

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um estudo sobre Agrupamento de Documentos  
Textuais em Processamento de Informações não  
Estruturadas Usando Técnicas de "Clustering"**

por

LEANDRO KRUG WIVES

Dissertação submetida à avaliação como requisito parcial para a  
obtenção do grau de Mestre em Ciência da Computação

Dr. José Palazzo Moreira de Oliveira  
Dr. José Mauro Volkmer de Castilho (*Em Memória*)

Orientadores

Porto Alegre, 22 de Abril de 1999.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

WIVES, Leandro Krug

Um Estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de "Clustering" / por Leandro Krug Wives. – Porto Alegre : PPGC da UFRGS, 1999.

102f. : il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-graduação em Computação, Porto Alegre, BR – RS, 1999. Orientador: Oliveira, José Palazzo Moreira de; Orientador: Castilho, José Mauro Volkmer de.

1. Agrupamento de informações. 2. Descoberta de Conhecimento em Textos. 3. Recuperação de informações. I. Oliveira, José Palazzo Moreira de. II. Castilho, José Mauro Volkmer de. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Reitora: Profa. Wrana Panizzi  
Pró-Reitor de Pós-Graduação: Franz Rainer Semmelmann  
Diretor do Instituto de Informática: Prof. Philippe O. A. Navaux  
Coordenador do CPGCC: Profa. Carla Maria Dal Sasso Freitas  
Bibliotecária-Chefe do Instituto de Informática: Beatriz Haro

## **Agradecimentos**

Ao querido amigo e professor Dr. José Mauro Volkmer de Castilho que, apesar de seus problemas de saúde, sempre encontrou tempo para orientar-me. Manifesto que muito aprendi com essa pessoa e que ela ficará para sempre em minha memória.

Ao professor Dr. José Palazzo Moreira de Oliveira, que me acolheu como orientando com toda sua vontade. Seus comentários e sugestões, sempre úteis e experientes, foram indispensáveis para a elaboração deste trabalho.

Aos meus tios, Ladislau e Elvira, que me acolheram como filho durante os dois anos de desenvolvimento deste trabalho.

A todos os colegas e amigos que, de uma forma ou de outra, contribuíram com mais este fruto de meus esforços.

À agência CAPES, pelo fomento.

Ao Instituto de informática da UFRGS, pela utilização de suas dependências, e todo seu pessoal, sempre disposto a cooperar.

## Sumário

<b>Lista de Abreviaturas.....</b>	<b>6</b>
<b>Lista de Figuras .....</b>	<b>7</b>
<b>Lista de Tabelas .....</b>	<b>8</b>
<b>Resumo .....</b>	<b>9</b>
<b>Abstract .....</b>	<b>10</b>
<b>1 Introdução.....</b>	<b>11</b>
1.1 Motivação.....	12
1.2 Objetivos .....	13
<b>2 Levantamento bibliográfico: o processo de agrupamento .....</b>	<b>14</b>
2.1 Introdução à técnica de agrupamento.....	15
2.2 Agrupamento de informações textuais.....	16
2.3 Tipos de agrupamento .....	18
2.3.1 Agrupamento por partição total (flat partition) .....	19
2.3.2 Agrupamento por partição hierárquica .....	19
2.4 Técnicas de agrupamento.....	21
2.4.1 Identificação e seleção de características .....	22
2.4.2 Identificação de similaridades entre objetos – funções de similaridade.....	25
2.4.3 Algoritmos de agrupamento .....	29
2.4.4 Análise dos algoritmos .....	33
<b>3 Agrupamento textual proposto .....</b>	<b>37</b>
3.1 Identificação de palavras.....	37
3.2 Remoção de palavras negativas .....	38
3.3 Cálculo de frequência relativa .....	39
3.4 Determinação de frequência mínima .....	39
3.5 Seleção de características .....	39
3.6 Cálculo de similaridades.....	40
3.7 Agrupamento.....	42
<b>4 Implementação.....</b>	<b>43</b>
4.1 Interface .....	43
4.1.1 Tabela de “stopwords” .....	44
4.1.2 Tabela de coleções.....	45
4.1.3 Tabela de agrupamento.....	50
4.2 Restrições .....	53
<b>5 Estudos de caso .....</b>	<b>55</b>
5.1 Coleção de mensagens eletrônicas .....	58
5.2 Coleção de patentes .....	64
5.2.1 Análise dos diferentes parâmetros .....	64
5.2.2 Análise dos algoritmos .....	66
5.3 Coleção “reuters” .....	67
5.3.1 Formato da coleção.....	68
5.3.2 Utilizando a coleção .....	69
5.3.3 Análise dos resultados .....	70
<b>6 Conclusões.....</b>	<b>73</b>
6.1 Aplicações possíveis.....	75

<b>6.2 Sugestões e trabalhos futuros.....</b>	<b>77</b>
<b>Anexo 1 – Hyperdictionary: A knowledge discovery tool to help information retrieval.....</b>	<b>80</b>
<b>Anexo 2 – Recuperação de informações usando a expansão semântica e a lógica difusa.....</b>	<b>87</b>
<b>Bibliografia.....</b>	<b>100</b>

## **Lista de Abreviaturas**

FRI	Ferramenta de Recuperação de Informações
GSM	Grau de Similaridade Mínimo(a)
IR	Information Retrieval (Recuperação de Informações)
LSI	Latentic Semantic Indexing (Indexação Semântica Latente)
RI	Recuperação de Informações
SI	Sistema de Informações
SRI	Sistema de Recuperação de Informações

## Lista de Figuras

FIGURA 2.1 – Objetivo do agrupamento de informações textuais.....	16
FIGURA 2.2 – Resultado de um agrupamento por partição total e disjunta .....	19
FIGURA 2.3 – Resultado de um agrupamento hierárquico aglomerativo.....	20
FIGURA 2.4 – Resultado de um agrupamento hierárquico global.....	21
FIGURA 2.5 – Etapas do processo de agrupamento .....	22
FIGURA 2.6 – Fórmula da frequência relativa .....	23
FIGURA 2.7 – Distância euclidiana .....	26
FIGURA 2.8 – Espaço com duas dimensões .....	27
FIGURA 2.9 – Função de similaridade “cosine”.....	27
FIGURA 2.10 – Processo de Identificação de grupos .....	29
FIGURA 2.11 – Agrupamento pelo método “cliques” .....	30
FIGURA 2.12 – Resultado do agrupamento pelo método “strings” .....	34
FIGURA 2.13 – Problema do agrupamento pelo método “strings” .....	34
FIGURA 2.14 – Resultado fictício de um agrupamento pelo método “single link” .....	35
FIGURA 2.15 – Similaridade mínima no algoritmo “stars” .....	35
FIGURA 3.1 – Fórmula da média por operadores “fuzzy” .....	41
FIGURA 3.2 – Fórmula do cálculo do grau de igualdade entre pesos .....	41
FIGURA 4.1 – Tabela de manipulação de palavras negativas .....	44
FIGURA 4.2 – Caixa de diálogo para adicionar categoria .....	44
FIGURA 4.3 – Caixa de diálogo para adicionar palavras negativas .....	45
FIGURA 4.4 – Orelha de manipulação de coleções .....	45
FIGURA 4.5 – Janela de progresso do processamento.....	46
FIGURA 4.6 – Seleção de arquivos de uma coleção .....	47
FIGURA 4.7 – Listagem de palavras (características) de um documento.....	48
FIGURA 4.8 – Área de seleção de palavras negativas .....	48
FIGURA 4.9 – Área de configuração de opções de pré-processamento .....	49
FIGURA 4.10 – Orelha de agrupamento .....	50
FIGURA 4.11 – Matriz de similaridades .....	51
FIGURA 4.12 – Janela de informações estatísticas.....	52
FIGURA 4.13 – Listagem de palavras principais de um grupo.....	53
FIGURA 5.1 – Fórmula de “macroaverage recall” .....	59
FIGURA 5.2 – Fórmula de “macroaverage precision” .....	59
FIGURA 5.3 – Resultados de “macroaverage recall” utilizando o método “best-star” .....	60
FIGURA 5.4 – Novos resultados de “macroaverage recall” utilizando o método “best-star” .....	61
FIGURA 5.5 – Comparação dos resultados de “macroaverage recall” .....	62
FIGURA 5.6 – Quantidade de “clusters” obtidos em cada algoritmo .....	62
FIGURA 5.7 – Desempenho médio geral.....	63

