pode ser medida simplesmente pela quantidade de sobreposição entre os vocabulários.

Um Tesauro pode ser genérico para uma determinada língua (por exemplo, Inglês) ou pertencer a um domínio de conhecimento (por exemplo, domínio médico).

O Tesauro fornece um vocabulário padrão para a indexação, ajudando o usuário a encontrar termos similares e apropriados para a consulta. Ele é representado como um grafo, onde cada nó representa um termo que está ligado a outros termos. As ligações apresentam pesos diferentes para representar associações diferentes.

Para se criar um Tesauro é necessário calcular a similaridade entre os pares de termos encontrados no processo de indexação [Bae99]. A fórmula utilizada para o cálculo de similaridade, sugerida por Salton [Sal83], baseia-se em um vetor de documentos contendo os pesos associados de cada termo para o referido documento.

Seja $D_i = \langle w_{i1}, w_{i2}, ..., w_{in} \rangle$ um vetor de documentos onde cada w_{ij} é o peso ou freqüência associada ao termo *j* do documento D_i. Por exemplo, suponha que os termos *computer*, *technology* e *devices* possuam os pesos associados com D_1 = <2,4,0>. Isso significa que o documento D_1 é associado ao termo *computer* com peso 2, *technology* com peso 4 e *devices* com peso 0. O tamanho do vetor D corresponde ao número de termos distintos pertencentes aos documentos da coleção. Pesos 0 são estabelecidos para termos que não estão associados a um dado vetor de documento.

Uma coleção de documentos é representada por uma matriz, tal como na figura 13:

Vetor de Documentos	Termos de Indexação			
	T_1	T_2	...	T_t
D_1	w_{11}	w_{12}	...	w_{1t}
D_2	w_{21}	w_{22}	...	w_{2t}
⋮	⋮	⋮		⋮
D_n	w_{n1}	w_{n2}	...	w_{nt}

Figura 13: Matriz de Documentos

Cada linha da matriz representa um vetor de documentos e as colunas identificam os termos associados aos documentos. Segundo Salton [Sal83], dados vetores de termos na forma $TERMO_j = (w_{1j}wt_{2j}, ..., w_{nj})$, onde w_{ij} indica o peso do $TERMO_j$ no documento *i* e assumindo *n* documentos na coleção, uma medida de similaridade é expressa por:

$$SIMILAR(TERMOk, TERMOh) = \frac{\sum_{i=1}^{n} w_{ik} w_{ih}}{\sum_{i=1}^{n} (w_{ik})^2 + \sum_{i=1}^{n} (w_{ih})^2 - \sum_{i=1}^{n} w_{ik} w_{ih}}$$

A matriz de associação termo a termo é criada, contendo $S(T_i,T_j)$ representando o valor resultante da similaridade entre os termos T_i e T_j. A figura 14 ilustra a matriz de associação termo a termo:

Com base nessa matriz de similaridade, coloca-se em uma classe comum todos os termos cujos coeficientes de similaridade são suficientemente grandes em relação a um limite pré-estabelecido.

Termos de Indexação	T_1	T_2	...	T_t
T_1	$S(t_1,t_1)$	$S(t_1,t_2)$	...	$S(t_1,t_t)$
T_2	$S(t_2,t_1)$	$S(t_2,t_2)$	...	$S(t_2,t_t)$
⋮	⋮	⋮		⋮
T_n	$S(t_n,t_1)$	$S(t_n,t_2)$	...	$S(t_n,t_t)$

Figura 14: Matriz de Associação Termo a Termo

Uma representação de um Tesauro pode ser vista na figura 15 a seguir:

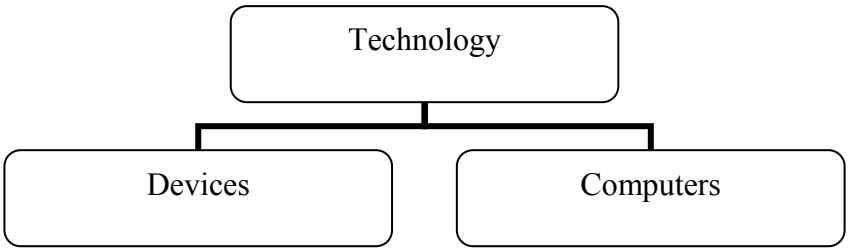


Figura 15: Representação de um Tesauro

Baseando-se na figura 15, um exemplo de utilização do Tesauro poderia ser efetuar uma busca por textos que contenham a palavra "*Computers*"; caso não fossem encontrados os textos, um conceito relacionado, tal como "*Devices*", seria utilizado para recuperar os documentos, aumentando a eficiência do sistema.

A manutenção do Tesauro pode ser necessária para acomodar o crescimento da coleção. Quando novos documentos são adicionados à coleção, várias estratégias são possíveis:

1) O Tesauro original deve ser deixado inalterado e usado para expandir a coleção;

2) Novos termos derivados dos itens adicionados devem ser colocados em categorias já existentes;

3) Novos termos devem ser colocados em novas classes separadas;

4) O Tesauro deve ser completamente reestruturado gerando uma nova classificação do vocabulário atualizado.

A estratégia a ser utilizada é escolhida de acordo com a necessidade do Sistema de Recuperação de Informação.

Algumas estratégias de recuperação encontradas na literatura são apresentadas a seguir.

3.2.3 Estratégias de Buscas

Um sistema de Recuperação de Informação é composto por documentos, um processo de indexação que gera estruturas de dados (índices) para os documentos, um processo de especificação de consultas que, a partir das necessidades do usuário, gera a consulta formulada, e finalmente o processo de recuperação que, a partir das estruturas de dados e da consulta formulada, recupera uma lista de documentos considerados relevantes.

A definição do conjunto de documentos recuperados depende do modelo utilizado. Os modelos de recuperação existentes para os Sistemas de Recuperação de Informação são: a) Modelo Booleano, b) Modelo Vetorial e c) Modelo Probabilístico [Sal83] [Bae99].

a) Modelo Booleano: é o modelo mais simples de recuperação, baseado na Álgebra Booleana. O modelo booleano considera uma consulta como uma expressão booleana convencional formada com os conectivos lógicos AND, OR e NOT. A estratégia de recuperação é baseada no critério de decisão binária. As expressões booleanas para a recuperação baseada em conteúdo de documentos texto são as seguintes:

Termos simples - "*databases*", "*association*", "*APriori*".

Expressões com conjunções - "*association*" AND "*APriori*".

Expressões com disjunções - "*databases*" OR "*association*".

Expressões com negações - NOT "*APriori*".

Considere uma consulta Q = (t_1 AND t_2 AND t_3) OR (t_4), onde t_i representa um termo. Sendo D_n o conjunto de documentos recuperados para Q, a representação é dada por $(D_{t1} \cap D_{t2} \cap D_{t3}) \cup (D_{t4})$. Para a coleção de documentos ilustrada a seguir, o conjunto de documentos recuperados para Q é $\{d_1,d_2,d_3,d_4\}$.

Termos dos documentos

$d_1 = \{t_1, t_2, t_3,t_5\}$

$d_2 = \{t_1,t_2, t_3, t_4\}$

$d_3 = \{t_1, t_3, t_4,t_5\}$

$d_4 = \{t_3, t_4\}$

$(D_{t1} \cap D_{t2} \cap D_{t3}) \cup (D_{t4})$

No modelo booleano básico, a ausência de ordem de resposta é um dos principais problemas.

Neste trabalho de mestrado, esse problema é tratado baseando-se no peso do termo para o documento e sua relevância para a consulta.

b) Modelo Vetorial: O modelo de espaço vetorial, ou simplesmente modelo vetorial, representa documentos e consultas como vetores de termos. O vetor resultado para uma consulta Q é montado através de um cálculo de similaridade baseado no ângulo entre o vetor que representa o documento e o vetor que representa a consulta.

Considere um espaço vetorial *t-dimensional E,* formado pelo conjunto de *t* termos da coleção, ou seja o vocabulário. Nesse espaço *E,* cada termo é interpretado como uma dimensão ortogonal. Documentos e consultas são representados por vetores nesse espaço *t-dimensiona1*. A seguinte notação é adotada:

t_i: i-ésimo termo, onde 1 <= *i* <= *t*;
$E = (t_1,...,t_t)$ espaço vetorial;
w_{id} : peso do termo *i* no documento *d;*
w_{iq}: peso do termo *i* na consulta *q*;
$\vec{d} = \{w_{1d},...,w_{td}\}$: o vetor de documento *d*;
$\vec{q} = \{w_{1q},...,w_{tq}\}$: o vetor de consulta *q*;

Os pesos quantificam a relevância de cada termo para as consultas e para os documentos no espaço vetorial *E.*

Existem várias formas de calcular o peso dos termos nos vetores, bem como de calcular a similaridade entre cada documento e a consulta. Em relação à similaridade, uma forma bastante difundida na literatura para o cálculo da similaridade entre um vetor de documento $\vec{d}$ e um vetor de consulta $\vec{q}$ é dado pelo cosseno entre esses dois vetores. Esse cálculo é feito através da função (1). Essa e outras funções para o cálculo da similaridade entre vetores podem ser encontradas em [Sal83].

$$sim(d,q) = \frac{\sum_{i=1}^{t} w_{id} \cdot w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{id}^2} \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \qquad (1)$$

Essa medida de similaridade calcula o cosseno do ângulo entre documentos ou entre consultas e documentos, quando estes são vistos como vetores em um espaço vetorial *t-dimensional.*

O exemplo a seguir mostra o cálculo de similaridade entre um vetor de consulta e cinco vetores de documentos, onde estão envolvidos 6 termos.

Vetores de documentos, onde $d_i = \{t_1,t_2,t_3,t_4,t_5,t_6\}$:

$d_1 = \{0, 2, 1,3,0,0\}$

$d_2 = \{3, 0, 0,2,1,4\}$

$d_3 = \{2, 0, 0,3,0,2\}$

$d_4 = \{1, 2, 1,3,1,1\}$

$d_5 = \{3, 0, 0,2,0,1\}$

Os valores nos vetores indicam a ocorrência de cada termo t_i no documento. Por exemplo, o documento d_1 não possui os termos t_1, t_5 e t_6, mas possui os termos t_2, t_3, t_4 com os pesos 2, 1, 3, respectivamente. Dessa mesma forma se define o vetor de consulta abaixo.

Vetor da Consulta: $q =- \{0,1,1,1,0,0\}$

Cálculo das similaridades entre o vetor de documento d_1 e o vetor de consulta q:

$$sim(d1,q) = \frac{(0.0 + 2.1 + 1.1 + 3.1 + 0.0 + 0.0)}{\sqrt{0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} \cdot \sqrt{0^2 + 1^2 + 1^2 + 1^2 + 0^2 + 0^2}} = 0.895$$

Da mesma maneira:

$sim(d_2,q) = 0.211$

$sim(d_3,q) = 0.420$

$sim(d_4,q) = 0.839$

$sim(d_5,q) = 0.300$

Isso resulta no seguinte vetor, ordenado por grau de satisfação da consulta:

$R=\{d_1,d_4,d_3,d_5,d_2\}$

Segundo Salton [Sal83], o modelo vetorial se comporta bem com coleções genéricas.

c) Modelo Probabilístico: o modelo probabilístico descreve documentos considerando pesos binários que representam a presença ou ausência de termos. O vetor resultado gerado pelo modelo tem como base o cálculo da probabilidade de que o documento seja relevante para a consulta. O *princípio probabilístico de ordenação* estabelece que o modelo probabilístico pode ser usado de uma forma ótima. Este princípio é baseado na hipótese de que a relevância de um documento para uma consulta é independente de outros documentos da coleção.

Se a resposta de um sistema de recuperação a cada requisição é uma ordem de documentos classificada de forma decrescente pela probabilidade de relevância para o usuário

que submeteu a requisição, onde as probabilidades são estimadas com a melhor precisão possível com base nos dados disponíveis, então a efetividade geral do sistema para seu usuário, será a melhor que pode ser obtida com base naqueles dados. O modelo probabilístico considera um processo iterativo de estimativas da probabilidade de relevância. O equacionamento apresentado a seguir considera que, de alguma forma, as probabilidades são conhecidas.

Considerando,

$P(+R_q|d)$ *:* a probabilidade de que um documento seja relevante para a consulta q, dado que o documento é *d*

$P(-R_q|d)$ *:* a probabilidade de que um documento seja não relevante para a consulta q, dado que o documento é *d.*

O documento d é considerado relevante para a consulta q se $P(+R_q|d) > P(-R_q|d)$, e o vetor resultado é decidido com base num fator $W_{d|q}$, definido pela seguinte equação,

$$W_{d|q} = \frac{P(+R_q \mid d)}{P(-R_q \mid d)}$$

O modelo probabilístico tem como desvantagem que o desempenho depende da precisão das estimativas de probabilidade. Além disso o método não explora a freqüência do termo no documento.

Os Sistemas de Recuperação de Informação ainda contam com medidas de eficiência das buscas realizadas. Algumas destas medidas são descritas na próxima seção.

3.2.4 Medidas de Eficácia

Algumas medidas são utilizadas para verificar a habilidade do sistema em satisfazer o usuário, ou seja, verificar quão satisfatória foi a resposta para uma determinada consulta. Algumas das medidas mais conhecidas são: *Recall* e *Precision* [Sal68]. *Recall* é a porcentagem de todas as respostas (documentos) relevantes que são efetivamente recuperadas por uma consulta em relação ao total de respostas relevantes previstas. *Precision* é a porcentagem de respostas relevantes efetivamente recuperadas numa consulta em relação ao total de respostas obtidas.

As medidas *recall* e *precision* para recuperação de textos também podem ser encontradas em [Kam01] e podem ser expressas como na Figura 16.

Um sistema de recuperação é considerado eficaz se possuir alto valor para *recall* e alto valor para *precision*.

Relevant:	Conjunto de documentos relevantes a uma consulta.
Retrieved:	Conjunto de documentos recuperados
Relevant ∩ *Retrieved*:	Conjunto de documentos relevantes e recuperados.

$$precision = \frac{|relevant| \cap |retrieved|}{|retrieved|}$$

$$recall = \frac{|relevant| \cap |retrieved|}{|relevant|}$$

Figura 16: Medidas de Eficácia para Recuperação de Textos

3.3 Extração de Informação

A extração de informação é usada na área de Processamento de Linguagem Natural (PLN) com o objetivo de transformar dados semi-estruturados ou desestruturados (textos) em dados estruturados que serão armazenados em um banco de dados. Uma vez estruturados, esses dados podem ser utilizados para processos tradicionais de descoberta de conhecimento. A extração de informações é geralmente utilizada como um passo anterior à mineração, e, portanto considerada uma etapa de pré-processamento.

A descoberta de conhecimento pode ser realizada sobre documentos protótipos produzidos com as técnicas de extração de informação, como apresentado em [Raj97] ou ainda ser utilizada sobre *templates.* O protótipo é informalmente definido como um texto contendo informações que ocorrem repetidas vezes na coleção de documentos, representando a classe de documentos similares na base textual. As técnicas utilizadas para pré-processar os textos são pertencentes à área de Processamento de Linguagem Natural (PLN) e agem identificando os termos freqüentes na coleção de documentos ou extraindo informações baseadas em regras pré-definidas.

As técnicas para extração de informação permitem rastrear o texto ou um conjunto de textos com o objetivo de extrair fatos presentes nestes. O sistema de extração depende do domínio específico e deve ser pré-programado ou treinado para reconhecer cada campo no texto de entrada [Dix97].

Uma comparação pode ser feita entre a extração de informação e a indexação (utilizada por sistemas de recuperação de informação). Enquanto a indexação permite identificar palavras capazes de caracterizar o documento e colocá-las em um índice, a extração identifica palavras dentro de conceitos específicos e ainda contém um processo de transformação que modifica a informação extraída em um formato compatível com um banco de dados [Kow97].

Do ponto de vista de Processamento de Linguagem Natural, a Extração de Informação é atrativa porque: a) as tarefas de extração são bem definidas (dependentes do domínio); b) a performance da extração pode ser comparada à performance humana em executar a tarefa de extração [Cow96].

A extração de informação pode ser útil em uma variedade de domínios. Há vários anos, conferências denominadas *Message Understanding Conference* (MUC's) utilizam sistemas que implementam conceitos de extração de informação ou recuperação de informação. As MUC's enfocam o preenchimento de *templates* com informação específica de um determinado domínio [Cal98].

Os Sistemas de Extração de Informação não tentam interpretar o texto em todas as partes do documento de entrada, mas sim analisam partes do texto que possuam informações relevantes ao domínio específico. Nas seções seguintes será analisado um processo de extração de informação.

3.3.1 Processo de Extração

Um Sistema de Extração de Informação trata uma coleção de textos, isolando fragmentos de texto para a extração de informações relevantes dos fragmentos. O sistema armazena as informações extraídas na forma tabular. Desse modo, a informação disponível em texto desestruturado pode ser traduzida para bancos de dados tradicionais para que os usuários possam explorar através de consultas padrões. Uma aplicação de extração de informação pode ser encontrada em [Cow96].

A extração de informação utilizada em Processamento de Linguagem Natural deve ser feita sobre um domínio pré-definido com normas sobre o tipo de informação que o sistema espera encontrar. Esse domínio contendo as informações que se espera encontrar pode ser expresso por *Slots*. Os *slots* são itens de dados ou partes de informação (formados por pares: atributo-valor, como campos num banco de dados tradicional), representando as informações que devem ser extraídas de um texto. Por exemplo, em textos sobre atentados terroristas, os *slots* poderiam conter o local onde ocorreu o atentado, a data e o número de vítimas.

Os passos envolvidos vão desde a criação de *slots* com as informações desejadas do domínio até a geração de *templates*.

Templates correspondem a documentos com lacunas a serem preenchidas com as informações extraídas dos textos.

A extração de informação pode ser representada como na Figura 17:

Figura 17: Extração de Informação

Baseando-se nos *Slots* criados, um analisador léxico verifica o texto a fim de preencher esses *slots*. O texto todo então é marcado com *tags* SGML (*Standard Generation Markup Language*) delimitando a informação a ser extraída. Tem-se, então, um texto na forma semi-estruturada.

As informações delimitadas por *tags* são extraídas, submetidas a tratamentos para serem armazenadas na forma tabular ou formatadas em um texto em linguagem natural; as técnicas de extração utilizam *templates*, ou seja, um extrator é utilizado para um determinado conjunto de *slots* desejados de um domínio, em seguida um banco de dados é construído de uma coleção de textos aplicando-se o extrator em cada documento para criar uma coleção de registros estruturados. Técnicas de Mineração de Dados podem ser aplicadas ao banco de dados resultante para descobrir relacionamentos interessantes.

Os Sistemas de Extração de Informação utilizam técnicas da área de Processamento de Linguagem Natural (PLN) para pré-processar o texto.

3.3.2 Marcação POS

O objetivo da marcação *Part-of-Speech* (POS) é automaticamente atribuir marcas (*tags*) de discurso (isto é, categorias morfossintáticas, tais como substantivo, verbo, adjetivo, etc.) para palavras no contexto. A principal dificuldade é a ambigüidade léxica que existe na linguagem natural. Por exemplo: "*process*" e "*programs*" poderiam ser tanto substantivos como verbos [Raj97].

A Figura 18 mostra um texto sobre "anúncio de emprego" com as *tags* atribuídas após a aplicação da técnica de marcação POS:

As *tags* mostradas após a barra (/) indicam a categoria morfossintática de cada palavra no texto. As categorias encontradas nesse texto foram: **N** – nome (substantivo), **V** – verbo, **ADJ** – adjetivo, **PR** – preposição e **DET** – artigo indefinido.

Uma vantagem da marcação POS é permitir filtrar palavras não significantes baseando-se na categoria morfossintática, como por exemplo, artigos, preposições, conjunções, etc, restringindo a extração aos substantivos, adjetivos e verbos. Os programas utilizados para marcação POS são chamados de *taggers* e permitem também o reconhecimento preliminar de unidades de frases em sentenças.