## Lista de Tabelas

TABELA 2.1 – Matriz de similaridade entre objetos .....	28
TABELA 2.2 – Matriz de similaridade entre objetos .....	33
TABELA 5.1 – Listas de palavras negativas utilizadas.....	57
TABELA 5.2 – Parâmetros e seus respectivos tempos de processamento .....	65
TABELA 5.3 – Quantidade de palavras por coleção.....	65
TABELA 5.4 – Resultados dos diferentes algoritmos.....	66
TABELA 5.5 – Resultados para GSM de cinquenta por cento.....	67
TABELA 5.6 – Tempo de processamento da matriz de similaridades .....	71



## Resumo

Atualmente, técnicas de recuperação e análise de informações, principalmente textuais, são de extrema importância. Após o grande BOOM da Internet, muitos problemas que já eram conhecidos em contextos fechados passaram a preocupar também toda a comunidade científica. No âmbito deste trabalho os problemas relacionados à sobrecarga de informações, que ocorre devido ao grande volume de dados a disposição de uma pessoa, são os mais importantes.

Visando minimizar estes problemas, este trabalho apresenta um estudo sobre métodos de agrupamento de objetos textuais (documentos no formato ASCII), onde os objetos são organizados automaticamente em grupos de objetos similares, facilitando sua localização, manipulação e análise.

Decorrente deste estudo, apresenta-se uma metodologia de aplicação do agrupamento descrevendo-se suas diversas etapas. Estas etapas foram desenvolvidas de maneira que após uma ter sido realizada ela não precisa ser refeita, permitindo que a etapa seguinte seja aplicada diversas vezes sobre os mesmos dados (com diferentes parâmetros) de forma independente.

Além da metodologia, realiza-se um estudo comparativo entre alguns algoritmos de agrupamento, inclusive apresentando-se um novo algoritmo mais eficiente. Este fato é comprovado em experimentos realizados nos diversos estudos de caso propostos.

Outras contribuições deste trabalho incluem a implementação de uma ferramenta de agrupamento de textos que utiliza a metodologia elaborada e os algoritmos estudados; além da utilização de uma fórmula não convencional de cálculo de similaridades entre objetos (de abordagem *fuzzy*), aplicada a informações textuais, obtendo resultados satisfatórios.

**Palavras-chave:** Agrupamento de Informações, Descoberta de Conhecimento em Textos, Recuperação de Informações

**TITLE:** “A STUDY ABOUT ARRANGEMENT OF TEXTUAL DOCUMENTS APPLIED TO UNSTRUCTURED INFORMATION PROCESSING USING CLUSTERING TECHNIQUES”

## **Abstract**

The Internet is the vital media of today and, as being a mass media, problems known before to specific fields of Science arise. One of these problems, capable of annoying many people, is the information overload problem caused by the excessive amount of information returned in response to the user’s query.

Due to the *information overload* problem, advanced techniques for information retrieval and analysis are needed. This study presents some aids in these fields, presenting a methodology to help users to apply the clustering process in textual data. The technique investigated is capable of grouping documents of several subjects in clusters of documents of the same subject. The groups identified can be used to simplify the process of information analysis and retrieval.

This study also presents a tool that was created using the methodology and the algorithms analyzed. The tool was implemented to facilitate the process of investigation and demonstration of the study. The results of the application of a fuzzy formula, used to calculate the similarity among documents, are also presented.

**Keywords:** Clustering, Knowledge Discovery from Texts, Information Retrieval

# 1 Introdução

Os processos de manipulação de informações estão cada vez mais incorporados no cotidiano das pessoas. Com o desenvolvimento de tecnologias de comunicação cada vez mais eficientes e, conseqüentemente, com o advento da Internet, o volume de informações que uma pessoa tem acesso cresce diariamente. Porém, este volume crescente de informações torna mais difícil a tarefa de assimilação da informação.

Quando uma pessoa tem facilidade de acesso à grande quantidade de informações, também deve possuir meios que indiquem quais caminhos deve tomar para obter a informação de que necessita. Porém, é sabido, através de estudo realizado anteriormente [WIV 97], que os meios de acesso (*Sistemas de Recuperação de Informações*), quando existem, não conseguem indicar de forma eficiente o local onde podem ser encontradas as informações desejadas pelo usuário. Isso porque as formas de acesso à informação, oferecidas por estes sistemas, não são as ideais.

A consulta por *termo* ou *palavra-chave* é uma das formas de acesso à informação mais utilizadas nos *Sistemas de Recuperação de Informações (SRI)*. Neste caso, o usuário fornece ao sistema palavras-chave que identifiquem o assunto desejado por ele e o sistema retorna todos aqueles documentos que possuem estas palavras. O *Altavista®* é um exemplo de ferramenta que utiliza este sistema na Internet.

Esta forma ou método de consulta possui muitos problemas que são inerentes à linguagem utilizada, esses problemas são pertencentes a uma classe denominada de classe dos *problemas do vocabulário (vocabulary problem)*, que advém da própria natureza ambígua da linguagem. Isso porque a *linguagem natural* quotidiana (o português, por exemplo) permite que as pessoas utilizem diversas palavras diferentes para indicar um mesmo objeto. Com isso, o vocabulário utilizado pelo autor pode diferir do vocabulário utilizado pela pessoa que realiza a busca.

Tudo isso dificulta a localização da informação, já que as pessoas não sabem que palavras devem utilizar para especificar o assunto que desejam obter. Este problema é agravado pelo fato de que existe uma diversificação muito grande de indivíduos utilizando a Internet, não sendo possível estabelecer um único vocabulário. Esta classe de problemas é discutida em um artigo anterior [WIV 98a], que se encontra em anexo (Anexo 1), e apresenta uma solução parcial, indicando que ainda são necessários muitos estudos neste campo.