Internet/**N** Provider/**N** using/**V** cutting/**ADJ** edge/**N** web/**N** technology/**N** in/**PR** Austin/**N** is/**V** accepting/**V** applications/**N** for/**PR** a/**DET** Senior/**ADJ** Software/**N** Developer/**N**. The candidate/**N** must/**V** have/**V** 5 years/**N** of/**PR** software/**N** development/**N**.

…

Figura 18: Texto marcado com tags mostrando a categoria morfossintática

Há várias técnicas utilizadas pelos *taggers* para a marcação de textos; dentre elas destaca-se a abordagem baseada em regras [Bri92].

Um exemplo de *tagger* baseado em regras é o sistema *RAPIER* (*Robust Automated Production of Information Extraction Rules*) que aprende regras para executar a tarefa de extração. O *RAPIER* é um sistema de aprendizado de máquina para induzir regras na extração de informação de textos em linguagem natural. Ele aprende padrões descrevendo restrições para o preenchimento de *slots,* tais como palavras específicas, categorias morfossintáticas ou classes semânticas. Maiores detalhes sobre o *RAPIER* são encontrados em [Cal97] e [Cal98].

A utilização do *RAPIER* é feita sobre alguns documentos com os *slots* preenchidos (chamados de conjunto de treinamento) para que ele adquira a base de conhecimento das regras de extração e então possa ser utilizado em novos documentos.

title: Sênior Software Developer

city: Austin

language: Perl, C, Javascript, Java

platform: NT, Windows

application: Oracle, Informix, Sybase

required years of experience: 5

Figura 19: Exemplo de slots preenchidos

A figura 19 é referente ao exemplo anterior sobre "anúncio de emprego" e mostra os *slots* preenchidos com as informações.

$$Perl \in language$$
$$C \in language$$
$$Javascript \in language$$
$$NT \in platform$$
$$Windows \in platform$$
$$Oracle \in application$$

Figura 20: Exemplo de regras

O *RAPIER* aprende regras a partir dos *slots* preenchidos, tais como as regras mostradas na figura 20. O documento com as *tags* é mostrado na figura 21:

<DOCUMENT>

Internet/N Provider/N using/V cutting/ADJ edge/N web/N technology/N in/PR **<CITY>** Austin **</CITY>** is/V accepting/V applications/N for/PR a/DET **<TITLE>** Senior_Software_Developer **</TITLE>**. The candidate/N must/V have/V **<YEARS_EXPERIENCE>** 5_years **</YEARS_EXPERIENCE>** of/PR software/N development/N and experience with **<APPLICATION>** Oracle, Sybase **</APPLICATION>**.

…

</DOCUMENT>

Figura 21: Documento com as tags

Depois de ter obtido o texto marcado com a informação desejada, o próximo passo é a extração propriamente dita, como discutido na seção seguinte.

3.3.3 Extração de Termos

A Extração de Termos visa detectar termos complexos dependentes de domínio. Os métodos de extração de termos geralmente são decompostos em duas situações [Dai94]:

- extração de termos candidatos baseada na informação lingüística; por exemplo, os termos candidatos poderiam ser selecionados de acordo com padrões morfossintáticos (tal como '**ADJ N**': Efficient Mining);

- filtragem dos termos candidatos baseada em algum método de classificação estatística, como por exemplo, a freqüência dos termos. A freqüência dos termos é importante no caso de criação de documentos protótipos.

3.4 Considerações Finais

Este capítulo apresentou características dos Sistemas de Recuperação de Informação e dos Sistemas de Extração de Informação; os últimos utilizam técnicas de Processamento de Linguagem Natural.

Os Sistemas de Recuperação de Informação executam várias etapas para a determinação de palavras-chave que identificam o conteúdo de um documento.

A recuperação dos documentos é feita com base em arquivos invertidos, onde os documentos estão associados às palavras-chave que esses contêm.

Além do processo de indexação de documentos, ainda é possível realizar uma associação entre os termos existentes na coleção de documentos. Tal associação é conhecida como classes Tesauro, a qual objetiva ampliar ou refinar o processo de busca.

Medidas de eficiência, conhecidas como *precision* e *recall,* podem ser utilizadas para avaliar a satisfação do usuário em relação à consulta realizada.

A extração de informação pode ser utilizada como pré-processamento de documentos, com a finalidade de obter uma forma estruturada onde serão aplicados algoritmos de mineração de informação.

Dentre as técnicas de Processamento de Linguagem Natural destacam-se a marcação POS, a qual utiliza um *tagger* capaz de aprender regras para executar a tarefa de extração.

Neste trabalho de mestrado, foram utilizadas as técnicas de Recuperação de Informação com base no modelo vetorial. Para possibilitar a classificação dos documentos recuperados,

considera-se o grau de similaridade entre os termos e a relevância destes para a consulta. Maiores detalhes sobre a estratégia de recuperação de documentos são apresentados no Capítulo 5.

No próximo capítulo, é apresentado o Ambiente AMMO, onde o Sistema para Manipulação de Documentos está inserido.

Capítulo 4

Ambiente AMMO

4.1 Introdução

O Sistema para Manipulação de Documentos baseada em Informação Semântica, apresentado no Capítulo 5, está inserido no projeto AMMO (*Authoring and Manipulation of Multimedia Objects*) que é desenvolvido pelo Grupo de Banco de Dados do Departamento de Computação da Universidade Federal de São Carlos.

O projeto AMMO foi concebido visando a criação, armazenamento e manipulação de aplicações multimídia, onde o usuário pode consultar uma aplicação multimídia usando consulta exata ou nebulosa através da World Wide Web.

O desenvolvimento do projeto AMMO foi motivado pela crescente necessidade de armazenar e recuperar aplicações multimídia, ou partes de uma aplicação, como uma cena ou até mesmo objetos de mídia que a compõem.

As aplicações estão estruturadas usando o padrão SMIL (*Synchronized Multimedia Integration Language*), que permite que a aplicação seja executada na Web com a ajuda de uma ferramenta de apresentação [Vie02].

A arquitetura do ambiente é apresentada na seção seguinte.

4.2 Arquitetura do Ambiente AMMO

O projeto AMMO possui os seguintes componentes:

Multimedia Objects Server (MmOS): responsável por gerenciar as aplicações multimídia armazenadas em um banco de dados multimídia;

Multimedia Application WebBuilder (MAW): provê o suporte à autoria e manipulação de aplicações multimídia na Web.;

Interface WEB: provê uma interface para realizar consultas ao banco de dados multimídia e também criar aplicações através da Web;

Sistema Gerenciador de Banco de Dados Orientado a Objetos (SGBDOO): gerencia os objetos mantidos no banco de dados multimídia.

A arquitetura do ambiente AMMO é ilustrada na Figura 22:

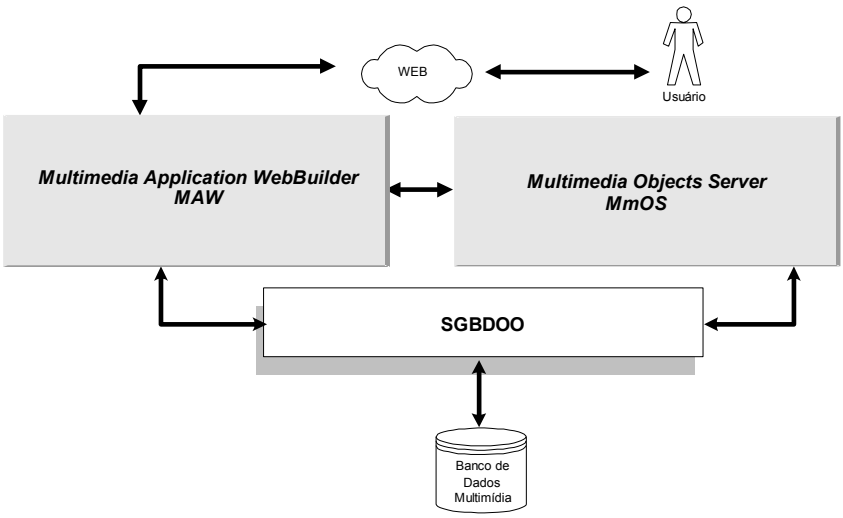


Figura 22: Estrutura do Ambiente AMMO

Dentro do módulo *Multimedia Application WebBuilder (MAW)* há uma ferramenta denominada MAE (*Multimedia Authoring Environment),* a qual foi inicialmente desenvolvida por Figueiredo [Fig00] e, posteriormente estendida por Santos [San03]. Esta ferramenta tem por objetivo auxiliar o processo de criação e alteração de aplicações multimídia.

Há também um módulo de consultas inserido no *Multimedia Objects Server (MmOS)*, onde é possível realizar buscas no banco de dados de aplicações multimídia usando busca por conteúdo exato ou nebuloso baseando-se em informações sobre o conteúdo das mídias [Bor01] [Vie02]. Para viabilizar essas buscas, foi criada uma estrutura de classes onde as informações semânticas relativas às mídias são armazenadas no ambiente.

Na próxima seção, é apresentada a estrutura do banco de dados multimídia do AMMO para armazenar as aplicações multimídia.

4.3 Banco de Dados Multimídia (BDMm)

As aplicações multimídia do projeto AMMO são compostas por documentos criados através da linguagem SMIL. A linguagem SMIL é uma linguagem de marcação definida a partir de XML (*Extensible Markup Language*), com o objetivo de integrar e sincronizar mídias em aplicações multimídia na Web.

Um documento SMIL, no ambiente AMMO, é considerado uma cena dentro de uma aplicação multimídia e, portanto, uma aplicação é um conjunto de cenas. A estrutura de classes do banco de dados multimídia para armazenar as aplicações SMIL e informações semânticas pode ser vista na figura 23, modelada em UML (*Unified Modeling Language*). O propósito dessa estrutura é oferecer acesso às cenas e mídias da aplicação, assim como o acesso à informação semântica referente às mídias.

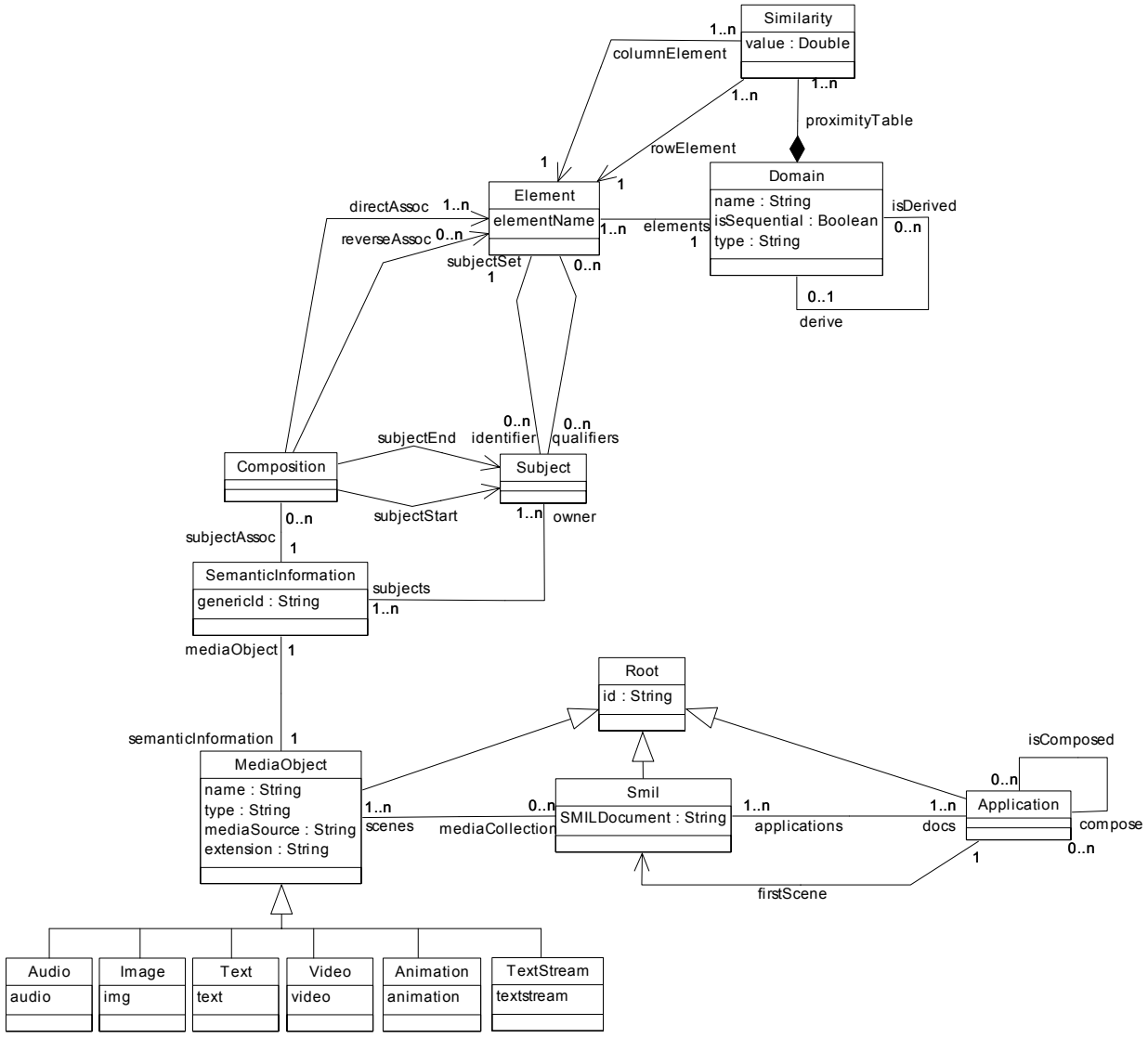


Figura 23: Estrutura de Classes do Ambiente AMMO

No ambiente AMMO [Vie02], uma aplicação multimídia é composta por um conjunto de documentos SMIL (*docs)*, onde cada documento SMIL possui um conjunto de mídias (*mediaCollection*) que estão generalizadas através da classe *MediaObject.* A associação *firstScene* indica qual é a primeira cena da aplicação e a seqüência de apresentação é definida nos próprios documentos SMIL através de marcações (*tags*) que representam ligações (*links)*

para outros documentos. Através dessas ligações o autor define a navegação entre as várias cenas que compõem a aplicação, sendo que apenas a primeira cena deve ser indicada para dar início à aplicação.

Cada documento SMIL pode pertencer a uma ou mais aplicações (*applications)*. O atributo *SMILDocument* armazena o código SMIL do documento. O conteúdo de uma mídia pode ser armazenado nos atributos *audio, img, text, video, animation* ou *textstream,* dependendo do seu tipo, ou ainda, apenas a localização pode ser indicada em *mediaSource* da classe *MediaObject.* A classe *MediaObject* também armazena informações gerais sobre a mídia, tais como seu nome, tipo, extensão. A extensão do arquivo contém informações para sua apresentação (MIME). Uma mídia pode pertencer a vários documentos SMIL sendo representado por *scenes*. A classe *Root* generaliza as características de todas as classes que representam uma aplicação SMIL. O atributo *id* dessa classe identifica uma aplicação, uma cena ou uma mídia.

Além das classes que representam aplicações multimídia, a estrutura do BDMm é composta de classes que foram criadas para representar as informações semânticas sobre as mídias dessas aplicações. Tais classes dão suporte à busca por conteúdo e busca nebulosa no ambiente do AMMO.

A classe *SemanticInformation* armazena informações semânticas sobre as mídias (áudio, vídeo, imagem etc) e possui um identificador (*genericId)*. A informação semântica referente ao conteúdo da mídia é composta por temas (*subjects)* e composições (*subjectAssoc).* A classe *Subject* possibilita guardar esses temas que compõem uma mídia (por exemplo, uma paisagem pode ser composta pelos temas *árvores*, *cabana*, *montanha*, etc.). Um tema pode estar relacionado a uma ou mais informação semântica; esse relacionamento é representado pelo atributo *owner.* A classe *Composition* é destinada a armazenar associações entre temas existentes em uma mídia (por exemplo, árvore *próxima da* cabana). A indicação dos temas iniciais e finais é feita através de *subjectStart* e *subjectEnd,* respectivamente. É possível guardar também qualificadores (*qualifiers)* para temas (por exemplo, árvore *florida* e *frondosa*).

As classes *Domain* e *Similarity* são usadas para guardar informações sobre os domínios dos elementos considerados (temas, qualificadores e associações) e sobre a similaridade existente entre elementos de mesmo domínio. Essas classes foram criadas para dar suporte à busca nebulosa desenvolvida no projeto AMMO. Um exemplo de domínio seria *Meios de Locomoção,* que seria composto pelos elementos *carro*, *barco*, *avião*, *navio,* etc.

A classe *Element* destina-se a manter conjuntos de termos que são semanticamente semelhantes, sejam eles temas (guri, garoto, etc.), qualificadores (belo, bonito, etc.) ou

composições (perto de, próximo de, etc.). O atributo *elementName* armazena um conjunto de termos semanticamente semelhantes.

As cenas das aplicações multimídia podem conter mídias-texto. Para essas mídias é possível empregar os recursos apresentados no próximo capítulo, para indexação e recuperação das mídias, utilizando o sistema desenvolvido neste trabalho de mestrado.

Assim, a contribuição deste trabalho para o ambiente AMMO é a extração automática da informação semântica e a recuperação dos documentos com base nessas informações.

A estrutura de classes utilizada para Informações Semânticas foi adaptada para a utilização no Sistema para Manipulação de Documentos baseada em Informação Semântica, que será apresentada no próximo capítulo.

4.4 Considerações Finais

Neste capítulo foi apresentado o ambiente AMMO, no qual o Sistema para Manipulação de Documentos está inserido, sendo apresentadas as estruturas de classes criadas para o armazenamento das aplicações e informação semântica.

A contribuição deste trabalho de mestrado para o ambiente AMMO é a extração automática de informação semântica de mídias-texto e a recuperação dessas mídias através de uma estratégia de busca que leva em consideração as informações extraídas.

O próximo capítulo mostra as técnicas utilizadas para a criação do Sistema, apresentando a sua arquitetura, a estrutura de classes utilizada na sua construção, seus módulos e suas funcionalidades.

Capítulo 5

Sistema para Manipulação de Documentos Baseada em Informação Semântica

5.1 Introdução

O Sistema para Manipulação de Documentos baseada em Informação Semântica tem por objetivo extrair características que discriminam adequadamente o conteúdo de um documento, permitindo descobrir relacionamentos entre os termos dos documentos de uma coleção. A motivação para o desenvolvimento de tal sistema é propiciar a extração automática de informação semântica de documentos, conforme a estrutura do ambiente AMMO, armazenando-a em banco de dados, assim como permitir a recuperação de documentos através de buscas exatas ou nebulosas. As buscas nebulosas são buscas que envolvem valores de similaridade mínima, composição de termos e importância de um termo para a busca, representado pelo grau de relevância deste. O sistema foi desenvolvido para ser utilizado no ambiente AMMO, mas seu uso não está limitado a esse ambiente. O sistema foi projetado como um módulo autônomo que para ser utilizado independente do ambiente.

Considerando uma coleção de documentos do idioma inglês, o sistema os analisa individualmente, extraindo informações (palavras-chave) que representam o conteúdo de cada documento e, através da utilização de relacionamentos entre termos (clusterização), a coleção é subdividida em domínios e subdomínios, proporcionando maior riqueza de informações que são utilizadas no momento da recuperação. Tais informações são mantidas no banco de dados para evitar um novo processamento dos documentos a cada recuperação.

O sistema é composto pelos módulos de Tratamento de Documentos e Recuperação de Documentos. O módulo de Tratamento de Documentos é responsável por realizar a busca por novos documentos, armazenados localmente ou na Web, a extração de informação semântica, determinação de domínios e subdomínios e armazenamento no Banco de Dados de Informação Semântica. O módulo de Recuperação de Documentos é responsável por disponibilizar buscas exatas ou baseadas em conteúdo impreciso, na coleção de documentos. A arquitetura do sistema, juntamente com os módulos, é apresentada na seção seguinte.

5.2 Arquitetura do Sistema

A arquitetura do Sistema para Manipulação de Documentos é apresentada na figura 24:

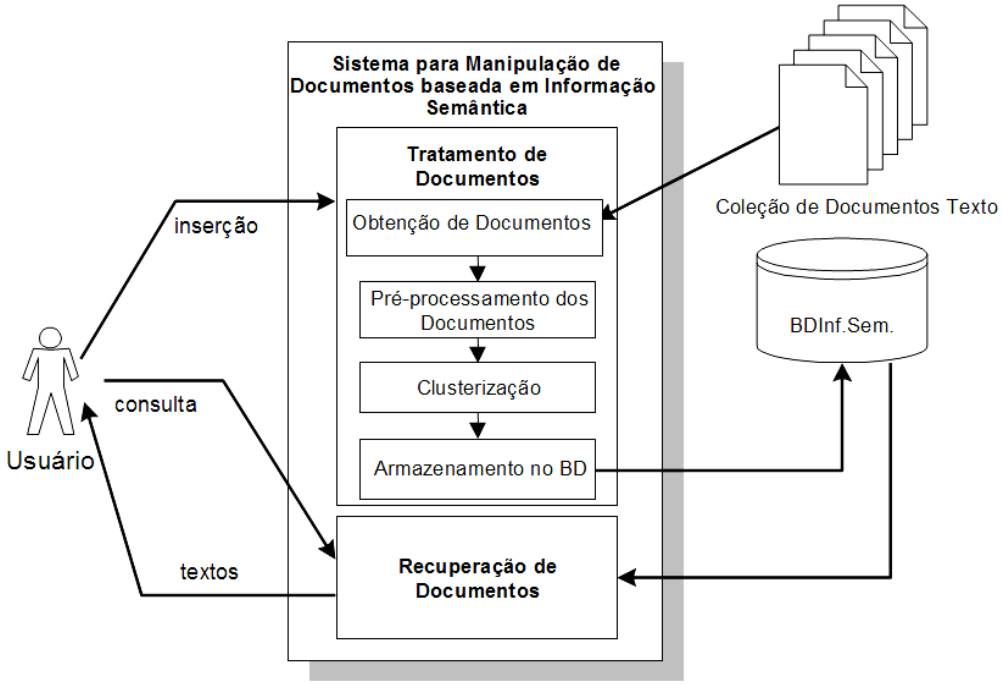


Figura 24: Arquitetura do Sistema para Manipulação de Documentos

O sistema é composto pelos Módulos de Tratamento de Documentos e Recuperação de Documentos, que interagem com o usuário via solicitação de consulta ou inserção de documentos.

O objetivo do Módulo de Tratamento de Documentos é a obtenção dos documentos para extração automática das informações semânticas destes. Tais informações são representadas por palavras-chave (termos), similaridades entre os termos, classificação destes em grupos (*clusters*) e armazenamento em um Banco de Dados de Informações Semânticas.

Para armazenar as informações semânticas referentes à mídia-texto, visando atender à abordagem adotada neste trabalho e para entendimento do sistema, uma visão do Banco de Dados é apresentada na figura 25, com base na estrutura de classes da figura 23 do Capítulo 4.

Tal visão representa as partes do Banco de Dados Multimídia (BDMm) que são utilizadas para o tratamento de textos. Fisicamente o Banco de Dados continua o mesmo, para que o mecanismo de armazenamento e recuperação dos textos e a informação semântica sejam apenas estendidos, aproveitando-se o que já existe.

Como pode ser visto na figura 25, a classe *Text* permite armazenar informações de textos, tais como o nome, tipo, extensão, a localização do texto, e o seu conteúdo, representados na estrutura respectivamente pelos atributos *name, type, extension, mediaSource* e *text*.

Cada texto possui uma representação de Informação Semântica (*SemanticInformation*). A classe *SemanticInformation* está relacionada com um ou mais termos extraídos do texto. Tais termos, representados na classe *Term*, correspondem às palavras-chave do texto. As informações sobre a freqüência e o peso do termo para um dado objeto da classe *SemanticInformation* são armazenadas em *TextTermRelation*.

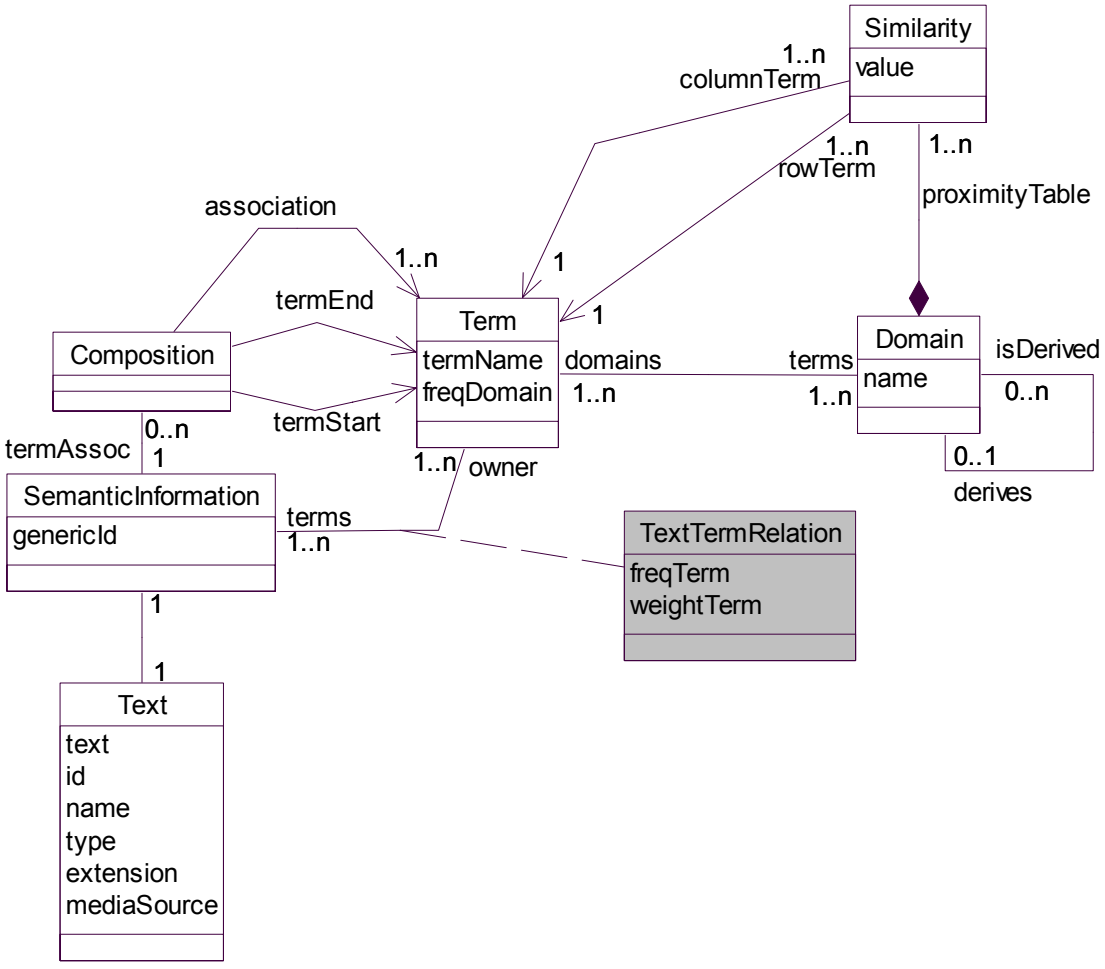


Figura 25: Estrutura de Classes para Informações Semânticas de Textos

Os termos podem pertencer a vários subdomínios de um domínio, criados pelo processo de clusterização. A classe *Domain* possui agregada a classe *Similarity*, que armazena o relacionamento entre os termos, expresso por valores de similaridade (*value).* A classe *Domain* permite representar um dicionário tesauro, através de domínios e subdomínios obtidos pela similaridade entre pares de termos. O dicionário é representado por um auto-relacionamento na classe *Domain,* onde há os atributos *derives* e *isDerived,* mostrando que um domínio deriva ou é

derivado de outro domínio, respectivamente. A classe *Composition* adiciona características no tratamento dos textos, pois é possível formar composições de termos, onde o termo de início (*termStart*) está associado ao termo final da composição (*termEnd*) através do termo referenciado por *association.*

Para exemplificar, suponha uma coleção de textos sobre "*data mining".* As informações, referentes à localização do texto, nome, extensão e conteúdo, são armazenadas em *Text.* A informação semântica é um conjunto de termos (palavras-chave), domínios e composições entre termos. As palavras-chave para a coleção de documentos em questão são: *mining, discovery, algorithms, categorization, clusterization, databases,* etc. O domínio para a coleção é "*data mining";* e a composição entre os termos pode ser expressa por "*mining* ***in*** *databases"*, onde o termo de início da composição é *mining,* o termo final é *databases* e o termo de associação é *in.*

Os termos são pertencentes a vários subdomínios de um determinado domínio e possuem um grau de similaridade entre eles. Através da similaridade é possível a construção de um tesauro, para representar o relacionamento entre os termos. Os níveis de relacionamento do tesauro são representados por subdomínios que derivam ou que são derivados de outro subdomínio. Considerando o exemplo citado, o nome do domínio é *"data mining",* todos os termos encontrados, extraídos dos textos, pertencem a esse domínio. Suponha que o termo *categorization* tem um valor de similaridade 0.95 em relação ao termo *algorithms,* que por sua vez tem um valor de similaridade de 0.80 em relação ao termo *databases;* se for adotado um grau de similaridade de 0.7, por exemplo, é possível criar uma classe para agrupar termos similares que estivessem acima deste grau de similaridade. O relacionamento entre o domínio e a classe criada é representado da seguinte forma:

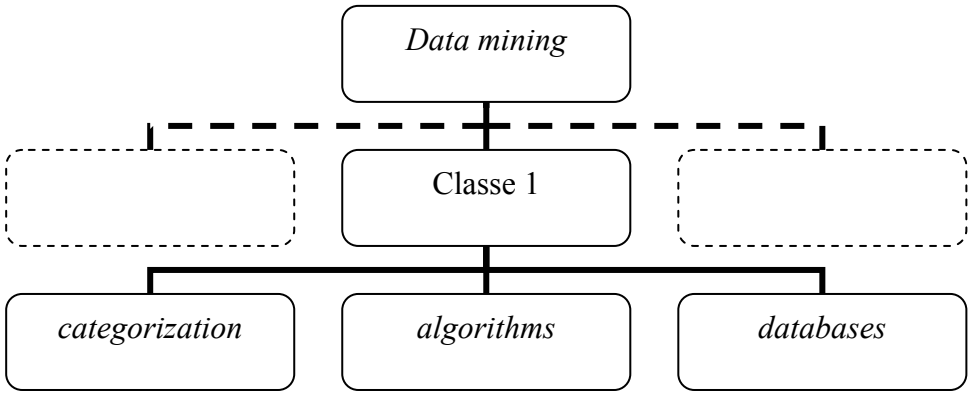


Figura 26: Representação da Hierarquia de Domínios e Subdomínios

A figura 26 representa que *categorization, algorithms e databases* pertencem à mesma classe e são derivados do domínio *data mining*.

Para efeitos de busca, se o usuário elaborar uma consulta com a expressão *data mining*, todos os termos pertencentes a este domínio serão utilizados na busca; entretanto, se o usuário desejar restringir a busca e utilizar apenas o termo *algorithms*, apenas os termos relacionados a ele serão utilizados na busca. Neste caso, os termos relacionados a *algorithms* são *categorization e databases.* Visto que *categorization* e *databases* têm um certo grau de similaridade com *algorithms,* considera-se que textos que tratam aspectos sobre *categorization* e *databases* (para o contexto de *data mining)* também tratam assuntos correlatos a *databases*. O resultado, portanto, abrange textos sobre *categorization, algorithms* e ainda *databases.*

Para permitir a disponibilização desta gama de informações semânticas relativas a textos, o Módulo de Tratamento de Documentos é composto pelos seguintes componentes (figura 24):

1. Obtenção de Documentos
2. Pré-processamento dos Documentos,
3. Clusterização,
4. Armazenamento no Banco de Dados.

Esses componentes são descritos detalhadamente nas próximas seções.

O Módulo de Recuperação de Documentos é ativado quando o usuário submete uma consulta ao sistema. A consulta é elaborada com os termos contidos no banco de dados e associações entre os termos (expressas pelas composições). Adicionalmente, o usuário pode informar a similaridade mínima desejada para os termos. Além disso, se desejar, pode definir qual é a importância de cada termo para a consulta, através da definição de um valor de relevância para eles. O Módulo de Recuperação é apresentado na seção 5.4, juntamente com a interface de busca.

5.3 Módulo de Tratamento de Documentos

O Módulo de Tratamento de Documentos é ativado quando o usuário opta por inserir novos documentos no Banco de Dados de Informação Semântica. Os componentes deste módulo são apresentados a seguir.

5.3.1 Obtenção de Documentos

Para a inserção de novos documentos, o usuário tem a opção de escolher documentos armazenados localmente, ou seja, que estão no disco rígido, ou em CD, ou ainda, pode optar por buscar os documentos na World Wide Web.

A busca por documentos na Web é realizada por um aplicativo chamado Wget [Wget01]. O aplicativo Wget é utilizado neste sistema para buscar arquivos, baseado na expressão de busca do usuário.

Para a extração de informação semântica, o usuário seleciona os arquivos de interesse e aciona o botão "*Process Files",* o que provoca a execução do Pré-processamento dos Documentos.

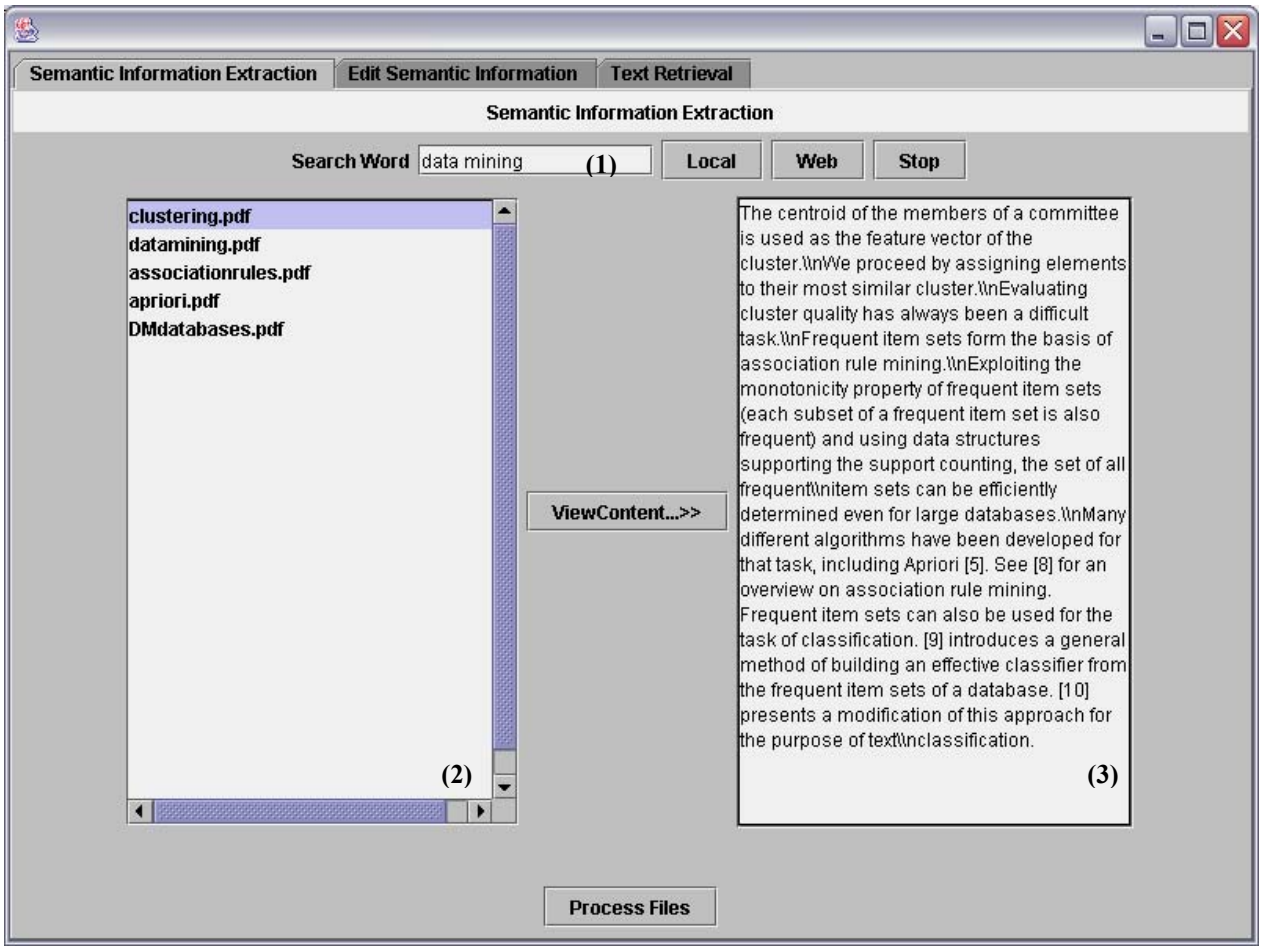


Figura 27: Obtenção de Documentos com Wget

Como pode ser visto na figura 27, o usuário digitou a palavra de busca *"data mining"* na área (1) e pressionou o botão "*Web",* para realizar buscas na Web. A partir da palavra de busca, o aplicativo Wget [Wget01] é ativado, realizando buscas recursivas em páginas Web, buscando e armazenando, no disco rígido, documentos que atendam à expressão. Os parâmetros passados para o aplicativo permitem que este realize buscas de documentos no formato desejado. Neste trabalho, o formato escolhido foi o *Portable Document Format (PDF)*. O site escolhido para a realização das buscas é o Google [Goo03]. O aplicativo Wget cria uma pasta com o mesmo

nome da palavra de busca, onde armazena os arquivos obtidos. Para interromper as buscas, o usuário pode pressionar o botão "*Stop*".

Quando o usuário pressiona o botão "*Stop*", o sistema exibe uma caixa de diálogo para que o usuário selecione os arquivos a serem visualizados. O lado esquerdo da figura 27 (área 2) lista os arquivos selecionados pelo usuário. Para visualizar o conteúdo de um arquivo, basta o usuário selecionar o arquivo e escolher *"View Content"*, o sistema exibe o conteúdo do arquivo no lado direito da figura 27 (área 3). Caso o documento seja maior do que a área de visualização, uma barra de rolagem é exibida para que o usuário possa ver o conteúdo todo do documento.

5.3.2 Pré-processamento dos Documentos

O pré-processamento inicia-se quando o usuário seleciona os arquivos (figura 27) e aciona o botão "*Process Files"*. Esta etapa tem como finalidade analisar cada um dos documentos selecionados, identificando quais termos serão definidos como palavras-chave. O resultado do pré-processamento é um conjunto de termos (palavras-chave) que identificam o conteúdo de um documento, juntamente com os pesos e freqüências associados aos termos. O documento é armazenado no banco de dados como um objeto da classe *Text,* enquanto os termos são armazenados como objetos da classe *Term,* os quais estão relacionados com o texto por meio das associações feitas entre *Term, SemanticInformation* e *Text.* As informações sobre peso e freqüência do termo são armazenadas na classe *TextTermRelation*.