Com isso muitos documentos irrelevantes (de assuntos correlatos ou não ligados diretamente) são retornados. Desta forma, as pessoas (usuários) recebem uma grande quantidade de informações, geralmente desordenadas e sem muito sentido, que devem ser analisadas exaustivamente até que alguma coisa interessante (relevante) seja realmente encontrada. Esta quantidade muito grande de informações, além de dificultar e tornar mais demorado o processo, pode levar uma pessoa a um problema conhecido por *sobrecarga de informações (information overload)*. A sobrecarga de informações ocorre quando a pessoa recebe uma quantidade muito grande de informações (mesmo que irrelevantes) e não consegue tratá-las ou assimilá-las [CHE 96].


É claro que existem outras formas mais práticas de localização de informações. Uma delas consiste na criação de índices hierárquicos, que agrupam as informações por assuntos similares, constituindo assim uma cadeia ou árvore de conhecimento (*portais*). O usuário pode então navegar pelos ramos desta árvore, selecionando os assuntos e sub-

assuntos de interesse, até que a informação que deseja seja encontrada. Um exemplo de ferramenta que utiliza este sistema é a ferramenta *Yahoo*®.

Um dos maiores problemas desta forma de acesso consiste na identificação correta dos assuntos que determinada informação trata. Esta identificação, realizada para fins de indexação, é feita manualmente por pessoas, o que acarreta em problemas de atraso (já que há um limite no número de informações que podem ser indexadas diariamente por um ser humano) ou de indexação imprecisa (onde a pessoa que indexa pode não categorizar corretamente a informação, colocando-a em uma categoria diferente da categoria que a informação realmente pertence).

Estes diversos problemas relacionados à busca e recuperação de informações são discutidos em dois trabalhos anteriores [WIV 97, WIV 98a]. Estes trabalhos motivam profundamente o estudo apresentado nesta dissertação. Diante de todos estes problemas, e diante do fato de que não existem soluções definitivas para eles, surge a possibilidade de elaboração de um sistema capaz de agrupar informações similares automaticamente.


Este sistema, identificando as diferentes características da informação (as diferentes palavras) de um mesmo grupo (assunto), é capaz de amenizar os problemas do vocabulário e da sobrecarga de informações.

Além disso, de posse de grupos de documentos de um mesmo assunto, as características peculiares de cada grupo de informações podem ser coletadas e armazenadas em uma base de conhecimento. Esta base de conhecimento pode ser utilizada por um *sistema especialista*, a fim de classificar informações automaticamente para o ário.

O presente trabalho oferece um meio de acesso ou localização de informações mais facilitado, onde as informações são agrupadas em categorias onde os documentos possuem um grau mínimo de similaridade (semelhança). Deste modo, o usuário pode identificar o grupo de documentos que mais se aproxima do assunto por ele desejado e então analisar todos os documentos que estão contidos neste grupo.

## 1.1 Motivação

Diante dos problemas citados anteriormente, em sua maioria relacionados à grande quantidade de informações disponíveis, conclui-se que novos meios de acesso e manipulação de grandes quantidades de informações textuais devem ser criados. Esta afirmação é suportada por muitos autores. O estudo realizado por *Jean Moscarola* [MOS 98] é um exemplo, pois cita dois problemas principais, decorrentes da sobrecarga de informações: um deles relacionado à localização de informações relevantes e outro relacionado à identificação e extração de conhecimento presente nas informações relevantes encontradas.

Para identificar a informação relevante é necessário passar horas diante de uma ferramenta de busca de informações (um *search engine*) como o vista™ (encontrada na *Internet* no endereço <http://www.altavista.digital.com>). Depois de identificada a informação relevante, e essa informação geralmente não vem isolada, mas sim acompanhada de muitas outras ou espalhada em diversos documentos, é necessário analisar o conteúdo desta informação e filtrar ou extrair os dados realmente importantes.

Atualmente há uma área emergente que se preocupa em estudar e solucionar estes dois passos previamente citados. A área é denominada *Análise de dados Textuais*

(*Textual data Analysis* conforme Jean Moscarola [MOS 98]) ou, mais recentemente, *Descoberta de Conhecimento em Textos (Knowledge Discovery from Texts)*, de acordo com Ronen Feldman [FEL 97]. Ambas referem-se à tarefa de recuperar, filtrar, manipular e resumir o conhecimento retirado de grandes fontes de informações textuais e apresentá-lo para o usuário final utilizando-se de diversos recursos, geralmente diferentes dos originais (como, por exemplo, gráficos, listas ou tabelas).

Tudo isto é necessário, pois, sem estas ferramentas ou técnicas, a tarefa de identificação do conhecimento torna-se tediosa e, muitas vezes, ineficiente. Com a utilização de uma ferramenta de análise de conhecimento, por exemplo, o conteúdo de um conjunto de documentos pode ser sumarizado e apresentado em forma de gráficos que indiquem a relação semântica dos termos que os compõem. Uma ferramenta com estas características pode ainda apresentar um resumo das palavras mais importantes dos documentos, facilitando muito sua análise.

Esse tipo de análise é oferecido pela ferramenta *Altavista*<sup>TM</sup>, que apresenta, na opção *Refine*, um gráfico contendo as palavras mais relevantes das páginas retornadas e sua relação semântica. Deste modo, através deste gráfico, o usuário consegue obter uma idéia do assunto abordado pela coleção de páginas retornada sem ter que ler todas estas páginas (poupando tempo). O usuário pode ainda refazer a consulta, aprofundando-se no assunto ou buscando novos conhecimentos.

## 1.2 Objetivos

O objetivo primário deste trabalho é estudar técnicas de agrupamento de objetos (informações) textuais, proporcionando uma forma de acesso facilitada à informação. Deste modo é possível particionar uma grande coleção de documentos, isolando aqueles pertencentes a um mesmo assunto e facilitando a identificação de documentos relevantes para o usuário. Com isso, aplicando-se técnicas adicionais, é possível identificar o assunto ou conhecimento específico de cada grupo, facilitando o processo de recuperação de informações ou descoberta de conhecimento.

Para realizar o processo de análise dos métodos ou técnicas de agrupamento, propõe-se, como objetivo complementar, a implementação de uma ferramenta de agrupamento de informações textuais (documentos textuais) que permita a aplicação prática dos métodos estudados.

Para que os métodos possam ser analisados e comparados torna-se necessária a adoção de uma coleção de referência, que contenha documentos selecionados e modelados para tal. Deste modo, um dos objetivos complementares deste trabalho é a adoção e utilização de uma coleção de referência.

No próximo capítulo, encontra-se uma breve introdução ao assunto – a revisão bibliográfica, com o objetivo de demonstrar sua abrangência e os trabalhos já desenvolvidos. No capítulo 3, é apresentada a metodologia adotada neste trabalho, assim como seu embasamento teórico. A implementação desta metodologia é descrita no capítulo 4. No capítulo 5, são apresentados os estudos de caso utilizados para validar a metodologia adotada, utilizando-se da ferramenta implementada e de técnicas de validação específicas, detalhadas no mesmo capítulo. Finalmente, o capítulo 6 apresenta os resultados e conclusões obtidos no decorrer deste estudo.

## 2 Levantamento bibliográfico: o processo de agrupamento

A tarefa de *agrupar objetos*, também conhecida por *clustering*, não é recente. O conceito de **agrupado** (*cluster*) é tão antigo quanto as bibliotecas. Muitos anos antes da criação dos primeiros computadores, as pessoas já realizavam este processo manualmente, pois agrupar elementos similares facilita a localização de informações.