O pré-processamento é executado para cada documento e é composto pelas seguintes etapas, baseado em [Sal83] :

1. Transformação do documento para o formato texto;
2. Limpeza e Padronização do Texto: Dígitos, quando aparecem no texto sem estarem associados a uma palavra, não representam informação relevante para o documento e, portanto, são retirados. Também é feita uma padronização que consiste em converter todos os caracteres do texto para o mesmo formato, ou todos os caracteres maiúsculos ou todos minúsculos. Por exemplo: *text*, *Text*, *TEXT* são convertidos para *text* (padrão minúsculo).
3. Remoção de *stop-words*: a remoção de *stop-words* é feita baseada em uma lista, chamada de *stoplist,* que contém as palavras que devem ser eliminadas do texto. As *stoplists* são criadas para um idioma específico, que neste caso é o inglês. Tal *stoplist* foi obtida do sistema KEA 2.0, desenvolvido por Frank et al. [Fra00]. Cada palavra do texto é comparada com as palavras contidas na *stoplist*; caso a palavra em questão seja encontrada na *stoplist,* esta deve ser retirada do texto

original, pois, devido à sua alta freqüência, não será representativa do conteúdo do texto.

4. *Stemming*: As palavras resultantes do processo anterior são submetidas ao algoritmo de *stemming.* O resultado deste processo é um conjunto de palavras na sua forma raiz. O algoritmo responsável por realizar esta tarefa é o algoritmo Porter [Por80], que, quando aplicado a palavras como *clustering, clusterization, clustered,* converte-as para *cluster*.
5. Determinação de pesos: Para cada palavra, resultante do processo de *stemming,* deve ser associado um peso indicando a relevância da palavra dentro do texto e na coleção. Para calcular o peso é utilizada a fórmula, apresentada no Capítulo 3, proposta por Salton [Sal83]:

$$WEIGHT_{ik} = FREQ_{ik} * IDOCFREQ_k \quad (1)$$

Os termos com fatores de peso mais altos são atribuídos como palavras-chave aos textos. A adoção de um número máximo de palavras-chave (*n*) por documento faz com que *n* palavras com pesos maiores sejam atribuídas como palavras-chave. Os termos obtidos são armazenados, como já dito anteriormente, como objetos da classe *Term* e seus valores de freqüência e peso ficam armazenados na classe *TextTermRelation,* servindo para a classificação dos resultados no momento da recuperação. A freqüência inversa ($IDOCFREQ_k$) corresponde à freqüência do termo na coleção e é armazenada como atributo da classe *Term (freqDomain)*.

A determinação de similaridades, para identificar relacionamentos entre termos, é abordada na próxima seção.

5.3.3 Clusterização

O objetivo da fase de clusterização é a identificação de grupos de termos similares associados à coleção de documentos. Tal processo é conhecido como dicionário tesauro. Para a construção do tesauro é necessário obter os valores de similaridades entre pares de termos resultantes do processo anterior.

A fórmula utilizada para o cálculo de similaridade, proposta por Salton [Sal83], baseia-se em um vetor de documentos contendo os pesos associados de cada termo para o referido documento, conforme apresentados no capítulo 3.

Para a clusterização, os seguintes passos são realizados:

1) Um vetor $D_i = \langle w_{i1}, w_{i2}, \ldots, w_{in} \rangle$ de documentos é gerado pelo sistema, onde cada w_{ij} é o peso ou freqüência associada ao termo *j* do documento D_i. A figura 28 mostra uma matriz de vetores de documentos, contendo suas palavras-chave e os pesos associados:

		Termos de Indexação de Documentos						
		mining	discovery	algorithms	categorization	clusterization	dictionary	databases
Vetores de Documentos	D1	3	4	0	0	0	0	2
	D2	2	3	2	0	0	1	1
	D3	0	0	3	5	7	1	1
	D4	1	4	1	0	1	0	2

Figura 28: Matriz de Vetores de Documentos

O processamento dos documentos selecionados identifica algumas palavras-chave, tais como os termos *mining*, *discovery, algorithms, categorization, clusterization, dictionary* e *databases*, respectivamente associados com D_1 = <3,4,0,0,0,0,2>. Isso significa que o documento D_1 , é associado ao termo *mining* com peso 3, *discovery* com peso 4, *algorithms* com peso 0, *categorization* com peso 0, *clusterization* com peso 0, *dictionary* com peso 0 e *databases* com peso 2.

2) Baseando-se na matriz de vetores de documentos, uma matriz de associação termo a termo é criada, contendo $S(T_i,T_j)$ representando o valor resultante da similaridade entre os termos T_i e T_j. A medida de similaridade é calculada de acordo com a expressão (2), extraída de [Sal83]. Essa expressão usa um fator de normalização, para limitar os valores entre 0 e 1.

$$SIMILAR(TERMOk, TERMOh) = \frac{\sum_{i=1}^{n} w_{ik} w_{ih}}{\sum_{i=1}^{n} (w_{ik})^2 + \sum_{i=1}^{n} (w_{ih})^2 - \sum_{i=1}^{n} w_{ik} w_{ih}} \qquad (2)$$

Por exemplo, o cálculo da similaridade entre os termos “categorization” e “clusterization” (figura 26) é feito da seguinte maneira:

$$SIMILAR(categorization, clusterization) = \frac{(0x0 + 0x0 + 5x7 + 0x1)}{(5^2 + 7^2 + 1^2) - (5x7)} = \frac{35}{40} = 0.87$$

A figura 29 mostra os valores de similaridade obtidos pela associação entre os termos:

		Termos de Indexação de Documentos						
		mining	discovery	algorithms	categorization	clusterization	dictionary	Databases
Termos de Indexação de Documentos	mining	–	0.66	0.22	0	0	0.14	0.26
	discovery	0.66	–	0.22	0	0	0.07	0.32
	algorithms	0.22	0.22	–	0.62	0.52	0.45	0.58
	categorization	0	0	0.62	–	**0.87**	0.23	0.68
	clusterization	0	0	0.52	**0.87**	–	0.15	0.68
	dictionary	0.14	0.07	0.45	0.23	0.15	–	0.18
	databases	0.26	0.32	0.58	0.68	0.68	0.18	–

Figura 29: Matriz de similaridade

3) O processo seguinte consiste em converter a matriz de similaridade em uma matriz binária, onde os pares de termos que possuem o valor de similaridade acima de um limite pré-estabelecido recebem o valor 1. Os valores de similaridade que estiverem abaixo do limite recebem 0.

Neste trabalho admitiu-se um limite para a similaridade k=0.6, o que representa que um termo é considerado similar ao outro se apresentar no mínimo um valor de 60% de similaridade entre eles. Outros trabalhos relatados na literatura [Sal83] [Rib99] [Net00], também adotam o valor de similaridade k=0.6. Trabalhos futuros devem ser realizados para analisar valores ideais para k.

Baseando-se no limite pré-estabelecido, a matriz binária (figura 30) é criada.

		Termos de Indexação de Documentos						
		mining	discovery	algorithms	categorization	clusterization	dictionary	Databases
Termos de Indexação de Documentos	Mining	–	1	0	0	0	0	0
	Discovery	1	–	0	0	0	0	0
	Algorithms	0	0	–	1	0	0	0
	categorization	0	0	1	–	1	0	1
	clusterization	0	0	0	1	–	0	1
	Dictionary	0	0	0	0	0	–	0
	Databases	0	0	0	1	1	0	–

Figura 30: Matriz Binária para k=0.6

4) Baseado na matriz binária usa-se o método de classificação automática (*single-link)* para construir classes de termos similares (equivalente a classes tesauros) [Sal83]. Esse método agrupa em uma classe comum todos os termos cujos coeficientes de similaridade são suficientemente grandes em relação a um limite pré-estabelecido (no exemplo 0.6). A figura 31 mostra a associação entre cada termo e seus termos similares:

Termo Original	Termos Similares
Mining	Discovery
Discovery	Mining
Algorithms	Categorization
Categorization	Algorithms, Clusterization, Databases
Clusterization	Categorization, Databases
Databases	Categorization, Clusterization

Figura 31: Termos associados com k=0.6

Desta forma, a formação das classes tesauros é feita criando-se uma classe para os termos que aparecem relacionados. Como pode ser visto na figura 31, os termos *mining* e *discovery* aparecem relacionados e portanto, podem ser colocados em uma classe (Classe 1). Utilizando o mesmo critério, os demais termos são agrupados, produzindo as seguintes classes:

Classe 1 – Mining, Discovery

Classe 2 – Algorithms, Categorization

Classe 3 – Categorization, Clusterization, Databases

Caso seja elaborada uma consulta contendo o termo "Categorization", são retornados os textos contendo os termos "Algorithms", "Clusterization" e "Databases".

Cada *cluster* é representado por um termo, no caso "Classe 1", "Classe 2", etc. e é armazenado no banco de dados como um sub-domínio do domínio "*data mining*". Os termos relativos àquele *cluster,* que já estavam armazenados como objetos da classe *Term,* são relacionados aos subdomínios criados. Todos os valores de similaridade entre os termos (figura 29) são armazenados na classe *Similarity* da Estrutura de Classes (figura 25).

Uma instanciação dos termos "mining" e "discovery" é exemplificada na figura 30. Neste exemplo, o termo "mining" possui freqüência no domínio igual a 6 (*freqDomain*=6). No documento DOC1 o termo apareceu 3 vezes (*freqTerm*=3), e portanto, o peso do termo no documento DOC1 é de 0.37 (*weightTerm*=0.37). O resultado é obtido a partir da expressão (1).

Quando o usuário elabora uma consulta, ele escolhe o domínio e o sistema exibe os termos daquele domínio. A partir daí, o usuário seleciona o termo desejado e o sistema busca todos os termos relacionados. Como pode ser visto na figura 32, supondo que o usuário tenha escolhido o termo "mining", o sistema verifica que este pertence à "classe 1", a qual contém outro termo relacionado ("discovery"), e então, recupera textos contendo ambos os termos "mining" e "discovery".

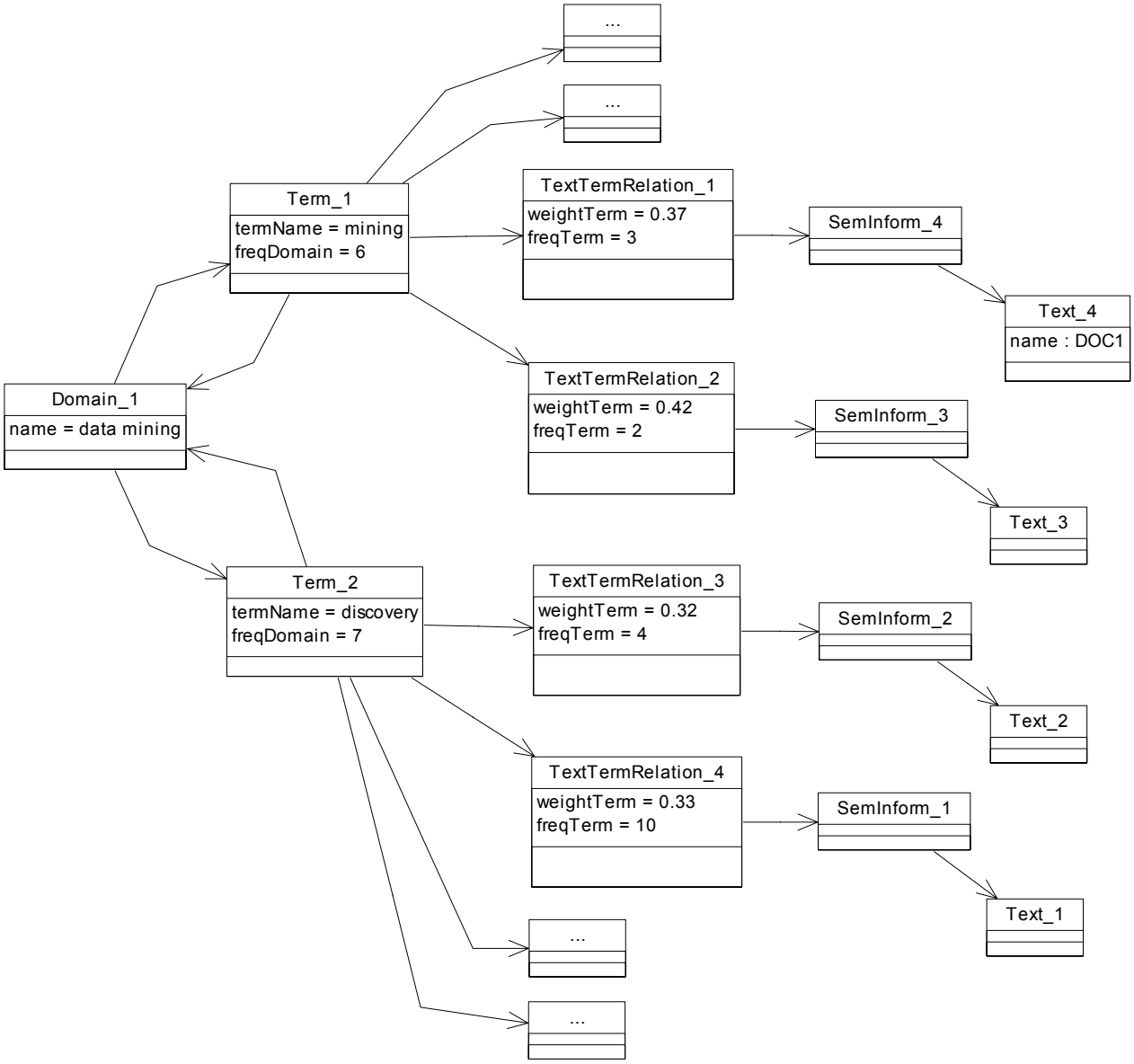


Figura 32: Exemplo de Instanciação dos Termos e Documentos

Além da informações extraídas automaticamente pelo sistema, o usuário pode acrescentar associações entre os termos (composições) aprimorando o conteúdo semântico do documento. Para tanto, a figura 33 exibe a interface utilizada para adicionar tais informações:

Como pode ser visto na figura 33, os nomes dos documentos armazenados no Banco de Dados de Informação Semântica são exibidos na área (1), o conteúdo do documento escolhido é exibido em (2), os termos ou palavras-chave do referido documento são apresentados em (3).

Para inserir as composições entre os termos, o usuário deve escolher, dentre os termos disponíveis em "*TermStart*" na área (4), "*TermEnd*" na área (6) e "*Association*" na área (5), o

termo que deseja para o início da composição, o termo final da composição e o termo de associação, respectivamente, e então, acionar o botão "*AddComposition*".

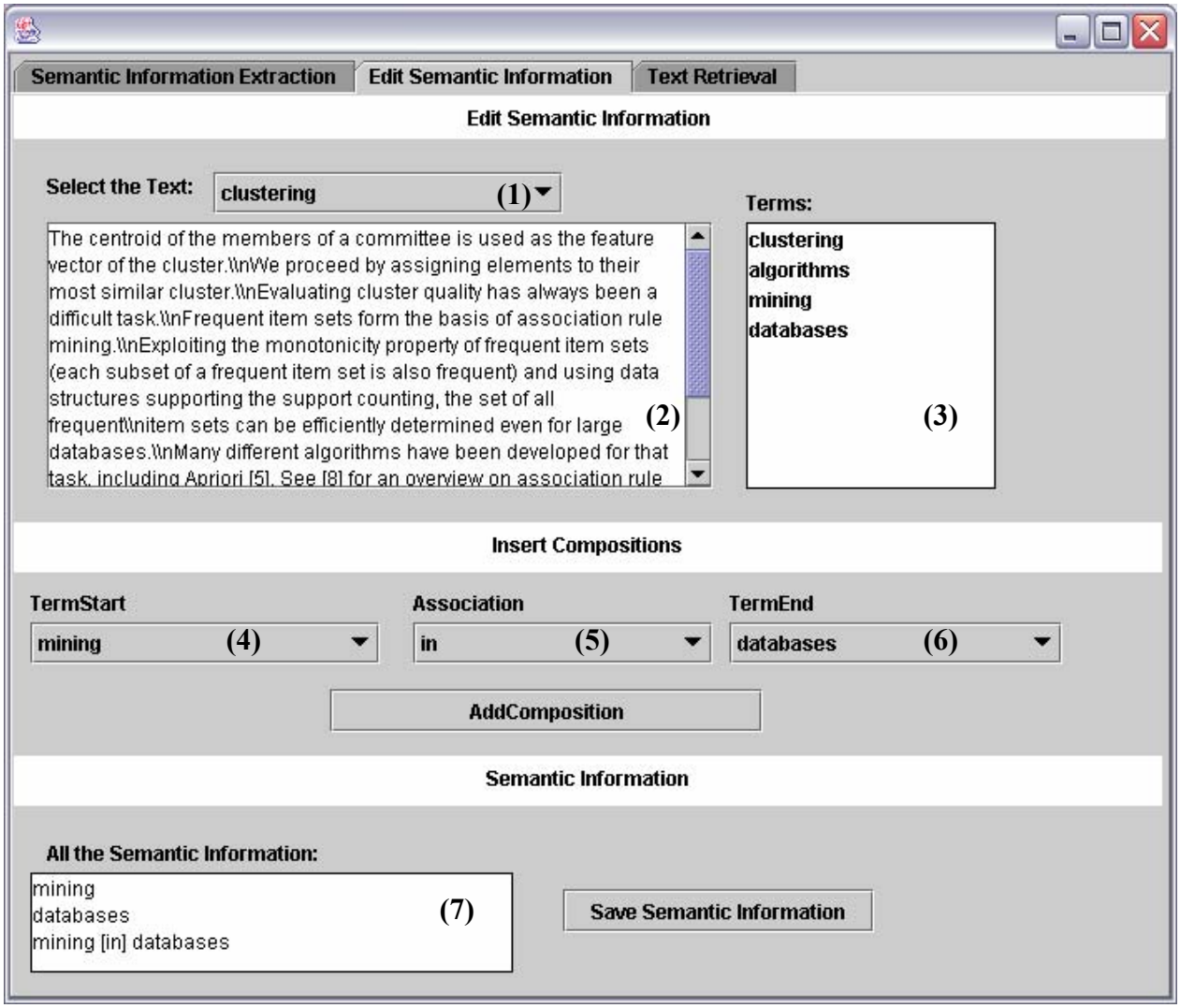


Figura 33: Acréscimo de Informação Semântica

Os termos iniciais e finais, que estão disponíveis, são os termos que estão relacionados ao texto em questão. Os termos utilizados em "*Association"* foram previamente cadastrados. Toda a informação semântica a ser acrescentada ao documento é exibida na área (7). O usuário deve, então, selecionar o botão "*Save Semantic Information*" para confirmar a inserção de informação semântica no Banco de Dados.

5.3.4 Armazenamento no Banco de Dados

A utilização de um Sistema Gerenciador de Banco de Dados permite que as informações obtidas dos textos sejam utilizadas diversas vezes e, possivelmente, por diversos usuários, portanto evita-se a extração das informações de texto constantemente. Por exemplo, se forem adicionados textos de um domínio já existente, não é necessário processar todos os documentos do domínio novamente. Os textos possuem informações imprecisas que podem ser úteis no processo de recuperação, permitindo melhor satisfazer os requisitos estabelecidos em uma consulta. A imprecisão nos dados armazenados pode ser melhor explorada através de mecanismos de busca nebulosa, o que permite ampliar o conjunto de resposta ao usuário, visando assim melhor atender a sua consulta. A busca nebulosa caracteriza-se no sistema pela utilização de percentuais de similaridade mínima e relevância dos termos envolvidos na consulta.

O Sistema Gerenciador de Banco de Dados utilizado é o Caché [Caché 2002] para o armazenamento e recuperação de documentos-texto, onde as informações extraídas dos processos anteriores são inseridas como objetos das classes representadas anteriormente (figura 25).

5.4 Módulo de Recuperação de Documentos

O Módulo de Recuperação de Documentos permite consultar o Banco de Dados de Informações Semânticas, recuperando os documentos armazenados. A informação semântica inserida sobre os documentos tem por objetivo permitir que sejam realizadas consultas que envolvam informações nebulosas ou exatas.