Poucos anos depois que os computadores passaram a ser utilizados por órgãos governamentais americanos, os primeiros algoritmos de *agrupamento* de objetos auxiliado por computador surgiram e foram implementados. Com isso, atualmente, existem muitos algoritmos de agrupamento de objetos.

Na informática, segundo *Jiawei Han* [HAN 96], o agrupamento de objetos/informações é uma técnica de *Descoberta de Conhecimento e Mineração de Dados* estudada pela área de *Inteligência Artificial*. Muitos algoritmos de agrupamento de objetos já foram implementados, estudados e aplicados em diversas áreas do conhecimento, tais como a psiquiatria (com o objetivo de redefinir categorias de diagnóstico existentes), a arqueologia (para investigar os relacionamentos entre os vários tipos de artefatos) e a Genética (especialmente após a criação do projeto *GENOMA Humano*). Atualmente, as áreas de marketing e de economia têm despertado grande interesse pelas técnicas de agrupamento, com o objetivo de obter conhecimento sobre os padrões de consumo.

Porém, na área de *Sistemas de Informação (SI)*, principalmente em objetos textuais, estas técnicas de agrupamento **agrupado** um pouco aplicadas. Na *Recuperação de Informações* (um dos ramos de *SI*), por exemplo, os algoritmos de agrupamento têm sido utilizados há algum tempo para fins de organização de dados, já que a recuperação de dados similares torna-se mais eficiente quando estes estão organizados em um mesmo bloco físico de dados. Porém, sua utilização para fins de sumarização de informações não foi muito explorada.

Isto demonstra que ainda há muito que pesquisar nesta área, buscando novas aplicações não só para a tarefa de *agrupamento*, mas também outras tecnologias como, por exemplo, a descoberta de conhecimento, a sumarização [KUP 95] e a extração de informações [COW 96]. O *agrupamento* de informações textuais é um destes sub-ramos que ainda necessita de maiores estudos pois somente alguns destes algoritmos foram ou estão sendo testados com informações textuais (documentos), outros específicos estão em desenvolvimento.

Os estudos que existem são, em grande parte, relacionados com a *classificação* de documentos, que não deve ser confundida com *agrupamento*. A seção seguinte contém maiores detalhes sobre as diferenças entre *agrupamento* e *classificação*.

## 2.1 Introdução à técnica de agrupamento

Basicamente, segundo *Gerald Kowalski* [KOW 97], um *aglomerado* é um grupo de objetos similares, geralmente uma classe, que possui um título mais genérico capaz de representar todos os elementos nela contidos.

No contexto de informações textuais, existem basicamente dois tipos de agrupamento: um para agrupar termos (palavras) e outro para agrupar documentos. No primeiro caso, grupos de termos similares são identificados a fim de construir dicionários de palavras (denominados *thesaurus*) que definam um mesmo assunto. Assim, estas palavras similares podem ser utilizadas em ferramentas de recuperação de informações a fim de expandir a consulta do usuário com o intuito de obter melhores resultados (documentos mais relevantes).

Este tipo de agrupamento não é relevante para esta dissertação, que busca realizar o agrupamento de documentos similares. Porém, estudos foram realizados em relação ao agrupamento de termos, durante o decorrer deste trabalho e de trabalhos anteriores [WIV 98a, WIV 98b] (anexos 1 e 2). Estes trabalhos anteriores apresentam os motivos e aplicações para tal processo. Detalhes teóricos, para aqueles que desejam aprofundar-se no assunto, são apresentados por *Leandro Wives* [WIV 96] em uma breve mas detalhada introdução ao assunto.

No escopo deste trabalho, a técnica de agrupamento consiste em organizar uma série desorganizada de objetos em grupos de objetos similares. Este tipo de técnica é recomendado quando não há uma discriminação prévia de classes, sendo útil em casos onde não há a possibilidade de alocar um especialista na tarefa de separação de objetos em classes. Em muitos casos, pode não ser possível utilizar um especialista humano por não existir na instituição alguém com conhecimento adequado sobre o domínio das informações em questão.

Já a classificação ou *categorização* consiste em identificar a classe a que pertence determinado objeto. Para tanto é necessário conhecer previamente as características de cada classe. Justamente por isso a classificação necessita de duas etapas: uma etapa de **E**ndizado, onde as classes são identificadas e caracterizadas, e uma etapa de *classificação* propriamente dita, onde os elementos são identificados (classificados) de acordo com as classes existentes.

Tendo-se esta visão, é possível considerar a técnica de *agrupamento* como uma etapa (passo) anterior à **E**nsificação. Um exemplo ilustrativo e não técnico é a “síndrome de Tarzan”, apresentada por *Edgar Chávez* [CHA 98]:

**Tarzan é um garoto esquecido na selva por acidente e é criado pelos animais. Na selva há uma quantidade muito grande de objetos, nos quais se desconhecem os nomes. Alguns destes objetos são comestíveis, outros são venenosos, perigosos ou inofensivos, curativos, etc. Como Tarzan sobrevive até a idade adulta, supõe-se que encontrou uma maneira de classificar os objetos sem morrer no processo. É claro que se não utilizamos a mesma linguagem não utilizaremos as mesmas etiquetas aos mesmos objetos; porém Tarzan pode encontrar uma ampla série de subdivisões nas etiquetas que nós podemos atribuir. Observação importante: Para sobreviver é irrelevante o tipo de etiqueta (números, símbolos ou letras) que se atribui a cada classe de objetos, sempre e quando sejam distintas para cada um destes e possam ser organizadas em hierarquias.**

Neste caso, tem-se que a técnica de agrupamento busca etiquetar uma selva abstrata de objetos, de tal forma que os objetos semelhantes permaneçam na mesma classe e objetos diferentes encontrem-se em classes distintas.

Portanto, informalmente, o objetivo da técnica de agrupamento é identificar os objetos que possuem algo em comum (características em comum) e separá-los, agrupando-os em subconjuntos de objetos similares. Estes objetos, antes de serem agrupados, podem ser os mais variados possíveis.

Como definição mais formal, o *aglomerado* de objetos, também chamado de *cluster* em alguns casos, pode ser definido segundo *Everitt* [EVE 74] como:

- a) Um aglomerado é um conjunto de entidades que são semelhantes, e entidades pertencentes a aglomerados diferentes são diferentes;
- b) Um aglomerado é uma agregação de pontos no espaço tal que a distância entre os pontos em um mesmo aglomerado é menor que a distância entre pontos de diferentes aglomerados;
- c) Os aglomerados podem ser descritos como regiões conexas de um espaço multidimensional que contém uma grande densidade relativa de pontos. As regiões estão separadas umas das outras por regiões de baixa densidade relativa de pontos.

## 2.2 Agrupamento de informações textuais

O objetivo do agrupamento de informações textuais é separar uma série de documentos dispostos de forma desorganizada em um conjunto de grupos que contenham documentos de assuntos similares (FIGURA 2.1).

Para que isto seja feito, parte-se do princípio da *Hipótese de Agrupamento* (*Cluster Hypothesis*), levantado por *Rijsbergen* [RIJ 79]. Este princípio diz que objetos semelhantes e relevantes a um mesmo assunto tendem a permanecer em um mesmo grupo (*cluster*), pois possuem atributos em comum.

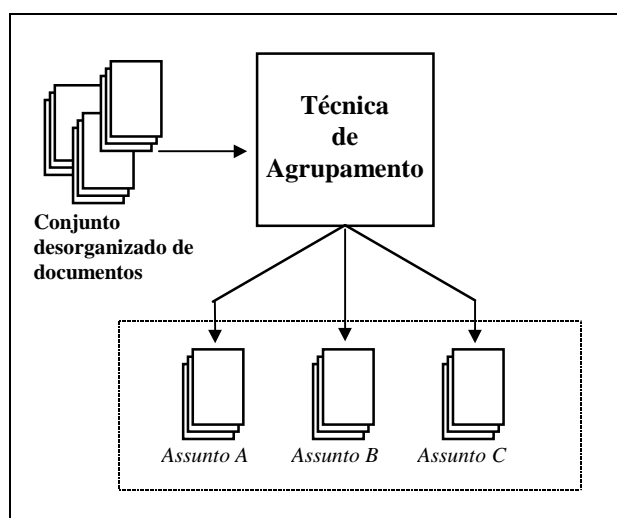


FIGURA 2.1 – Objetivo do agrupamento de informações textuais

O problema, segundo *Feldman* [FEL 97], é que a maioria das técnicas de *agrupamento* foi desenvolvida para atuar sobre dados estruturados, ou seja, aqueles



dados convencionais, armazenados em Sistemas de Gerência de Bancos de Dados, mais fáceis de serem tratados por meios computacionais.

Somente há pouco tempo, principalmente com o surgimento da Internet, os chamados dados não-estruturados, formados por imagens, textos, gráficos, etc, ganharam maior atenção. O interesse por este tipo de informação cresceu muito na área de *Descoberta de Conhecimento*, surgindo inclusive uma nova ramificação conhecida como *Extração de Informações* [COW 96]. Estes dados possuem as mesmas informações que os dados tradicionais, estruturados, porém sua forma é textual o que dificulta a identificação de suas características importantes.

Com isso, alguns algoritmos foram ou estão sendo testados com informações textuais, e outros específicos para estes tipos de informação estão em desenvolvimento. Porém, segundo *Edgar Chávez* [CHA 98], cada algoritmo possui um ou mais parâmetros próprios que são difíceis de sintonizar corretamente. Logo, pelo fato destes parâmetros serem muito dependentes do algoritmo ou do domínio dos objetos utilizados, ainda não há um padrão estabelecido e torna-se muito difícil definir um procedimento geral de *agrupamento* automático de informações.

Como já citado anteriormente, dois grandes grupos de algoritmos destacam-se na área de agrupamento de informações textuais. Um focando o agrupamento de termos (palavras-chave) similares e outro focando o agrupamento de documentos.

No primeiro grande grupo as técnicas de *agrupamento* visam identificar relacionamentos entre palavras. Com isso, podem ser identificadas as palavras mais utilizadas em determinado assunto, incluindo sinônimos e variações morfológicas, que podem ser armazenadas em uma estrutura conhecida como *thesaurus* (dicionário contendo informações semânticas e correlações entre palavras). De posse do *thesaurus* ou de um *Hiperdicionário* [WIV 98a], é possível adicionar termos na consulta do usuário, levando-o mais rapidamente para o conjunto de documentos relevantes.

Estes termos adicionados na consulta são supostamente mais ligados com o assunto, porém, não há técnica de análise de relações entre palavras perfeita. Com isso, podem ocorrer casos onde as palavras adicionadas acabam atrapalhando no resultado, desviando a consulta para outros assuntos. Existe uma técnica específica que se preocupa em solucionar os problemas relacionados à identificação de termos correlacionados e em expandir consultas. Esta técnica é denominada *expansão semântica*, já que busca identificar o assunto de uma consulta (ou seu contexto) e então expande esta consulta com palavras que identificam melhor este assunto. Em estudos anteriores [WIV 98a, WIV 98b], que se encontram em anexo, podem ser obtidos maiores detalhes sobre esta técnica.

O segundo grande grupo é muito utilizado por ferramentas de recuperação de informações para identificar documentos similares e armazená-los em seções (blocos) contíguas do dispositivo de armazenamento. Desta forma, quando um dos documentos for retornado em resposta a uma consulta, todos os outros documentos do mesmo grupo são considerados relevantes para a mesma consulta, sendo então retornados rapidamente. Esta hipótese parte do princípio de que todos os documentos pertencentes a um grupo tratam de um assunto similar, e que grupos diferentes possuem documentos que tratam de assuntos diferentes. Deste modo, a informação permanece agrupada, classificada de acordo com o assunto, facilitando sua localização e manipulação.

Além destes, identifica-se uma grande quantidade de algoritmos genéricos de agrupamento que foram desenvolvidos ao longo dos anos. Alguns destes poderiam ser

utilizados para ambas as aplicações (agrupamento de termos ou de documentos). Porém, devido a esta diversidade, é necessário optar por um grupo ou família de algoritmos que se encaixe melhor ao tipo de informação textual.

Grande parte dos estudos da área de agrupamento textual realizados até o momento focam-se nos algoritmos provenientes da família denominada *graphic-theoretic*. Esta dissertação basear-se-á nesta mesma família, tanto pela diversidade (já que não é possível estudar todos os algoritmos existentes) quanto pela sua aceitação.

## 2.3 Tipos de agrupamento textual

O agrupamento pode ser classificado de acordo com dois critérios: o primeiro em relação à forma como os grupos são construídos e o segundo em relação à complexidade do tempo de execução do algoritmo.

Segundo *Douglass Cutting* [CUT 92], quanto à forma, há dois tipos de agrupamento: o agrupamento por partição e o agrupamento hierárquico. Eles dizem respeito à forma na qual os grupos são constituídos. No primeiro tipo de agrupamento, denominado *por partição*, os objetos são distribuídos em classes distintas, não havendo relação direta entre as classes. Este tipo de agrupamento é denominado *agrupamento de partição total (flat partition)* e os documentos são separados exaustivamente e colocados em grupos totalmente diferentes. No segundo tipo, denominado *partição hierárquica (hierarchic partition)*, o processo de identificação de grupos é geralmente realimentado recursivamente, utilizando tanto objetos quanto grupos já identificados previamente como entrada para o processamento. Deste modo, constrói-se uma hierarquia de grupos de objetos, estilo uma árvore.

Quanto à complexidade em relação ao tempo de processamento, os algoritmos de agrupamento podem ser considerados *constantes* [SIL 97], *lineares* [CUT 92] ou *exponenciais de ordem quadrática* [CUT 92]. Porém, como o objetivo desta dissertação não é realizar uma análise de complexidade de algoritmos, a classificação dos algoritmos de agrupamento em relação a este critério é somente citada, e não discutida.

Os algoritmos de tempo constante têm o objetivo de limitar o tempo máximo de processamento, ou ao menos descobrir o tempo gasto por cada elemento e estimar um tempo ou número máximo de comparações a ser feito. Geralmente limitam o tamanho ou o número de *clusters*. Porém, no estágio atual da tecnologia, não há algoritmos de tempo constante que possam ser utilizados para processamento em tempo real (tempo constante não significa, necessariamente, tempo real).