Ao elaborar uma consulta, o usuário pode informar quais os termos desejados e as associações entre os termos. Além disso, podem ser informados graus de similaridade mínima para cada termo estabelecido na consulta. Caso o usuário não estabeleça graus de similaridade para os termos, a consulta tem o comportamento de uma busca exata (similaridade mínima igual a 100%). Entretanto, caso informe os graus de similaridade, a consulta tem o comportamento de uma busca nebulosa. O usuário também pode informar o valor que representa a relevância de um elemento, que é um valor numérico no intervalo [0, 1], representando a importância da sua ocorrência no resultado da consulta.

As consultas são feitas através de uma interface, onde são especificadas as informações semânticas de documentos que estão armazenadas no banco de dados. Por exemplo, considere que o usuário queira recuperar documentos, que tenham os termos: "*mining e databases, ou clusterization e categorization",* com os respectivos valores de similaridade mínima e relevância

para os termos: mining (0.9, 1.0), databases (0.7, 0.8), clusterization (1.0, 0.9) e categorization (0.5, 0.8). Os valores de relevância são utilizados para calcular a importância do documento para a consulta, e com isso, classificar o resultado obtido. Melhores detalhes sobre os cálculos utilizados são apresentados no Capítulo 6.

Ainda é possível atender consultas onde composições são definidas entre os termos tal como: *"Recuperar os documentos contendo* ***mining in databases****",* que define como termo inicial - *mining*, termo final – *databases* e composição – *in.* O resultado da consulta incluirá todos os textos que apresentam tal composição.

A interface de Recuperação de Documentos é ilustrada na figura 34:

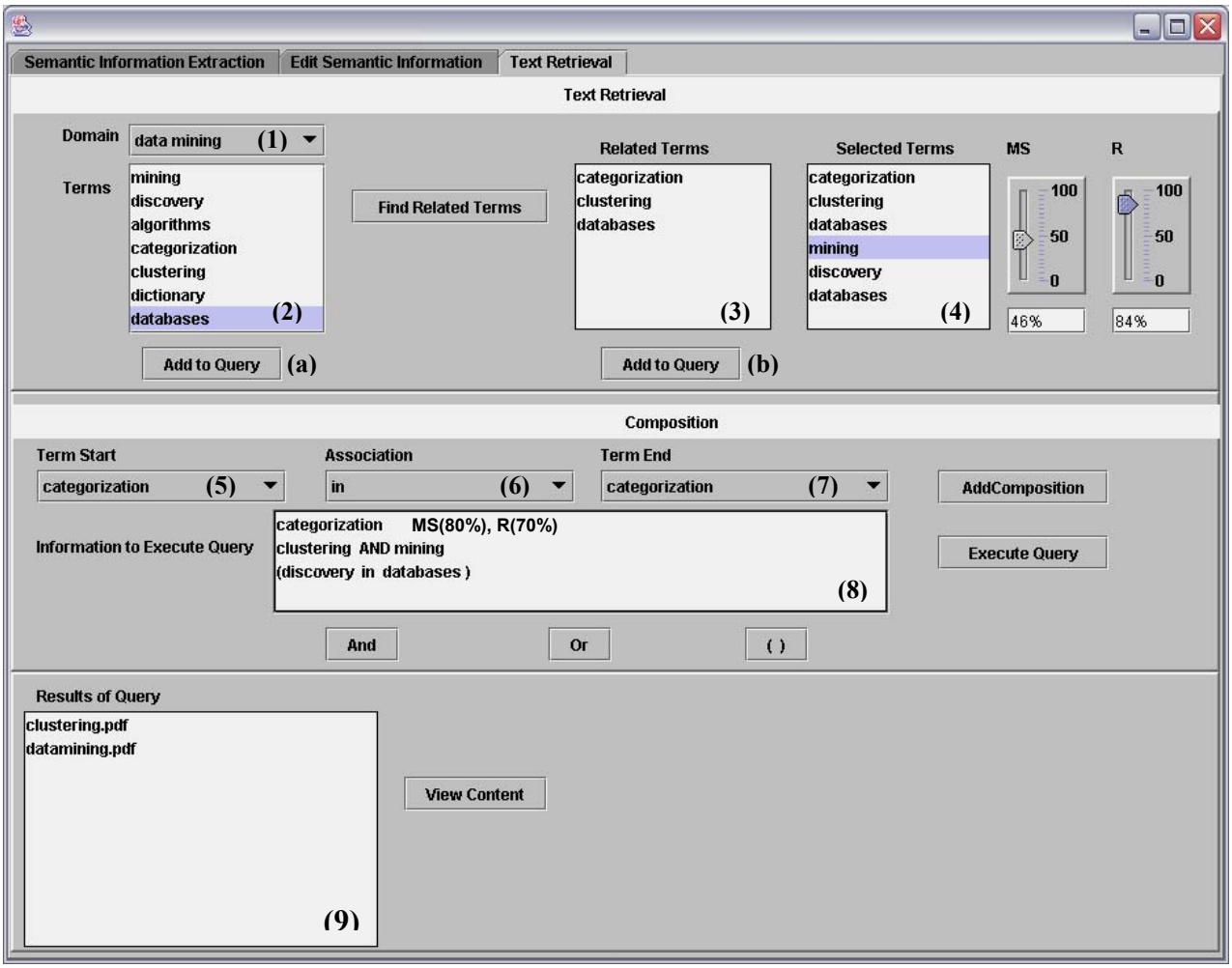


Figura 34: Módulo de Recuperação de Documentos

O primeiro passo é informar quais os termos que compõem a expressão de busca. Para isso, o usuário escolhe o domínio desejado em (1) e então, o sistema exibe os termos pertencentes ao domínio na área (2). Os termos que serão utilizados na expressão de consulta devem ser adicionados pelo usuário, através da seleção do termo na área (2) e pressionamento do

botão "*Add to Query"* (a). Para facilitar a escolha dos termos feita pelo usuário, o sistema oferece a opção de exibição de termos relacionados a um termo selecionado. Para que o usuário possa escolher termos mais sugestivos, ele deve selecionar o termo desejado na área (2) e pressionar o botão "*Find Related Terms*", fazendo com que o sistema exiba os termos relacionados na área (3). Caso o usuário desejar utilizar na consulta algum termo relacionado, basta selecioná-lo na área (3) e pressionar o botão "*Add to Query*" (b). Os termos estão previamente relacionados, usando-se o critério de 60% de similaridade mínima, adotado no momento da clusterização. Isto representa que, se um termo está relacionado a outro, é porque ele é no mínimo 60% similar. A lista de termos que será utilizada na consulta é apresentada na área (4).

A recuperação de documentos considera a presença de um determinado termo no documento. Entretanto, com a utilização da similaridade mínima, é possível ampliar a busca, especificando-se um valor mínimo de similaridade para que outro termo seja utilizado na recuperação. Para tanto, o usuário pode selecionar o valor de similaridade mínima (MS) para cada termo da lista e pode também selecionar um valor de relevância (R) para indicar a importância do termo selecionado para a consulta. O valor de relevância do termo é utilizado para a classificação dos documentos resultantes.

Como dito anteriormente, um termo está previamente relacionado a outro se apresentar uma similaridade mínima de 60%. A similaridade provoca uma ampliação da consulta, pois se estabelecida uma similaridade de 20% para um determinado termo, todos os termos que possuam uma similaridade mínima de 20%, em relação ao termo origem, são utilizados no processo de busca. Caso seja informado um valor de similaridade de 80%, o sistema utiliza os termos que tenham um grau de similaridade mínima de 80% em relação ao termo especificado.

O segundo passo é combinar termos, com o auxílio da interface, através de conectores AND/OR, ou composições (por exemplo, with, in, between, etc.) entre termos. Além disso, uma combinação pode estar relacionada com outras combinações. Por exemplo, considerando as expressões (clustering AND mining) OR (discovery IN databases), tem-se duas combinações que são relacionadas pelo conector OR.

As combinações de termos são denominadas grupos, que são classificados em grupos de termos e de composição de termos. Os grupos de termos são aqueles constituídos de termos relacionados pelos conectores AND/OR, e os grupos de composição de termos são aqueles constituídos por dois termos relacionados por uma associação.

Os grupos de termos são definidos através dos botões AND/OR da figura 34, enquanto que os grupos de composição de termos são definidos quando pressionado o botão "*AddComposition*". Para a composição, utilizando o termo "*in",* o usuário deve selecionar o

termo inicial (*"TermStart"*) na área (5), o termo de associação (*"Association")* na área (6) e o termo final (*"TermEnd")* na área (7) e acionar o botão "*AddComposition*". A expressão de busca é exibida na área (8). Quando o usuário aciona o botão "*Execute Query"*, o sistema retorna os documentos que atendam à expressão de busca, exibindo seus nomes na área (9).

A ordem em que os documentos são apresentados leva em consideração a relevância (R) especificada para cada termo da consulta, juntamente com os pesos dos termos para cada documento. Os cálculos utilizados para a classificação dos resultados são apresentados no Capítulo 6. Para visualizar o documento, o usuário seleciona seu nome e aciona o botão *"View Content"*, como pode ser visto na figura 35.

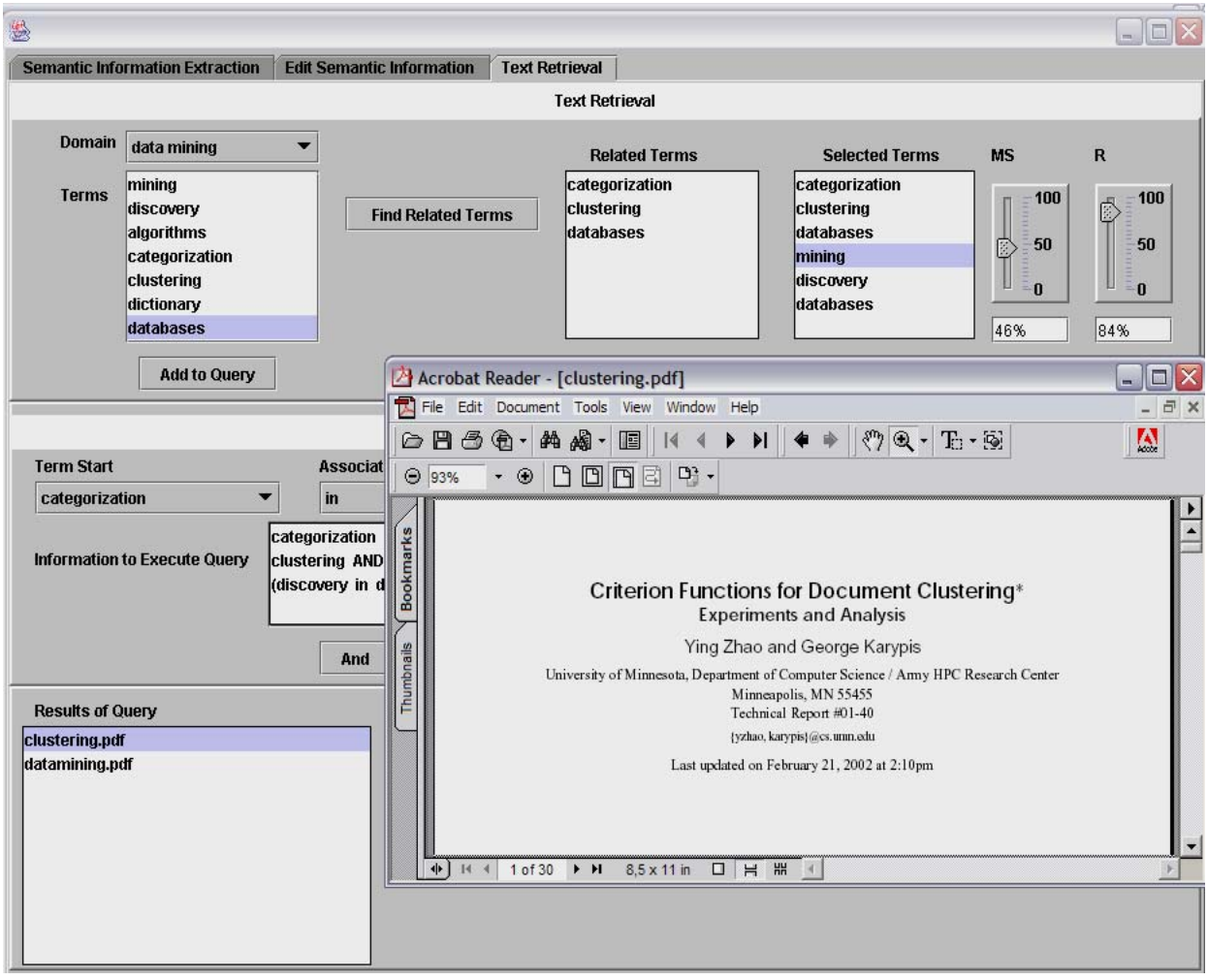


Figura 35: Visualização de Documentos

5.5 Adaptação de Enfoque de Documentos

Como uma proposta de um refinamento na recuperação de documentos, foi também investigada a viabilidade de inclusão de uma nova classe, chamada *Focus,* que poderia ser acrescentada à estrutura de classes apresentada na figura 25, conforme mostrado na figura 36. A

classe *Focus* seria utilizada para representar os vários enfoques apresentados no documento. Por exemplo, para o domínio *data mining,* diversos documentos compõem a coleção, entre eles, os documentos que apresentam ferramentas de data mining, outros que apresentam tarefas de data mining, outros que possuem enfoque teórico, enquanto outros tratam de aspectos práticos, etc.

A utilização do enfoque na recuperação de documentos possibilita um refinamento pelo mecanismo de busca, pois o usuário, além de escolher os termos e composições desejados, pode ainda especificar qual o enfoque desejado dos documentos procurados.

Uma forma de instanciar o enfoque, para cada documento da coleção, poderia ser através do usuário, que estabeleceria quais enfoques são abordados em determinado documento, com base no conteúdo deste.

Esse recurso ainda não foi incorporado ao sistema, pois, apesar de não exigir recursos especiais na implementação, requer modificação da interface do usuário.

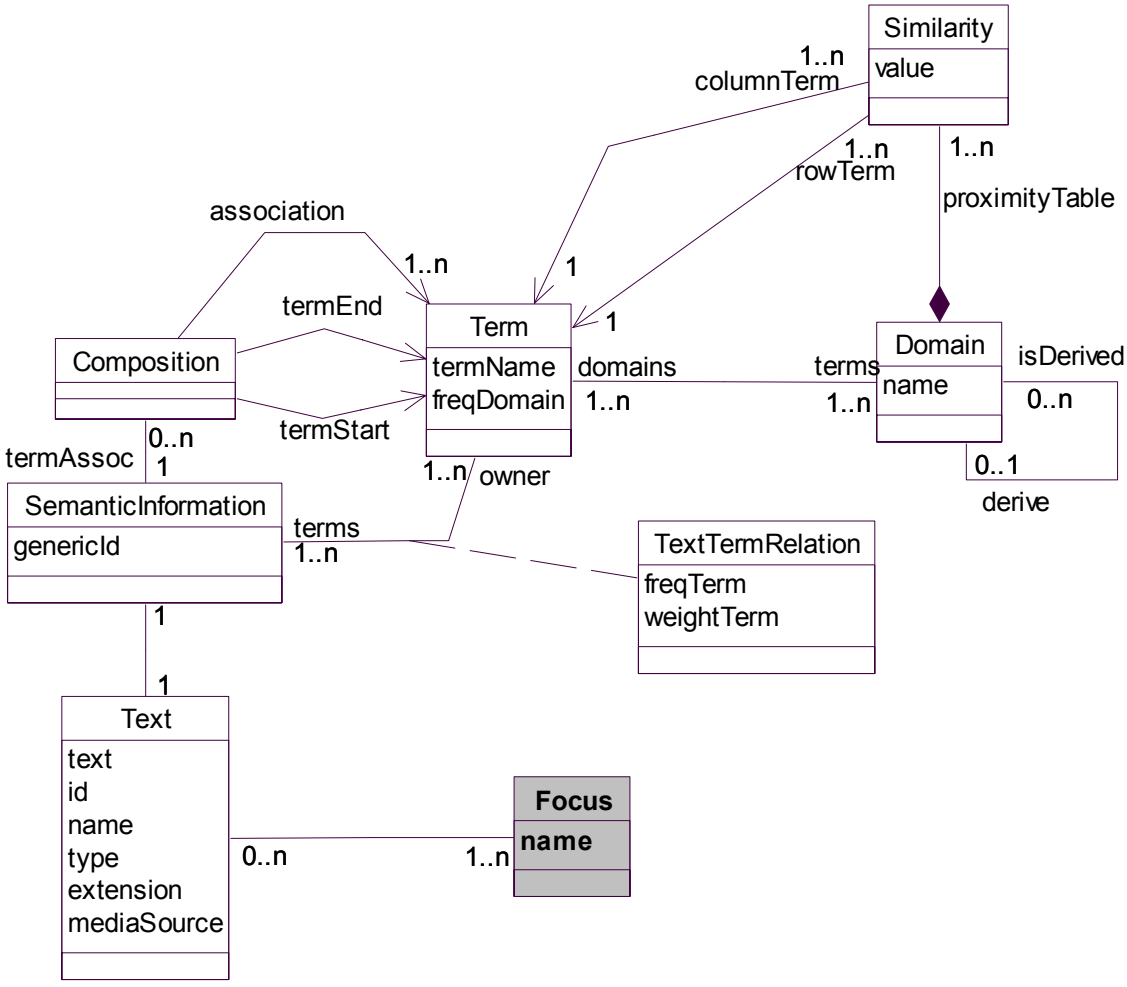


Figura 36: Estrutura de Classes com o acréscimo da classe Enfoque

5.6 Recursos Computacionais

Os recursos computacionais utilizados no desenvolvimento do protótipo do Sistema para Manipulação de Documentos baseada em Informação Semântica englobam:

- Linguagem de Programação: O desenvolvimento do Sistema para Manipulação de Documentos utilizou a linguagem de programação Java 2 SDK – Standard Edition, versão 1.4.1.
- Sistema Gerenciador de Banco de Dados: Para o armazenamento das informações, o Sistema Gerenciador de Banco de Dados utilizado foi o Caché 2002, fornecido pela InterSystems.

5.7 Considerações Finais

Este capítulo apresentou um Sistema para Manipulação de Documentos baseada em Informação Semântica, propondo uma forma de auxiliar a extração de informações relevantes de documentos, armazenando-as em um banco de dados.

Técnicas de mineração de textos foram utilizadas, dentre elas, algoritmos para o pré-processamento dos textos e clusterização dos termos encontrados. A clusterização resultou na criação de domínios e subdomínios de termos, enriquecendo as informações relativas aos textos e conseqüentemente, ampliando as características utilizadas na elaboração de buscas feitas pelo usuário.

No ambiente AMMO, o sistema pode ser incorporado para extração e recuperação de textos pertencentes às cenas das aplicações multimídia, possibilitando a recuperação dos próprios textos, das cenas ou da aplicação multimídia.

O sistema oferece um Módulo de Recuperação de Documentos onde as informações semânticas podem ser utilizadas para buscas exatas ou nebulosas. A classificação dos documentos recuperados é feita considerando-se o grau de relevância do documento para a expressão de busca, obtido pelos valores de relevância definidos pelo usuário e os pesos dos termos nos documentos. Os cálculos utilizados para a classificação dos resultados obtidos são descritos no próximo capítulo.

Capítulo 6

Estratégias de Busca e Classificação dos Documentos Recuperados

6.1 Introdução

Este capítulo apresenta as estratégias utilizadas na realização da busca e na classificação dos documentos. A estratégia de recuperação considera os graus de similaridade dos termos envolvidos nas consultas, assim como a relevância dos termos para a classificação dos documentos recuperados.