Os chamados de tempo linear possuem a característica de aumentar seu tempo de processamento de acordo com o número de elementos. Este tempo, porém, ao aumentar, cresce de modo sutil e gradativo (não exponencial), já que nem todos os elementos necessitam ser comparados com todos.

Os considerados quadráticos crescem exponencialmente, pois sempre que um elemento é adicionado ele deve ser comparado com todos os outros, aumentando a quantidade de comparações entre os elementos.

### 2.3.1 Agrupamento por partição total (flat partition)

Basicamente, a técnica consiste em agrupar os documentos em um número predeterminado de *aglomerados* (*clusters*) distintos. Os documentos são agrupados de forma tal que todos os elementos de um mesmo *aglomerado* possuem um grau mínimo de semelhança, que é indicado pelo número de características em comum que possuem.

Apesar de constituir grupos distintos, esta técnica permite a possibilidade de colocar ou não determinado documento em mais de um grupo. Quando os documentos são atribuídos a um único grupo diz-se que o processo é *disjunto*. Caso um documento seja atribuído a mais de um grupo, por possuir forte relação com mais de um grupo, diz-se que o processo não é disjunto.

Geralmente os algoritmos adotam restrições que impedem que um elemento pertença a mais de um grupo (ver seção 2.4.3), atribuindo o objeto ao *aglomerado* de maior relação (similaridade). Porém, esta restrição pode ser (e é) quebrada em alguns casos. De qualquer modo, mesmo que um objeto (documento) seja atribuído a mais de um *aglomerado*, não há construção de hierarquias e os grupos continuam sendo considerados isolados.

Na FIGURA 2.2 os documentos são representados pelos pequenos círculos. Os *aglomerados*, que aglomeram os documentos, são representados pelos grandes círculos. Como pode ser visto, os *aglomerados* não possuem ligações entre si, sendo totalmente isolados. Assim, os documentos são totalmente separados.

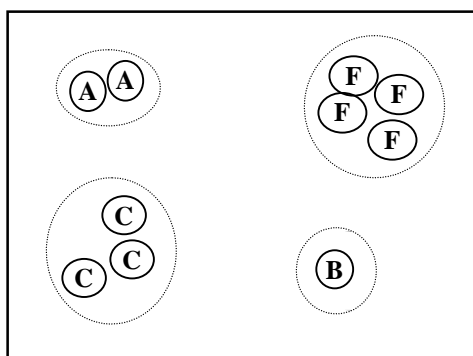


FIGURA 2.2 – Resultado de um agrupamento por partição total e disjunta

A maior desvantagem é constituída pelo fato de não haver uma estrutura de navegação, que indique os assuntos mais abrangentes, os assuntos mais específicos e os inter-relacionamentos entre eles. Neste caso, pode ser difícil para o usuário localizar os documentos de que necessita.

### 2.3.2 Agrupamento por partição hierárquica

No agrupamento de tipo hierárquico os grupos identificados são recursivamente analisados, fazendo com que as relações entre os grupos também sejam identificadas.

Para que a hierarquia seja identificada podem ser aplicados algoritmos específicos ou podem ser utilizados algoritmos de partição total, aplicados de forma recursiva. A maior diferença entre os algoritmos hierárquicos e os anteriores está no fato de que os algoritmos de agrupamento por partição total realizam somente um passo de processamento.

Há duas formas de partição hierárquica: *aglomerativa ou global*. Na partição *hierárquica aglomerativa*, os elementos são agrupados em pares de maior similaridade.

Para tanto, inicialmente, todos os documentos são colocados em *aglomerados* diferentes, ou seja, um *aglomerado* para cada documento. A seguir, os *aglomerados* são analisados aos pares. O par de elementos que possuir maior grau de similaridade é agregado<sup>1</sup> em um único *aglomerado*, que passa a representá-los. A partir deste momento, tem-se um cluster a menos. Repete-se o processo de análise de similaridades, onde o par mais similar é agrupado. A análise é aplicada recursivamente em toda a coleção, até que haja somente um único *aglomerado*. Deste modo, uma hierarquia de *aglomerados* é construída, conforme a FIGURA 2.3.

No agrupamento global, os objetos são agrupados de forma similar à forma utilizada no agrupamento de partição total, ou seja, todos os documentos de um grupo são identificados (não somente os pares). Após, o processo é refeito e os *aglomerados* entre os grupos resultantes são identificados. O processo é aplicado recursivamente até que um único grupo seja identificado. A FIGURA 2.4 apresenta o resultado da utilização desta técnica. Porém, a técnica de análise global de elementos consome muitos recursos computacionais. A técnica aglomerativa é um pouco mais econômica, já que analisa pares de elementos.

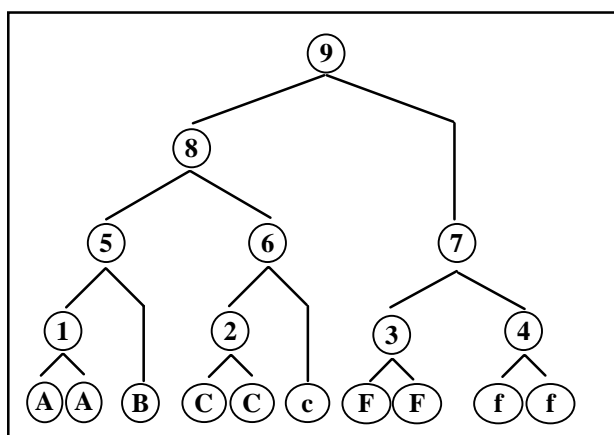


FIGURA 2.3 – Resultado de um agrupamento hierárquico aglomerativo

<sup>1</sup> Quando um conjunto de documentos é agregado, é possível utilizar-se de um elemento central capaz de representar todos os elementos deste conjunto. Este elemento central é denominado *profile* ou *centróide* (*cluster centroid*), e recebe este nome devido ao fato de cada documento ser considerado um vetor (uma lista) de palavras. Com isso, o centróide é o vetor central que contém todas as características dos demais vetores. Deste modo, o centróide é um conjunto de características centrais do cluster, que consegue representar todos os elementos que pertencem a este cluster (e somente estes). Há diversos estudos que indicam que quando se necessita realizar comparações entre determinado grupo e outro (ou entre um grupo e um elemento externo) não é necessário realizar comparações entre todos os elementos do grupo, mas sim, somente, entre os seus centróides. Um dos estudos mais interessantes sobre este assunto é realizado por *Douglass Cutting* [CUT 93]. Este estudo apresenta testes que confirmam que a utilização de um centróide (chamado no caso de *profile* ou *cluster digest*), além de tornar mais rápido o processo de análise, corresponde aos resultados obtidos com a utilização de todos os elementos do grupo. Porém, apesar destas vantagens, o centróide deve ser construído com cuidado. Os resultados podem ser falhos se o conjunto de características selecionadas para fazer parte do centróide não forem muito bem escolhidas, não representando corretamente os elementos de determinado grupo. A determinação do centróide deve ser feita por um especialista, ou por uma técnica automática muito bem planejada. Nos estudos de *Salton* [SAL 83] podem ser encontradas algumas destas técnicas.