Para cada documento recuperado é calculado um grau de relevância, denotado por GR_{Di}, que representa o quanto o documento satisfaz aos requisitos estabelecidos na consulta. As diferentes fórmulas utilizadas para calcular o valor do grau de relevância do documento (GR_{Di}) levam em consideração as combinações, através de conectores AND/OR, entres os termos e composições definidos na consulta. Tais fórmulas foram baseadas no trabalho de Borges [Bor01] [Vie02], que utilizou conceitos da lógica nebulosa [Zad87] para o tratamento de consultas nebulosas em dados multimídia do ambiente AMMO. Neste trabalho, algumas idéias propostas por Borges foram adaptadas e reformuladas para atender às características das informações aqui tratadas.

A seguir são apresentadas as estratégias de busca e classificação utilizadas no presente projeto.

6.2 Estratégia de Busca

A primeira etapa da busca consiste em pré-selecionar os documentos armazenados com base nos termos envolvidos e a segunda etapa realiza o cálculo do grau de relevância (GR_{Di}) para cada documento pré-selecionado.

Os documentos pré-selecionados na primeira etapa contêm, no mínimo, um dos termos desejados pelo usuário, ou algum termo similar aos desejados, o qual satisfaz a similaridade mínima definida. Por exemplo, considere a tabela de similaridade reapresentada na figura 37:

		Termos de Indexação de Documentos						
		mining	discovery	algorithms	categorization	clusterization	dictionary	Databases
Termos de Indexação de Documentos	mining	–	0.66	0.22	0	0	0.14	0.26
	discovery	0.66	–	0.22	0	0	0.07	0.32
	algorithms	0.22	0.22	–	0.62	0.52	0.45	0.58
	categorization	0	0	0.62	–	**0.87**	0.23	0.68
	clusterization	0	0	0.52	**0.87**	–	0.15	0.68
	dictionary	0.14	0.07	0.45	0.23	0.15	–	0.18
	databases	0.26	0.32	0.58	0.68	0.68	0.18	–

Figura 37: Tabela de Similaridade

Suponha que o usuário deseja recuperar o termo *mining* com grau de similaridade mínima de 20%. Para esse termo *mining,* um conjunto de documentos é pré-selecionado, com base na tabela de similaridade, contendo o termo *mining* e também documentos contendo termos similares a *mining*, tais como *algorithms* (de *data mining)* e *databases (*com enfoque em *data mining)*, supondo que o grau de similaridade entre esses termos seja maior ou igual a 0.2. Essa pré-seleção ocorre para cada termo selecionado na consulta.

Na segunda etapa, o cálculo do GR_{Di} é efetuado para cada documento recuperado, com a finalidade de verificar qual o grau de relevância do documento recuperado para a expressão de busca e, então ordenar os documentos selecionados segundo o grau de relevância de cada um.

A expressão de busca é constituída por grupos, que podem ser classificados em:

- Grupo simples: grupo constituído por apenas um termo;
- Grupo de Termos: grupo constituído por dois ou mais termos relacionados entre si através de conectores AND/OR;
- Composições de termos: grupo constituído por termos conectados entre si através de uma composição.

Cada grupo G_j é definido formalmente como:

$$Gj = \begin{cases} Tj_1, \\ Tj_1 \; Assoc \; Tj_2, \text{ ou} \\ \qquad Tj_1 \; \theta \; Tj_2 \; \ldots \; \theta \; Tj_m \quad p/\; \theta = AND/OR \end{cases}$$

onde *Assoc* representa a associação entre dois termos de uma composição e, T_{ji}, i=1, ..., m os termos do grupo G_j.

Um exemplo de expressão de busca que contempla os grupos é apresentado a seguir:

(database AND (discovery OR mining)) AND (association WITH Apriori)

Para calcular o valor de relevância do documento (GR_{Di}) para a expressão de busca, são considerados os grupos G_j envolvidos na expressão de busca e os conectores utilizados entre eles (conectores AND/OR).

A seguir são apresentadas as fórmulas utilizadas para os cálculos dos graus de relevância dos termos e dos grupos.

6.2.3 Cálculo do grau de relevância do termo ($GR_{t_k,Di}$)

O valor do grau de relevância de cada termo t_k de um grupo em relação ao documento *i,* denotado por $GR_{t_k,Di}$ é obtido através da expressão:

$$GR_{t_k,D_i} = weight_{t_k,D_i} * relevance_{t_k} \quad (1)$$

onde $weight_{t_k,Di}$ é o peso do termo t_k no documento *Di,* e $relevance_{t_k}$ é o valor da relevância do termo t_k definido na expressão de busca. Esse valor de relevância é especificado pelo usuário.

Por exemplo, considere o termo *database* com valor de relevância de 0.9 e peso, no documento *D1,* de 0.5. Utilizando a expressão para calcular o grau de relevância, tem-se:

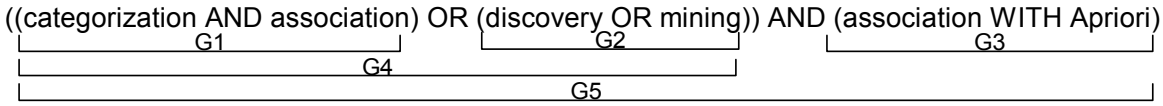
$$GR_{database,D_1} = 0.5 * 0.9 = 0.45$$

O grau de relevância do termo é utilizado para avaliar a importância do termo para o documento. A classificação dos documentos recuperados é feita através de cálculos que utilizam tal grau de relevância.

A seguir são apresentadas as fórmulas utilizadas nos cálculos dos graus de relevância dos grupos ($GR_{Di,Gj}$).

6.2.4 Cálculo do grau de relevância do grupo ($GR_{Di,Gj}$)

Os cálculos dos graus de relevância dos grupos para os documentos diferem conforme os conectores envolvidos na formação dos grupos. Por exemplo, considere a expressão de busca a seguir:



Suponha que os seguintes valores foram definidos para a relevância dos termos: categorization (0.9), association (0.8), discovery (0.8), mining (0.7) e apriori (0.8), e que estão sendo utilizados nos cálculos.

O cálculo dos graus de relevância dos grupos inicia-se percorrendo a expressão de busca, da esquerda para a direita respeitando a precedência definida pelos parênteses, obedecendo a seguinte classificação:

a. **Grupos compostos com apenas um termo**

Os grupos compostos por apenas um termo *t* possuem, como grau de relevância do grupo *Gj* para o documento *Di* ($GR_{Di,Gj}$), o grau de relevância do termo *t* no documento *Di*, como apresentado na expressão (2):

$$GR_{D_i,G_j} = GR_{t,D_i} \qquad (2)$$

b. **Grupos de termos com conectores AND (grupo G_1 do exemplo)**

O cálculo do valor $GR_{Di,Gj}$ para grupos compostos por dois ou mais termos conectados através do operador AND, é definido pela seguinte fórmula:

$$GR_{D_i,G_j} = \frac{\sum_{k=1}^{n} (GR_{t_k,D_i})}{\sum_{k=1}^{n} (relevance_{t_k})} \qquad (3)$$

onde $GR_{t_k,Di}$ é o grau de relevância do termo t_k no documento *Di* e $relevance_{t_k}$ é o valor de relevância definido pelo usuário para o termo t_k $kj=1,...,n$..

O cálculo é realizado através da média ponderada dos graus de relevância dos termos no documento, utilizando-se como fator de ponderação as relevâncias dos mesmos. Por exemplo, considerando o grupo G_1 da expressão utilizada como exemplo:

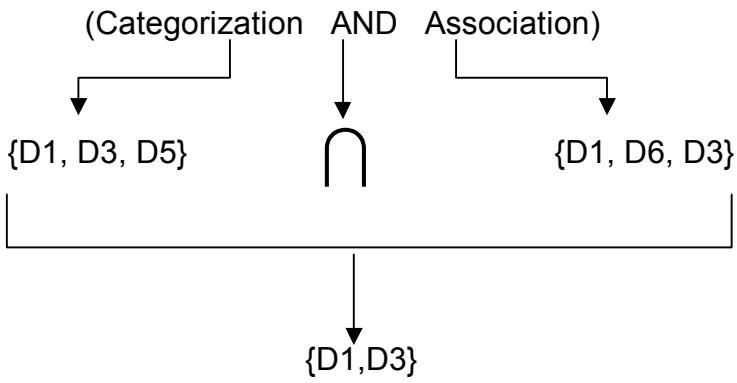


Figura 38: Exemplo com conector AND

Neste grupo, têm-se o termo *categorization* combinado, através do conector AND, com o termo *association*. Caso o usuário não especifique a similaridade mínima para os termos, a estratégia de busca considera apenas os termos que estão na expressão, isto é, são selecionados os documentos que tenham, necessariamente, os termos *categorization* e *association*.

Para o termo *categorization*, o mecanismo de busca pré-seleciona os documentos D1, D3 e D5. Para o termo *association*, o algoritmo de busca pré-seleciona os documentos D1, D6 e D3. Por se tratar de um conector AND, o resultado deve conter documentos que estejam em ambas as pré-seleções; portanto, o conjunto resultante de documentos é composto por D1 e D3.

Para ambos os documentos, devem ser calculados os graus de relevância do documento no grupo ($GR_{Di,Gj}$), seguindo a expressão (3).

Para tanto, os graus de relevância de cada termo para D1, foram calculados, através da expressão (1), obtendo-se os seguintes resultados:

$GR_{categorization,D1} = 0.72$ e $GR_{association,D1} = 0.72$

Aplicando-se a expressão (3), é obtido o resultado:

$$GR_{D1,G1} = \frac{(0.72 + 0.72)}{(0.9 + 0.8)} = 0.84$$

Para o documento D3, os graus de relevância considerados são:

$GR_{categorization,D3} = 0.54$ e $GR_{association,D3} = 0.40$, portanto, o grau de relevância do documento no grupo é:

$$GR_{D3,G1} = \frac{(0.54 + 0.40)}{(0.9 + 0.8)} = 0.55$$

***c.* Grupos de termos com conectores OR (grupo G_2 do exemplo)**

O cálculo do valor de $GR_{Di,Gj}$ para grupos compostos pelo termos *t1, t2, ..., tn* conectados através do operador OR, é definido pela fórmula:

$$GR_{Di,Gj} = \max(GR_{t_k,Di}), k = 1,..., n \quad (4)$$

Por exemplo, considerando o grupo G_2 da expressão utilizada como exemplo:

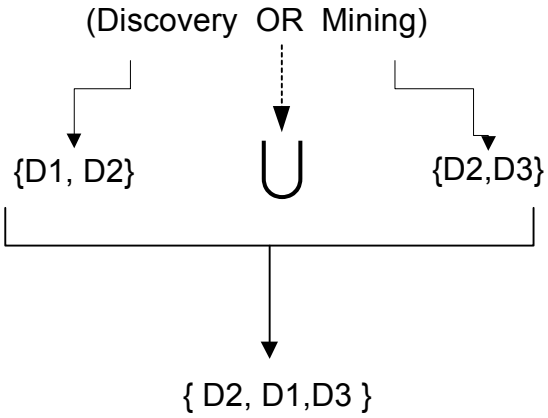


Figura 39: Exemplo com conector OR

O termo *discovery* é combinado, através do conector OR, com o termo *mining,* formando um grupo de termos.

Para o termo *discovery*, o mecanismo de busca pré-seleciona o documento D1 e D2, onde os graus de relevância do termo para os documentos são: $GR_{discovery,D1} = 0.56$ e $GR_{discovery,D2} = 0.72$.

Para o termo *mining*, o mecanismo pré-seleciona os documentos D2 e D3, com os seguintes graus de relevância do termo nos documentos: $GR_{mining,D2} = 0.56$ e $GR_{mining,D3} = 0.30$.

O cálculo do grau de relevância do documento no grupo ($GR_{Di,Gj}$) é feito utilizando-se a fórmula (4):

$$GR_{D2,G1} = \max\{0.72,0.56\} = 0.72$$

d. **Grupos com composições de termos**

Os documentos pertencentes a grupos com composições de termos possuem o valor de $GR_{Di,Gj}$ calculado através da fórmula apresentada na expressão:

$$GR_{Di,Gj} = \frac{GR_{t_k,Di} + GR_{t_{k+1},Di}}{relevance_{t_k} + relevance_{t_{k+1}}} \quad (5)$$

onde, t_k, t_{k+1} são os termos da composição, GR_{t_k} e $GR_{t_{k+1}}$ representam o grau de relevância dos termos *k* e *k+1* no documento *Di*.

Para exemplificar, considere o grupo G3:

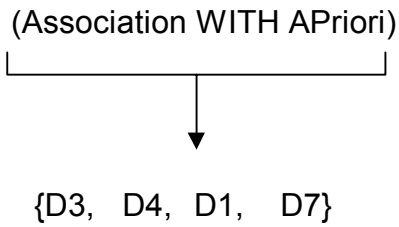


Figura 40: Exemplo com associação

Nesta expressão o termo *Association* é combinado, através de uma associação WITH, com o termo *APriori*. O mecanismo de busca pré-seleciona os documentos que satisfazem exatamente a expressão de busca, resultando nos documentos D1, D2, D3, D4.

Considerando o documento D1, têm-se os seguintes graus de relevância dos termos para o documento:

$GR_{association,D1} = 0.70$ e $GR_{apriori,D1} = 0.50$

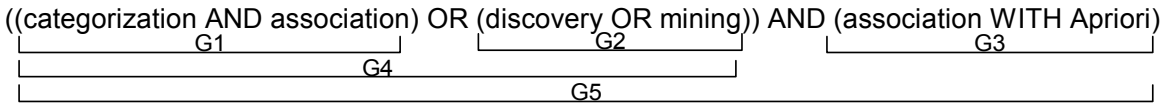
o cálculo do $GR_{Di,Gj}$ é feito da seguinte forma:

$$GR_{D1,G3} = \frac{0.70 + 0.50}{0.8 + 0.8} = 0.75$$

6.2.5 Cálculo do grau de relevância do Documento (GR_{Di})

O cálculo de GR_{Di} do documento é realizado após obtidos os graus de relevância do documento nos grupos de termos ou de composições de termos.

Considerando a expressão utilizada no exemplo e reapresentada a seguir:



pode ser observado que o grupo G1 e G2 estão conectados pelo conector OR.

É preciso, então, calcular o grau de relevância do documento em ambos os grupos. Para tanto, os cálculos para o grau de relevância dos documentos (GR_{Di}) diferem de acordo com a composição dos grupos.

a. Grupos conectados com OR (G1 OR G2**)**

Para os grupos conectados com OR, é calculado o grau de relevância do documento no grupo, segundo a expressão:

$$GR_{Di} = \max\{GR_{Di,Gj}, j = 1,\ldots,m\} \quad (6)$$

onde $max\{GR_{Di,Gj}\}$ é o maior valor obtido para o grau de relevância do documento *Di*, no grupo *Gj*.

Baseado no exemplo, tem-se G4 = (G1 OR G2), sendo que:

G1={D1,D3}, onde $GR_{D1,G1}$=0.84, $GR_{D3,G1}$=0.55 e

G2={D1,D2,D3}, onde $GR_{D1,G2}$=0.56, $GR_{D2,G2}$=0.72 e $GR_{D3,G2}$=0.30

O conjunto de documentos resultante de (G1 OR G2) é G4 = {D1,D2,D3}.

Utilizando-se a expressão (6), os valores do grau de relevância do documento são calculados, como mostrado a seguir para os documentos D1, D2 e D3:

$GR_{D1}=Max\{0.84,0.56\}=0.84$

$GR_{D2}=Max\{0.72\}=0.72$

$GR_{D3}=Max\{0.55,0.30\}=0.55$

b. Grupos conectados com AND (G4 AND G3**)**

Para os grupos conectados com AND, é calculado o grau de relevância do documento no grupo, segundo a expressão:

$$GR_{D_i} = \frac{\sum_{j=1}^{m} GR_{D_i,G_j}}{m} \quad (7)$$

onde *m* é a quantidade de grupos conectados por AND.

Baseado no exemplo, tem-se G5 = (G4 AND G3), sendo que:

G4={D1,D2,D3} e G3={D1,D4,D3,D7}. O conjunto de documentos resultantes da intersecção é G5={D1,D3}.

Sabendo-se que:

GR_{D1}=0.84 e GR_{D3}=0.55, aplicando-se a expressão (7) para calcular os graus de relevância dos documentos D1 e D3, são obtidos os valores:

$$GR_{D1} = \frac{0.75 + 0.84}{2} = 0.79$$

$$GR_{D3} = \frac{0.56 + 0.55}{2} = 0.55$$

Considerando os valores dos graus de relevância dos documentos resultantes, é feita a classificação destes; portanto, o documento D1 aparece em primeiro lugar na classificação e o documento D3, em segundo lugar.

A estratégia para a classificação dos documentos resultantes, com base na expressão de busca, considerou a relevância e o peso do termo dentro de cada documento. Nos exemplos citados, considerou-se que não tenha sido especificado um valor de similaridade mínima para os termos; entretanto, caso seja fornecida a similaridade mínima para os termos, a estratégia é novamente aplicada, considerando apenas os termos similares.

Neste caso, os documentos resultantes do processamento da expressão de busca que não utiliza similaridade mínima precedem os documentos obtidos com a similaridade mínima.

6.3 Considerações Finais

Este capítulo apresentou as estratégias de busca e classificação dos documentos armazenados no Banco de Dados de Informação Semântica.

As fórmulas utilizadas baseiam-se em trabalhos realizados para recuperação de mídias de Borges [Bor01], onde são utilizados os graus de similaridade das mídias com a expressão de busca.

Alguns testes foram realizados para a validação da classificação dos documentos recuperados pelo grau de relevância. Tais testes são apresentados no Apêndice B.

Capítulo 7

Conclusões

7.1 Considerações Iniciais

Neste trabalho foram apresentadas técnicas de mineração de textos como uma forma de extrair informação semântica de documentos, as quais foram utilizadas em um sistema para manipulação de documentos baseada em informação semântica. Para tanto, a abordagem adotada baseia-se na indexação de textos e clusterização de termos, através de cálculos de similaridades entre eles. A partir dos termos obtidos, o sistema oferece a possibilidade de incluir outras informações semânticas, que se torna possível pela utilização de composições entre os termos. Foi também apresentada a forma de recuperação de documentos com base na informação semântica envolvida no documento. O sistema disponibiliza opções de similaridade mínima, para que o resultado possa ser ampliado com termos relacionados entre si, e ainda relevância, onde é possível estabelecer a importância dos termos para a expressão de busca.

7.2 Contribuições e Resultados

As fórmulas elaboradas são de suma importância para a estratégia adotada na recuperação de documentos com base em informações semânticas automaticamente extraídas. Para testar a idéia, foi desenvolvido um protótipo de Sistema para Manipulação de Documentos baseada em Informação Semântica, o qual possibilita aos usuários extrair automaticamente as informações semânticas de documentos, assim como elaborar consultas com base nestas informações semânticas. A estratégia utilizada na recuperação leva em conta a relevância dos termos que pertencem à expressão de busca.

Foi proposto um conjunto de fórmulas para possibilitar a classificação dos documentos recuperados com base em sua relevância com relação à expressão de busca.

7.3 Trabalhos Futuros

Foram identificadas algumas funcionalidades que seriam importantes para aumentar os recursos utilizados na recuperação de documentos. São eles:

- Combinar o tratamento de documentos aqui proposto com a técnica de recuperação de áudio de Ribeiro [Rib99], para ampliar os recursos de recuperação de documentos de áudio.
- Incorporar expressões de busca negativas no sistema, utilizando-se o operador NOT.
- Incorporar um recurso para sumarização de documentos, para oferecer ao usuário um sumário do documento e, assim, facilitar a utilização do sistema.
- Incluir a classe *Focus* e seu tratamento para permitir recuperar documentos com base nos enfoques neles abordados.
- Permitir o processamento e recuperação de documentos para o idioma Português.
- Customizar o sistema para ser utilizado em um sistema de Biblioteca Digital.
- Automatizar o processo de composição de termos, utilizando-se técnicas de Processamento de Linguagem Natural.
- Permitir que o usuário defina os parâmetros a serem utilizados no tratamento dos documentos, tais como:
    - Escolher o site de busca desejado
    - Informar o tipo do arquivo a ser obtido
    - Definir o número máximo de palavras-chave para cada documento
    - Definir o limite (k) para o cálculo da similaridade entre os termos.