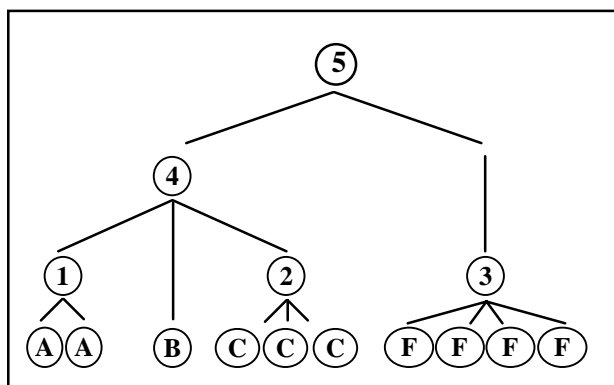


FIGURA 2.4 – Resultado de um agrupamento hierárquico global

Em ambos os casos é produzida uma árvore, onde as folhas desta árvore representam os elementos individuais e os *nodos* intermediários correspondem a grupos formados pelo agrupamento (*merge*) de seus grupos filhos.

Já que este tipo de agrupamento produz uma árvore, sua grande vantagem, segundo *Moses charikar* [CHA 97], diz respeito à facilidade de localização da informação, já que o usuário pode ir navegando pelos ramos de informação mais relevantes à sua necessidade. Isso porque as informações estão agrupadas por assunto, e estes estão interligados de acordo com seus relacionamentos, constituindo uma hierarquia de assuntos. No caso, começa-se pelo *nodo pai* (topo), e decide-se qual dos dois lados é mais similar ao que se procura. A análise é aplicada recursivamente, tomando o rumo do sub-ramo mais adequado até que se chegue ao elemento desejado.

Um dos grandes problemas nesta classe de agrupamento diz respeito à atribuição de nomes (labels) aos nodos da árvore que vai sendo construída. Se um *label* incorreto é atribuído ao nodo, o usuário não tem condições de compreender (não tem uma idéia correta do assunto) do que se trata o sub-ramo, e não consegue navegar corretamente.

## 2.4 Técnicas de agrupamento

Das diversas técnicas de agrupamento existentes, as técnicas mais utilizadas em agrupamento de objetos textuais são aquelas pertencentes à classe *graphic-theoretic*. É claro que existem muitas outras técnicas, porém, esta dissertação está limitada ao escopo desta classe.

Na grande maioria das técnicas, inclusive estas, são utilizados algoritmos que realizam três etapas básicas e distintas: *identificação e seleção de características*, *cálculo de similaridades* e *identificação de aglomerados (clusters)* – o agrupamento propriamente dito. Estas etapas são ilustradas na FIGURA 2.5.

A primeira etapa identifica características nos objetos, ou seja, identifica palavras nos documentos e após seleciona (globalmente ou localmente) aquelas que possuem maior grau de discriminação (que caracterizam melhor o objeto). Como resultado desta etapa, são geradas listas de palavras (características) relevantes, que identificam cada documento.

A segunda etapa identifica os graus de similaridade entre os objetos (documentos), utilizando para isso as listas de características identificadas na etapa anterior. Como resultado desta etapa, obtém-se uma matriz que contém os valores de

similaridade entre todos os objetos. Quanto mais características em comum possuem os objetos, mais similares eles são.

Por fim, a etapa de agrupamento (em si) consiste em identificar correlações entre os elementos da matriz, de acordo com as restrições impostas por cada algoritmo. Dependendo da técnica utilizada para a montagem da matriz, e do algoritmo de agrupamento, todos os elementos devem ser comparados com todos, o que pode classificar o algoritmo como sendo de ordem quadrática. Ao final desta etapa, têm-se os grupos e seus respectivos elementos.

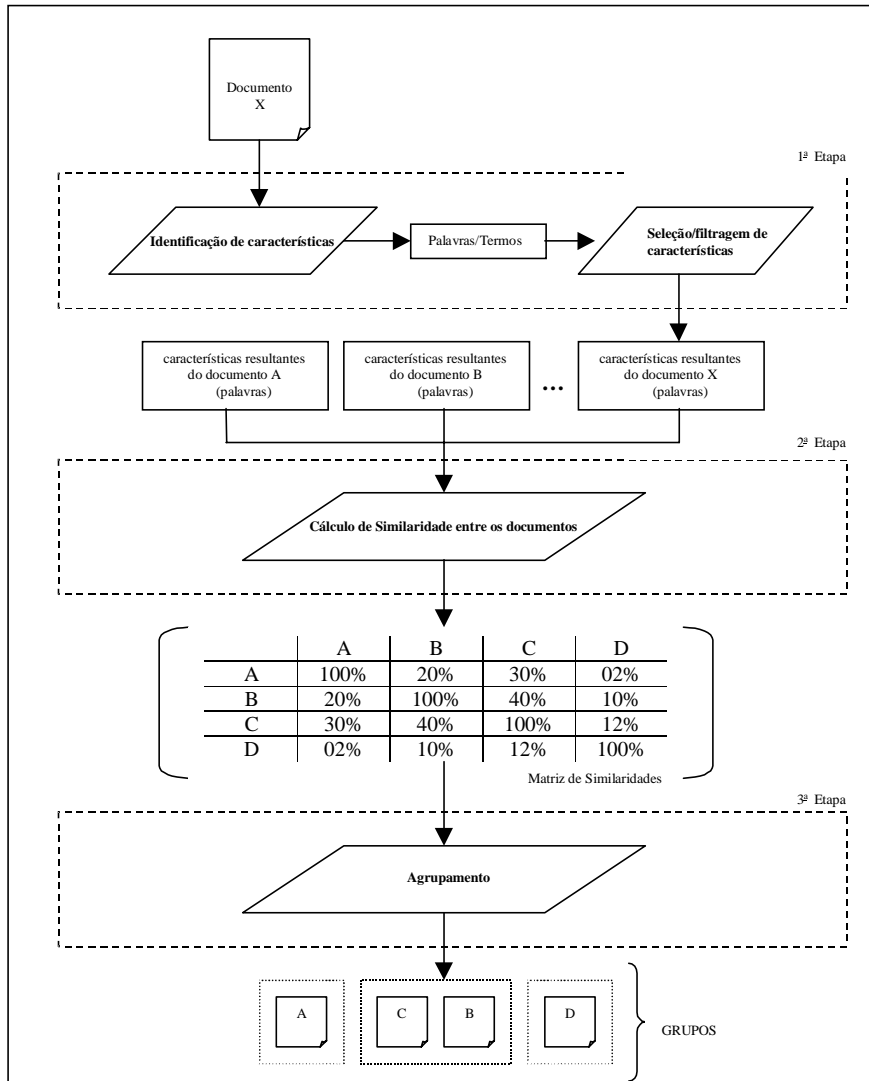


FIGURA 2.5 – Etapas do processo de agrupamento

Todas estas etapas são detalhadas nas próximas seções.

### 2.4.1 Identificação e seleção de características

Segundo *Peter Willet* [WIL 88], os métodos de agrupamento (clustering) estão todos de alguma forma baseados em alguma medida de similaridade entre objetos. Portanto, para que os objetos possam ser agrupados, é necessário identificar as semelhanças existentes entre eles. Mas, para identificar semelhanças entre objetos, é

necessário identificar as características dos objetos e agrupá-los de acordo com a quantidade de características em comum (ou similares) que estes objetos possuem.