Referências Bibliográficas

[Agr93] Agrawal, R., Imielinski, T., Swami, A. *Mining Association Rules between Sets of Items in Large Databases.* Proc. Of Int. Conf. ACM SIGMOD, Washington, pp. 207-216, 1993.

[Agr94] Agrawal, R., Srikant, R. *Fast Algorithms for Mining Association Rules*. In Proc. Of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Setembro 1994.

[Bae99] Baeza-Yates, R., Ribeiro-Neto, B; *Modern Information Retrieval,* Addison-Wesley, 1999.

[Bor01] Borges, S. R. *Consultas Nebulosas Baseadas em Informações Semânticas em um Banco de Dados Multimídia*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, DC- UFSCar, São Carlos, Agosto de 2001.

[Bri92] Brill, E. *A Simple Rule-Based Part of Speech Tagger*. In Proc. of the Third Conference on Applied Natural Language Processing, pp. 152-155, Italy, 1992.

[Cac02] Intersystems; Caché 4.1.6 - http://www.intersystems.com.br/; Julho de 2002.

[Cal97] Callif, M.E., Mooney, R.J. *Relational learning of pattern-match rules for information extraction*. In ACL-97 Workshop in Natural Language Learning, 1997.

[Cal98] Callif, M.E. *Relational Learning Techniques for Natural Language Information Extraction*. PhD thesis, University of Texas at Austin, August 1998.

[Cop01] Copernic Summarizer, Copernic Technologies Inc. Documentação on-line disponível em http://www.copernic.com. Acesso em 15/12/2001.

[Cow96] Cowie, J., Lehnert, W. *Information Extraction.* Communications of the ACM. Vol. 39, Nº 1, January 1996.

[Dai94] Daille, B. *Combined approach for terminology extraction: lexical statistics and linguistic filtering*. Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives. Ph.D. thesis, University Paris 7, 1994.

[Dix97] Dixon, M. *An Overview of Document Mining Technology*. 1997. Disponível em: http://citeseer.nj.nec.com/dixon97overview.html. Acesso em 30/01/2002.

[Fay96] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. *Advances in Knowledge Discovery and Data Mining*. AAAIPress / The MIT Press. 1996.

[Fay96b] Fayyad, U.; Shapiro, G.P.; Smyth, P. *From Data Mining to Knowledge Discovery in Databases.* AI Magazine, 17(3):37-54, 1996.

[Fel95] Feldman, R.; Dagan, I. *Knowledge Discovery inTextual Databases (KDT)*. In First International Conference on Knowledge Discovery (KDD´95), Montreal, August, 1995.

[Fel96] Feldman, R.; Hirsh, H. *Mining associations in text in the presence of background knowledge*. In Proceedings of the Second International Conference on Knowledge Discovery and *Data Mining* (KDD-96), 343-346, 1996.

[Fel98] Feldman, R. et al. *Text mining at the term level*. In Proc. of the 2nd European Symposium on Principles of *Data Mining* and Knowledge Discovery (PKDD'98), Nantes, France, Sept 1998.

[Fel98a] Feldman, R. et al. *Knowledge Management: A Text Mining Approach*. Proc. Of the 2nd Int. Conf. on Practical Aspects of Knowledge Management (PAKM98), Basel, Switzerland, 29-30. Oct. 1998.

[Fig00] Figueiredo, J. M. *Desenvolvimento de um Ambiente para Autoria e Manipulação de Aplicações Multimídia na World Wide Web*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, DC-UFSCar, São Carlos, Brasil, Agosto de 2000.

[Fra00] Frank E. et al. "*KEA: Practical automatic keyphrase extraction*." Working Paper 00/5, Department of Computer Science, The University of Waikato. 2000.

[Gly97] Glymour, C., Madigan, D., Smyth, P., *Statistical Themes and Lessons for Data Mining*. In *Data Mining and Knowledge Discovery 1*. Kluwer Academic Publishers, pg 11-28. 1997.

[Goo03] Google 2003. Disponível em: http://www.google.com. Acesso em 12/01/2003.

[Han99] Hand, D. J., Statistics and Data Mining: Intersecting Disciplines. In ACM SIGKDD, Volume 1, Issue 1, pg 16. June, 1999.

[Hea96] M.A.Hearst; J.O.Pedersen. Reexamining the cluster hypothesis. In Proceedings of SIGIR 96, pp. 76-84, 1996.

[Kam01] Kamber, M.; Han, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[Kar00] Karanikas, H.; Tjortjis, C.; Theodoulidis, B. *An Approach to Text Mining using Information Extraction*, Knowledge Management Theory and Applications Workshop. http://www.sciences.univ-nantes.fr/irin/KMTA2000/ In conjunction with PKDD 2000, 12 September, 2000, Lyon, France.

[Kow97] Kowalski, Gerald. *Information Retrieval Systems: Theory and Implementation.* Boston: Kluwer Academic Publishers, 1997.

[Lan98] Landau, D., et al. *TextVis: An Integrated Visual Environment for Text Mining*. Proc. of

the 2nd European Symposium on Principles of *Data Mining* and Knowledge Discovery, PKDD1998.

[Lew96] Lewis, D. D.; Jones, K. S. *Natural Language Processing for Information Retrieval*. Communications of the ACM, 39(1):92-101, 1996.

[Meg00] Megaputer Intelligence, MicroSystems Co. 1997-2000. Documentação on-line. http://www.megaputer.com. Acesso em 01/04/2002.

[Nah00] Nahm, U. Y.; Mooney, R. J. *Using information extraction to aid the Discovery of prediction rules from text*. In Proceedings of the Sixth International Conference on Knowledge Discovery and *Data Mining* (KDD-2000) Workshop on *Text Mining*, pages 52-58, Boston, MA, August 2000.

[Nas01] Nasukawa, T.; Nagano, T. *Text Analysis and Knowledge Mining System.* IBM Systems Journal, vol. 40, n° 4, 2001.

[Net00] Neto, J. L., Santos, A. D., Kaestner, C. A. A., and Freitas, A. A. *Document clustering and text summarization*. In Proceedings, 4th Int. Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), 41-55. London: The Practical Application Company, 2000.

[Por80] Porter, M.F. *An algorithm for suffix stripping*, *Program*, 14(3) :130-137. 1980.

[Qui93] Quinlan, J., C4.5: *Programs for Machine Learning*. Morgan Kaufman, 1993.

[Raj97] Rajman, M.; Besançon, R.: *Text Mining: Natural Language Techniques and Text Mining Applications.* In: Proceedings of the seventh IFIP 2.6 Working Conference on Database Semantics (DS-7), Chapam & Hall IFIP Proceedings serie, 1997.

[Raj98] Rajman, M.; Besançon, R. *Text Mining - Knowledge Extraction from Unstructured Textual Data*. 6th Conference of International Federation of Classification Societies (IFCS-98), Rome, 1998.

[Rib99] Ribeiro, L.C.M. *Busca Baseada em Conteúdo em Documentos de Texto/Fala em um Banco de Dados MHEG-5.* Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, DC-UFSCar, São Carlos, 27/08/1999.

[Sal68] Salton, G. *Automatic Information Organization and Retrieval*. Computer Science Series, USA: McGraw-Hill, 1968.

[Sal83] Salton, G.; McGill, M. J. *Introdution to Modern Information Retrieval.* Computer Science Series, USA: McGraw-Hill, 1983.

[San03] Santos, F.G. Ferramenta MAE: Autoria Multimídia e Integração com Banco de Dados. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, DC- UFSCar, São Carlos, Fevereiro de 2003.

[Sen02] Seno, W. P. *Modelo e Base de Dados para Automatização do Planejamento e*

Execução de Cursos em Ambientes de Educação a Distância. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, DC-UFSCar, São Carlos, Brasil, Dezembro de 2001.

[Tan99] Tan, A.H. *Text Mining: The state of the art and the challenges*. In Proceedings of the. PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, pages 65-70, 1999.

[Vie02] Vieira, M.T.P.; Biajiz, M.; Borges, S.R.; Teixeira, E.C.; Santos, F.G.; Figueiredo, J.M. *Content-Based Fuzzy Search in a Multimedia Web Database*. In Intelligent Exploration of the Web. Springer-Verlag, "Studies in Fuzziness and Soft Computing" series, P.S.Szczepaniak, F.Segovia, J. Kacprzyk, and L.A.Zadeh, Eds, 2002 .

[Vie99] Vieira, M. T. P; Seno, W. P; Santos, M. T. P; Figueiredo, J. M. *Ambiente EAD-MAW para Educação a Distância*. Workshop Internacional sobre Educação Virtual – WISE'99, Fortaleza – Ceará, Brasil, 1999.

[Wge01] GNU WGET. Free Software Foundation, Inc. Disponível em: http://www.gnu.org/software/wget/. Acesso em 20/12/2001.

[Wiv97] Wives, L.K.; Castilho, J.M.V. *Indexação de Documentos Textuais*. Disponível em: http://www.inf.ufrgs.br/~wives/portugues/publicacoes.html. Acesso em 02/02/2002.

[Zad87] Zadeh L. A.; *Similarity Relations and Fuzzy Orderings*. In: Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh, Yager R.R., et al., eds., pp. 81-104, Wiley-Interscience Publication,1987.

[Zai01] Zaïane, O.R., Antonie, M.L., *Associating Terms with Text Categories*, Technical Report TR01-04, Department of Computing Science, University of Alberta, April 2001. Acesso em: 16/02/2002. Disponível em: http://www.cs.ualberta.ca/~zaiane/htmldocs/publication.html.

Bibliografia Consultada

Agrawal, R., Gehrke, J., Gunopulos, & D., Raghavan, P. Automatic Subspace Clustering of Dimensional Data for Data Mining Applications. Proc. Of the ACM SIGMOD Int. Conf. On Management of Data, Seattle, Washington, 1998.

Ahonen, H., Heinonen, O.,Klemettinen, M., Verkamo, A.I. Applying Data Mining Techniques in Text Analysis. Report C-1997-23, Department of Computer Science, University of Helsinki, 1997.

Ahonen, H.; Heinonen, O.; Klemettinen, M.; Verkamo, A. I. Applying data mining techniques for descriptive phrase extraction in digital document collections. In Advances in Digital Libraries (ADL'98), Santa Barbara, California, USA, April 1998. IEEE Computer Society Press.

Allen, J. Natural Language Understanding. The Benjamin/Cummings Publishing Company, Inc. Second Edition, USA, 1995.

Apte, C.; Damerau, F.; Weiss, S.M. Text mining with decision rules and decision trees. In Proc. 1998 Conf. Automated Learning and Discovery, workshop 6: Learning from text and the Web, pages 487-499, Pittsburgh, PA, 1998.

Brand, S. & Gerritsen, R., Neural Networks, DBMS Magazine - Data Mining Solutions Supplement. Julho de 1998.

Brown, E.W.; Callan, J.P.; Croft, W.B. Fast Incremental Indexing for Full-Text Information Retrieval. Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

Cooley, R., Srivastava, J., Mobasher, B., Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.

Cutting, D., Kupiec, J., Pedersen, J., Sibun, P. A practical part-of-speech tagger. In Proceedings of the 3th Conference on Applied Natural Language Processing, Trento, Italy, 1992.

Heckerman, D., Bayesian Networks for Knowledge Discovery, in Advances in Knowledge Discovery and Data Mining, edited by Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R., AAAI Press, USA. 1996.

Kodratoff. Knowledge Discovery in Texts: A Definition and Applications, Proc. Of the 11th International Symposium on Foundations of Intelligent Systems (ISMIS-99), 1999.

Kufrin, R.. Decision trees on parallel processors. In Proceedings of IJCAI-95 Workshop on Parallel Processing for Artificial Intelligence, Montreal, Canada, August 1995.

Loh, S.; Wives, L.K.; Oliveira, J.P.M. Concept-Bases Knowledge Discovery to Texts Extracted from the Web. SIGKDD Explorations, ACM SIGKDD, July, 2000. Volume 2, Issue 1, 29-38.

Scarinci, R.G.; Oliveira, J.P.M. SES Sistema de Extração Semântica de Informações. Simpósio Brasileiro de Banco de Dados, 1997.

Silva, D. R. Uma Ferramenta para Descoberta de Conhecimento com Suporte de Data Warehousing e sua Aplicação para Acompanhamento do Aluno em Educação a Distância. Dissertação de Mestrado. Departamento de Computação, Universidade Federal de São Carlos, São Carlos, Brasil, 2002,108p.

Sousa, M.S.; Mattoso, M.L.Q.; Ebecken, N.F.F. Data Mining: a database perspective. WITPress – 1998.

Srikant, R., & Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT). Avignon, France (1996).

Witten, I.H. et al. Text Mining: A new frontier for lossless compression. In Proc. Data Compression Conference, Snowbird, Utah, 1999.

Wives, L.K. Técnicas de descoberta de conhecimento em textos aplicadas à inteligência competitiva. Exame de Qualificação. Porto Alegre: PPGC/UFRGS, 2001. Disponível em: http://www.inf.ufrgs.br/~wives/portugues/publicacoes.html. Acesso em 02/02/2002.

Wives, L.K.; Loh, S. Tecnologias de descoberta de conhecimento em informações textuais (ênfase em agrupamento de informações). In: Oficina De Inteligência Artificial (OIA), III, 1999, Pelotas, RS. P 28-48. (TUTORIAL)

Apêndice A

Algoritmos Utilizados

1. Introdução

Neste apêndice são descritos os principais algoritmos utilizados para o tratamento dos documentos, assim como, os algoritmos para recuperação destes.

A seção 2 mostra os algoritmos utilizados no módulo de Tratamento de Documentos, enquanto a seção 3 mostra os algoritmos utilizados no módulo de Recuperação de Documentos.

2. Algoritmos do Módulo de Tratamento de Documentos

Nesta seção são apresentados os algoritmos dos métodos pertencentes ao Módulo de Tratamento de Documentos.

A primeira etapa do Módulo de Tratamento de Documentos consiste na Obtenção dos Documentos, a qual pode ser feita na *World Wide Web* ou, até mesmo, em mídias existentes. As etapas seguintes são responsáveis pelo pré-processamento dos documentos, clusterização destes e armazenamento dos documentos e seus termos no Banco de Dados.

O primeiro algoritmo, representado na figura 41, realiza o processamento de cada documento, extraindo palavras-chave de cada um deles.

```
Para cada documento selecionado faça
     Converte(documento) //conversão para o formato texto
     LimpaTexto(documento) //retira caracteres inválidos, números e converte
                           //para minúsculo
     Para cada termo do documento
          Se(tamanho(termo)>2) e (tamanho(termo)<20)
               Se(Não(termo.isStopWord( )) //verifica se o termo não está na
                                           //stoplist
                    novoDocumento = novoDocumento + termo
               Fim-se
          Fim-se
      Fim-Para
     ApplyFilter(novoDocumento)// utiliza o PorterStemmer para retirar sufixos
     ColecaoDoc.add(novoDocumento)//adiciona o documento a uma coleção de
```

```
                                    //documentos
Fim-Para
tfidf.addDocSet(colecaoDoc) // adiciona a coleção de documentos a um objeto
                            //da classe tfidf que calcula os pesos para cada
                            //termo em cada documento
Para cada documento de tfidf
  VetorTermos.add(tfidf.getVector(documento).topN(20)) //obtem os 20 termos
                                                       //de pesos maiores
  GravaBD() // método para armazenar no BD as palavras-chave e o documento
            //que as contém
Fim-Para
```

Figura 41: Algoritmo para extrair palavras-chave dos documentos

O segundo algoritmo apresentado a seguir é utilizado para criar um dicionário de termos encontrados nos documentos, sem que haja termos repetidos. Tal dicionário será utilizado mais adiante para a criação da matriz de similaridade.

```
Para cada documento da colecaoDoc
     Para cada termo do documento
          Booleano contem = dicionário.contains(termo) //verifica se o termo
                                                       //já existe no
                                                       //dicionário
          Se(Não(contem)) // termo não está no dicionário
               dicionário.add(termo)
           Fim_Se
     Fim_Para
Fim_Para
```

Figura 42: Algoritmo para criar um dicionário de termos

O algoritmo apresentado na figura 43 é utilizado para criar a matriz de similaridade, onde o valor de similaridade entre cada par de termos é calculado.

```
VetorSimilaridade = new Vector()
ObjetoSimilar = new ClassSimilaridade()
Para cada termo do dicionário
     Valor = Calcula_similaridade(termo, termo+1) //calcula a similaridade
                                                  //entre os pares de termos
     ObjetoSimilar.termo1=termo
     ObjetoSimilar.termo2=termo+1
     ObjetoSimilar.valorSim=Valor
     VetorSimilaridade.add(objetoSimilar)
Fim-Para
```

Figura 43: Algoritmo para criar a matriz de similaridade

O método Calcula_similaridade é responsável por calcular a similaridade entre os termos passados como parâmetros e retornar o valor de similaridade entre os termos. Um vetor de objetos da classe "*ClassSimilaridade*" é criado armazenando os pares de termos e os valores de similaridade.

O próximo algoritmo é utilizado para criar as classes tesauros, ou seja, a partir de um limite estabelecido para o valor de similaridade (k), as classes contendo termos similares são criadas (clusterização), como apresentado na figura 44.

```
K = 0.60 //adoção do valor 0.6 para k, poderia ser fornecido pelo usuário
Para cada elemento de vetorSimilaridade
     Se (objetoSimilar.valorSim > K)
          grava_BD_associacoes() // armazena os objetos persistentes no BD.
     Fim_Se
Fim_Para
```

Figura 44: Algoritmo utilizado para a clusterização

3. Algoritmos do Módulo de Recuperação de Documentos

Adotando-se a estratégia proposta por Borges [Bor01], é criado um vetor para armazenar os grupos da expressão de busca e os conectores entre eles. Tal vetor é denominado *expressionVector*.

Uma etapa denominada de Pré-Seleção realiza uma pré-seleção dos documentos armazenados no BD de aplicações.

Essa pré-seleção busca os termos solicitados na expressão, ou termos similares (satisfazendo os graus de similaridade mínima estabelecido para os termos solicitados), combinados da mesma forma que a definida na expressão de busca. O algoritmo apresentado na figura 45 mostra como esta etapa foi realizada.

```
Para cada elemento do vetor expressionVector Faça
     Se o elemento é um grupo Então
          Para cada termo do grupo Faça
               Recuperar os termos similares ao termo do grupo
               //método getSimilarTerms
               Para cada termo similar recuperado Faça
                    Recuperar os documentos que o contém
                    Para cada documento recuperado Faça
                         Adicionar o documento a uma coleção de documentos do
                                   termo similar
                    Fim-Para
               Fim-Para
               Armazenar os objetos da coleção de documentos do termo similar
                         à coleção de documentos do termo selecionado
          Fim-Para
          Combinar as coleções de documentos de acordo com o conector do
                    grupo
          Adicionar a coleção resultante no vetor DocumentVector na mesma
                    posição que do grupo
     Senão
          Adicionar o conector que está no vetor expressionVector ao vetor de
                    documents
     Fim-Se
Fim-Para
Combinar as coleções de documents do vetor de documents de acordo com os
conectores entre as coleções
```

Figura 45: Algoritmo de Pré-Seleção

O algoritmo deve ser entendido nas seguintes etapas:

1ª.Etapa: Os termos similares para cada termo do grupo são obtidos. O programa deve percorrer o vetor *expressionVector* e armazenar os termos similares em uma coleção de termos similares aos termos do grupo. Os termos similares são aqueles que atendem aos graus de similaridade definidos pelo usuário. Os valores de similaridade estão armazenados nas tabelas de proximidade pertencentes à classe *Domain.*

2ª.Etapa: Em seguida, os documentos que contem os termos similares devem ser recuperados, para isso, para cada termo similar encontrado, uma coleção de documentos que contenha o termo é obtida. Para cada grupo são recuperadas várias coleções de documentos referentes a cada termo do grupo.

3ª.Etapa: A combinação das coleções de documentos recuperados é feita para cada grupo de acordo com os conectores. Dessa forma, se os termos do grupo são combinados pelo conector AND, realiza-se a intersecção dos documentos das coleções, ou seja, obtêm-se os documentos que contêm todos os termos do grupo. Se for o conector OR, realiza-se a união dos documentos contidos nas coleções, obtendo-se os documentos que possuem pelo menos um dos termos do

grupo. Além disso, quando o grupo é uma composição de dois termos têm-se duas coleções de documentos, sendo uma recuperada para cada termo, combinadas pelo conector AND, pois os dois termos devem estar presentes nesses documentos.

4ª.Etapa: O último passo consiste em percorrer o vetor de documentos obtido (*DocumentVector*) , combinando as coleções de documentos de acordo com o conector existente entre elas. Assim, se for o conector AND, é realizada uma interseção dos documentos dessas coleções, e se for o conector OR, uma união. O resultado é combinado com a próxima coleção obedecendo ao conector.

Após as etapas apresentadas, os documentos já foram pré-selecionados pelo sistema, para os quais são calculados seus valores de similaridade com a expressão de busca.

Apêndice B

Testes Realizados

1. Introdução

Neste apêndice são descritas as etapas utilizadas para testar e avaliar a capacidade de recuperação do sistema. A realização dos testes foi feita em duas etapas: a primeira etapa é referente ao Módulo de Tratamento de Documentos, e a segunda etapa é referente ao Módulo de Recuperação de Documentos.

2. Primeira Etapa: Tratamento de Documentos

Utilizando-se o módulo de Obtenção de Documentos, foram obtidos 11 documentos texto de domínio público sobre "clusterização". Tais documentos foram submetidos ao módulo de pré-processamento, onde o primeiro passo foi a extração das palavras da coleção de documentos. Em seguida, as palavras que estão incluídas no dicionário auxiliar Stoplist foram ignoradas e os sufixos definidos no dicionário auxiliar Suffixlist foram retirados das palavras que os contém. Para cada termo foi calculado o peso, que representa sua importância para o documento e para a coleção. Com base nos pesos calculados, as palavras com maiores pesos foram atribuídas como palavras-chave, ou termos de indexação, para o documento.

As palavras comuns pré-determinadas foram armazenadas em uma lista denominada Stoplist, lembrando que palavras comuns são aquelas que não são consideradas boas identificadoras do conteúdo de um documento por terem a freqüência de ocorrência demasiadamente grande. A tabela 1 exibe algumas palavras contidas na Stoplist.

about	be	everybody	More
above	because	everyone	Most
across	before	everything	Much
afore	beside	except	Must
aforesaid	best	following	Mustn'
after	both	hard	t
again	but	has	My
against	by	hasn't	Myself
ago	can	her	Near
all	couldn't	here	Neath
almost	despite	herself	Need
alone	did"	himself	Of
along	doesn't	his	Off
always	during	immediately	Often
among	either	inside	On
			Once

Tabela 1: Parte da Stoplist

A tabela 2 mostra alguns dos sufixos contidos na Suffixlist que são retirados das palavras:

eed	at	tion
ed	ate	enci
ing	ble	izer
ful	iz	entli
iveness	ize	ization
fulness	ational	alism

Tabela 2: Parte da Suffixlist

Um exemplo de palavras-chave para um determinado documento pode ser visto na tabela 3, que contém os termos atribuídos ao documento, juntamente com os pesos de cada termo.

Beil02frequent	
Palavras-chave	Pesos
fscore	0.80
dataset	0.54
partit	0.54
hierarch	0.53
agglom	0.79
centroid	0.44

Tabela 3: Documento obtido com palavras-chave

Clusterização

As classes de termos similares foram criadas por um processo automático (clusterização), com base nos documentos obtidos e nas palavras-chave de cada documento. A tabela 4 exibe uma parte da matriz binária criada para o limite estabelecido de similaridade de k=0.6 (60%), ou

seja, cada componente de uma classe possui no mínimo 60% de similaridade em relação aos outros componentes.

	agglom	aggreg	bisect	centroid	Classif	concept	...
agglom	-	-	1	1	1	-	
aggreg	-	-	-	-	-	1	
bisect	1	-	-	-	-	-	
centroid	1	-	-	-	-	-	
classif	1	-	-	-	-	-	
concept	-	1	-	-	-	-	
customiz	-	-	-	-	-	-	
databas	-	-	-	-	-	-	
dataset	1	-	-	-	-	-	
...							

Tabela 4: Parte da matriz binária

A tabela 5 mostra alguns exemplos de classes de termos similares geradas automaticamente pelo sistema e que foram armazenados no banco de dados de informação semântica. Na classe 1, por exemplo, pode-se constatar que as palavras-chave “agglom”, “bisect”, “centroid”, “classif”, “dataset” e “hierarch” se agrupam como termos similares.

Classe 1	Agglom,bisect,centroid,classif,dataset,hierarch
Classe 2	Aggreg,concept
Classe 3	Bisect,entropi
Classe 4	Vector,bisect

Tabela 5: Classes de termos similares

3. Segunda Etapa: Recuperação de Documentos

O objetivo desta etapa foi verificar e validar as fórmulas utilizadas nos cálculos dos graus de relevância dos documentos em relação às consultas definidas. Para isso, as consultas foram realizadas em níveis de complexidade, como definido a seguir:

a. Consultas envolvendo um único termo;
b. Consultas envolvendo dois ou mais termos em um grupo;
c. Consultas envolvendo grupos de termos.

A seguir, são apresentadas algumas consultas realizadas nos testes. Os valores definidos para os graus de similaridade e relevância aparecem dentro de colchetes, respectivamente.

3.1 Consultas envolvendo um único termo

a. "Hierarch"[1,1]

Considere que o usuário deseja recuperar documentos que contenham o termo "Hierarch" com graus de similaridade e relevância iguais 1 (busca exata). A tabela 6 apresenta os nomes dos documentos recuperados para a consulta, classificados pelo grau de relevância.

A consulta apresentada possui apenas um termo e, portanto, representa um grupo composto com apenas um termo; por isso, os valores obtidos para o grau de relevância do documento foram calculados através da expressão (2), reapresentada a seguir:

$$GR_{Di,Gj} = GR_{t,Di} \qquad (2)$$

sendo que $GR_{t,Di}$ é expressa por:

$$GR_{t_k,Di} = weight_{t_k,Di} * relevance_{t_k} \qquad (1)$$

Documento	weight	relevance	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf	0.73	1	0.73
2) Beil02frequent.pdf	0.53	1	0.53
3) Clustering.pdf	0.49	1	0.49

Tabela 6: Consulta "hierarch"

b. "Hierarch" [0.5, 1]

Considerando que o usuário refaça a consulta utilizando o grau de similaridade de "hierarch" 0.5, a consulta permite recuperar documentos que contenham um termo similar ao termo "hierarch". O sistema efetua a busca exata, como apresentado anteriormente e depois, realiza a busca nebulosa, ou seja, utiliza termos que sejam no mínimo 50% similares a "hierarch". O resultado, apresentado na tabela 7, exibe primeiramente os documentos referentes à primeira busca e, em seguida, os documentos referentes à busca com termos similares.

Documento	Termo similar	weight	Relevance	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf		0.73	1	0.73
2) Beil02frequent.pdf		0.53	1	0.53
3) Clustering.pdf		0.49	1	0.49
4) Frank.pdf	"bisect"	0.50	1	0.50
5) Zao02.pdf	"bisect"	0.49	1	0.49
6) Hotho01.pdf	"classif"	0.45	1	0.45
7) Steinb.pdf	"agglom"	0.44	1	0.44

Tabela 7: Consulta "hierarch" [0.5,1]

c. "Hierarch" [0.5, 0.6]

O usuário pode ainda especificar um grau de relevância para o termo "hierarch" de 60%, obtendo os documentos apresentados na tabela 8.

Documento	Termo similar	weight	Relevance	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf		0.73	0.6	0.43
2) Beil02frequent.pdf		0.53	0.6	0.31
3) Clustering.pdf		0.49	0.6	0.29
4) Frank.pdf	"bisect"	0.50	0.6	0.30
5) Zao02.pdf	"bisect"	0.49	0.6	0.29
6) Hotho01.pdf	"classif"	0.45	0.6	0.27
7) Steinb.pdf	"agglom"	0.44	0.6	0.26

Tabela 8: Consulta "hierarch" [0.5,0.6]

Note que a classificação dos documentos obtidos não se altera pois o valor de relevância utilizado para os cálculos permanece o mesmo para os todos documentos.

3.2 Consultas envolvendo dois ou mais termos em um grupo

As consultas que envolvem termos em um grupo possuem os mesmos valores de GR_{Di} e $GR_{Di,Gj}$, sendo que o grau de relevância do documento dentro de um grupo ($GR_{Di,Gj}$) utiliza a fórmula (3), para grupos de termos com conectores AND, a fórmula (4), para grupos de termos conectados com OR e a fórmula (5), para a composição de termos.

a. ("hierarch" [1,1] AND "classif" [1,1])

Considere que o usuário deseja recuperar documentos que contenham "hierarch" e "classif" com valores de similaridade e relevância de 1 (busca exata). Após os graus de

relevância dos documentos terem sido calculados com a utilização da fórmula (3), foram recuperados os documentos apresentados na tabela 8.

$$GR_{Di,Gj} = \frac{\sum_{k=1}^{n} (GR_{t_k,Di})}{\sum_{k=1}^{n} (relevance_{t_k})} \qquad (3)$$

Documento	**Weight hierarch**	**Relevance hierarch**	**Weight classif**	**Relevance Classif**	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf	0.73	1	0.44	1	=(0.73+0.44)/2 =0.58
2) Clustering.pdf	0.49	1	0.55	1	=(0.49+0.55)/2 =0.52

Tabela 9: Consulta ("hierarch" AND "classif")

Se nessa consulta o usuário tivesse indicado que o termo "hierarch" é menos importante do que o termo "classif", diminuindo sua relevância para 0.30, a classificação dos documentos seria alterada, como apresentado na tabela 10.

Documento	**Weight hierarch**	**Relevance hierarch**	**Weight classif**	**Relevance Classif**	$GR_{Di} = GR_{Di,Gj}$
1) Clustering.pdf	0.49	0.3	0.55	1	=(0.14+0.55)/1.3 =0.53
2) Vscluster.pdf	0.73	0.3	0.44	1	=(0.21+0.44)/1.3 =0.50

Tabela 10: Consulta ("hierarch" [1,0.3] AND "classif" [1,1])

Suponha que o usuário refaça a consulta anterior diminuindo o grau de similaridade do termo "classif" para 0.5. A tabela 11 apresenta os nomes dos documentos que foram recuperados na presente consulta, utilizando termos com 50% de similaridade.

Documento	**Termo similar**	$GR_{Di} = GR_{Di,Gj}$
1) Clustering.pdf		0.53
2) Vscluster.pdf		0.50
3) Beil02frequent.pdf	"agglom"	0.73

Tabela 11: Consulta ("hierarch"[1,0.3] AND "classif" [0.5,1])

Note que além dos documentos recuperados pela consulta anterior, o documento "Beil02frequent.pdf" foi recuperado com valor de 0.73 para o grau de relevância. Esse valor foi obtido através da média ponderada dos graus de relevância dos termos.

Nesta consulta, três documentos que possuem termos similares a "classif" não foram recuperados porque não foi satisfeito o conector do grupo (AND).

b. ("hierarch" [1,1] OR "classif" [1,1])

Suponha que o usuário deseja recuperar documentos que contenham o termo "hierarch" ou documentos com o termo "classif". Considerando como uma busca exata, todos documentos recuperados, com os valores para os graus de relevância calculados através da fórmula (4), são apresentados na tabela 12.

$$GR_{Di,Gj} = \max(GR_{t_k,Di}), k = 1,\ldots, n \quad (4)$$

Documento	**Weight hierarch**	**Relevance hierarch**	**Weight classif**	**Relevance Classif**	GR_{Di}
1) Vscluster.pdf	0.73	1	0.44	1	=Max{0.73,0.44} =0.73
2) Clustering.pdf	0.49	1	0.55	1	=Max{0.49,0.55} =0.55
3) Beil02frequent.pdf	0.53	1	-	-	=Max{0.53,0} =0.53
4) Steinb.pdf	-	-	0.51	1	=Max{0,0.51} =0.51
5) Hotho01.pdf	-	-	0.45	1	=Max{0,0.45} =0.45

Tabela 12: Consulta ("hierarch" OR "classif")

Suponha que o usuário refaça a consulta anterior estabelecendo o grau de similaridade do termo "classif" 0.5 . Essa consulta permite recuperar documentos que contenham o termo "hierarch" ou algum termo similar a "classif". Os documentos que foram classificados são apresentados na tabela 13.

Documento	**Termo Similar**	GR_{Di}
1) Vscluster.pdf		0.73
2) Clustering.pdf		0.55
3) Beil02frequent.pdf		0.53
4) Steinb.pdf		0.51
5) Hotho01.pdf		0.45
6) Slonim.pdf	"keyphras"	0.99

Tabela 13: Consulta ("hierarch" [1,1] OR "classif"[0.5,1])

c. (“hierarch” WITH “classif”)

Suponha que o usuário deseja recuperar documentos que contenham a semântica “hierarch WITH classif” (composição de termos). Após o grau de relevância ter sido calculado através da fórmula (5), os documentos (tabela 14) foram recuperados. Esses documentos satisfizeram exatamente o predicado da consulta.

$$GR_{Di,Gj} = \frac{GR_{t_k,Di} + GR_{t_{k+1},Di}}{relevance_{t_k} + relevance_{t_{k+1}}} \quad (5)$$

Documento	Weight hierarch	Relevance hierarch	Weight classif	Relevance Classif	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf	0.73	1	0.44	1	=(0.73+0.44)/2 =0.58
2) Clustering.pdf	0.49	1	0.55	1	=(0.53+0.55)/2 =0.52

Tabela 14: Consulta (“hierarch WITH classif”)

É importante ressaltar que para composição de termos assume-se que o grau de relevância do termo é 1.0, pois quando definida em uma consulta, considera-se que os documentos recuperados devem satisfazê-la.

Suponha agora que o usuário diminua o grau de similaridade do termo “classif” para 0.5, ou seja, o usuário deseja recuperar documentos que contenham “hierarch WITH termo”, onde termo é qualquer termo que seja 50% similar a “classif”. Após o grau de relevância ter sido calculado, foram classificados os documentos listados na tabela 15.

Documento	Weight hierarch	Relevance hierarch	Termo Similar / weight	$GR_{Di} = GR_{Di,Gj}$
1) Vscluster.pdf	0.73	1	-	0.58
2) Clustering.pdf	0.49	1	-	0.52
3) Beil02frequent.pdf	0.53	1	“agglom”/0.79	=(0.53+0.79)/2 =0.66

Tabela 15: Consulta (“hierarch” WITH “classif” [0.5,1])

3.3 Consultas envolvendo grupos de termos

Em consultas desse tipo o cálculo do valor de GR_{Di}, para cada documento recuperado, utiliza as fórmulas (6) e (7), que por sua vez, utilizam os valores de $GR_{Di,Gj}$ dos grupos obtidos pela fórmulas (3), (4) e (5) discutidas na seção anterior.

$$GR_{Di} = \max\{GR_{Di,Gj}, j = 1,\ldots,m\} \quad (6)$$

$$GR_{Di} = \frac{\sum_{j=1}^{m} GR_{Di,Gj}}{m} \quad (7)$$

A seguir são apresentadas as consultas utilizando as fórmulas (6) e (7).

a. (hierarch AND classif) OR (kea AND extract)

Suponha que o usuário deseja recuperar documentos que contenham os termos "hierarch" e "classif", ou os termos "kea" e "extract". Essa consulta permite recuperar documentos que contenham no mínimo um dos dois grupos requeridos na expressão de busca. Os documentos recuperados são apresentados na tabela 16.

Documento	GR_{Di}
1) Vscluster.pdf	0.58
2) Hotho01.pdf	0.58
3) Slonim.pdf	0.56
4) Clustering.pdf	0.52

Tabela 16: Consulta (hierarch AND classif) OR (kea AND extract)

O cálculo do valor de GR_{Di} do primeiro documento, que utiliza a fórmula (6) é apresentado abaixo,

$$GR_{D_1} = \max\{GR_{D_1,G_1}, GR_{D_1,G_2}\}$$
$$GR_{D_1} = \max\{0.58,0\} = 0.58$$

Nesse cálculo, o valor 0 representa que no segundo grupo o documento "Vscluster.pdf" não foi recuperado.

Considere agora, que o usuário diminua o grau de similaridade do termo "classif" para 0.5. Essa consulta permite recuperar documentos que contenham os termos "hierarch" e qualquer termo 50% similar a "classif" ou documentos que contenham os termos "kea" e "extract". Após o grau de relevância dos documentos ter sido calculado, os documentos que foram classificados são descritos na tabela 17.

Documento	GR_{Di}
1) Vscluster.pdf	0.58
2) Hotho01.pdf	0.58
3) Slonim.pdf	0.56
4) Clustering.pdf	0.52
5) Beil02frequent.pdf	0.66

Tabela 17: Consulta (hierarch AND classif [0.5,1]) OR (kea AND extract)

Note que foi recuperado o documento "Beil02frequent.pdf" com GR_{Di}=0.66 que não pertencia ao conjunto solução da consulta anterior. Esse documento foi recuperado porque possui o termo "agglom", o qual satisfaz o grau de similaridade estabelecido para o termo "classif" (0.5). O cálculo do valor de GR_{Di} para esse documento utilizando a fórmula 6 é apresentado abaixo:

$$GR_{D5} = Max\{0.66,0\} = 0.66$$

Nesse cálculo, o valor 0 indica que o documentos não contempla o segundo grupo da expressão de busca, e o valor de GR_{Di} = 0.66 foi obtido levando em consideração o primeiro grupo.

Como visto, o usuário pode combinar grupo de termos considerando os graus de similaridade que influenciam na recuperação dos documentos.

b. (database OR centroid) AND (hierarch WITH classif)

Suponha que o usuário deseja recuperar documentos que contenham exatamente os termos "database" ou "centroid" e "hierarch WITH classif". Nessa consulta, como os grupos são combinados através do conector AND, o sistema realiza no cálculo do valor de GR_{Di}, para cada documento, a média aritmética dos valores $GR_{Di,Gj}$ dos dois grupos (fórmula (7)), e esses valores de $GR_{Di,Gj}$ devem ser maiores do que zero, ou seja, os dois grupos devem estar presentes nos documentos para que eles sejam considerados.

Após o grau de relevância ter sido calculado, os documentos que foram recuperados podem ser vistos na tabela 18.

Documento	GR_{Di}
1) Clustering.pdf	0.6
2) Vscluster.pdf	0.57

Tabela 18: Consulta (database OR centroid) AND (hierarch WITH classif)

Ambos os documentos contemplaram os dois grupos da expressão de consulta, e o cálculo do valor de GR_{Di} para o primeiro documento foi calculado da seguinte forma:

$$GR_{D_1} = \frac{(0.68 + 0.52)}{2} = 0.6$$

4. Considerações Finais

Neste apêndice foram apresentadas algumas consultas realizadas durante os testes do sistema. Nessas consultas as seguintes constatações foram obtidas:

- A importância do grau de relevância na consulta;
- O uso adequado das fórmulas que combinam termos e grupos de termos relacionados pelos conectores OR e AND;
- A importância da utilização da similaridade para a ampliação da busca.