Nas técnicas de agrupamento tradicionais, os objetos em questão já possuem atributos (características) bem definidos (em um banco de dados relacional tradicional, os campos das entidades indicam as características).

Nos textos, porém, não é fácil identificar um atributo (campo), simplesmente porque não há algum sinal ou local predeterminado que indique os atributos. Pode ocorrer, ainda, o fato de um atributo ser identificado em um documento e nem ao menos existir em outro (os textos não são obrigados a conter as mesmas palavras). Em decorrência deste fato, as características marcantes dos objetos podem ser variadas e ambíguas (o problema do vocabulário, citado na introdução desta dissertação).

É necessário, em primeiro lugar, estabelecer um método que identifique as características mais marcantes de cada objeto (texto). Em uma informação deste tipo, as palavras é que caracterizam-na. Portanto, nas técnicas de agrupamento de informações textuais as palavras são utilizadas como atributos. Porém, nem todas as palavras podem ser utilizadas. É o caso de palavras inerentes à linguagem, que auxiliam na construção das orações e não precisam ser incluídas no processo (preposições, conjunções e artigos, por exemplo). Estas, e algumas outras palavras, geralmente possuem baixo valor de discriminação, pois aparecem em vários elementos (quase todos) e não ajudam a distinguir o conteúdo dos textos. Palavras deste tipo são chamadas de *palavras negativas* (*stop-words*), e são retiradas em uma etapa de pré-processamento conhecida como *Remoção de palavras negativas*, conforme *Korfhage* [KOR 97] e *Gerard Salton* [SAL 83].

Além disso, em um documento textual, nem todas as palavras aparecem com a mesma frequência. Diante disso, é possível selecionar aquelas palavras (atributos ou características) que são mais marcantes [NG 97, KOR 97]. Isso porque a utilização de todas as palavras do documento torna o processo de agrupamento muito mais demorado, mesmo em algoritmos de *tempo constante* (ver a seguir). Em aplicações de agrupamento reais e práticas os usuários necessitam de um tempo de resposta muito curto. Portanto, o maior *gargalo* para as técnicas de agrupamento de informações está justamente na etapa de seleção de características, que tem como objetivo diminuir o número de características a serem processadas.

A técnica mais comum de identificação de atributos (palavras) marcantes é a *frequência relativa* [SAL 83], que indica o quanto determinada palavra é importante para um documento de acordo com o número de ocorrências desta palavra no documento.

$$F_{rel}x = \frac{F_{abs}x}{N}$$

FIGURA 2.6 – Fórmula da frequência relativa

Na fórmula anterior (FIGURA 2.6), pode-se ver que a *frequência relativa* ( $F_{rel}$ ) de uma palavra  $x$  em um documento qualquer, é calculada dividindo-se sua *frequência absoluta*<sup>2</sup> ( $F_{abs}$ ) pelo número total de palavras no mesmo documento ( $N$ ).

<sup>2</sup> Número de vezes que a palavra aparece no documento.

Existem diversas outras funções, mais discriminativas, que levam em conta o número de documentos em que a palavra aparece [KOR 97]. Porém, conforme Willet [WIL 88] sugere, o resultado das técnicas de agrupamento não é muito influenciado pela função de identificação escolhida, mas sim, pelo algoritmo de agrupamento utilizado. Particularmente, de acordo com os testes realizados na **Seção 5**, conclui-se que o agrupamento é influenciado pela quantidade de atributos selecionados e pelo tipo de atributo selecionado (isto é, se ele é relevante ou não para o documento).

A quantidade de atributos selecionados pode ser um fator independente da fórmula de importância utilizada (até mesmo porque os resultados não variam muito quando se utiliza uma ou outra fórmula). Além disso, fórmulas mais complexas podem ser muito demoradas quando realizam o processo de análise. Deste modo, o ganho que poderiam obter com uma análise mais acurada acaba não sendo vantajoso, pois o agrupamento torna-se muito mais demorado e a diferença nos resultados pode não ser tão significativa. Nos documentos analisados não foi encontrado um estudo que indique uma fórmula de discriminação mais eficiente do que as outras.

Após terem sido identificadas as características dos objetos, é possível aplicar-se uma técnica de *seleção* ou *detecção* de características importantes. Estas técnicas, muitas vezes, são independentes da função de discriminação (cálculo de frequência) escolhida.

*Hinrich Shütze* [SCH 97] apresenta análises sobre a utilização de duas técnicas de detecção de características importantes: a *Truncagem* (*Truncation*) e a *Indexação Semântica Latente* (*Latent Semantic Indexing – LSI*). Ambas projetam o espaço de palavras em um sub-espaço menor, reduzindo o número de termos em cada documento.

A primeira delas, *Truncagem*, é local, ou seja, realizada individualmente para cada documento. E, neste caso, cada documento é projetado em um *sub-espaço* de palavras diferente. Para tanto, ordena-se o vetor (lista) de palavras (características) de cada documento por ordem de importância (a frequência relativa, caso esta tenha sido adotada) e estabelece-se um número máximo de palavras que irão representar cada documento (50 palavras por exemplo). Assim, somente as primeiras  $x$  palavras são utilizadas e as características menos importantes são eliminadas.

É comum aplicar-se uma técnica de pré-processamento, que elimine palavras menos importantes. Pode-se adotar, por exemplo, um valor mínimo de importância, um *limiar* (*threshold*), no qual as palavras (características) com importância (frequência) inferior a este valor são, simplesmente, ignoradas.

A outra técnica de detecção de características apresentada por *Hinrich*, a *LSI*, identifica as palavras utilizadas por todos os documentos e analisa-as globalmente, localizando as palavras menos importantes *antes* que o processo seja realizado. Assim, as palavras identificadas como menos importantes para toda a coleção, são excluídas de todos os documentos (e não apenas em um único documento como no caso anterior). Neste caso, a dimensão de palavras é reduzida e projetada para um único sub-espaço global (todos os documentos passam a pertencer ao mesmo sub-espaço).

*David Lewis* [LEW 91] define *LSI* como sendo uma técnica capaz de transformar a representação de uma coleção de documentos em outra representação que possua propriedades matemáticas desejáveis. Com isso, os termos originais são adaptados à nova representação, cujas características oferecem ganho de performance na recuperação de informações.



Com isso, a *LSI* possui a vantagem de poder comparar qualquer documento, mesmo que eles sejam totalmente diferentes (já que os projeta para uma mesma dimensão). Porém, conforme *Hinrich Shütze* [SCH 97], possui a desvantagem de não se adaptar às características únicas de cada documento. Além disso, *Lewis* [LEW 91] cita que a técnica exige muitos recursos computacionais, tornando o processo mais demorado (já que exige uma pré-análise de todos os termos da coleção de documentos em questão), e não oferece uma melhora de eficiência significativa (ou seja, não oferece melhores resultados).

Além destas, há ainda outras técnicas que levam em conta a estrutura do documento, dando mais importância às palavras que aparecem nos títulos, subtítulos e outras marcações – ver o trabalho de *Jim Cowie* [COW 96]. Porém, a estruturação do documento pode ser considerada uma etapa adicional, e que pode encarecer o processo caso as informações não tenham sido estruturadas no momento de sua elaboração.

Outras técnicas propõem a análise sintática dos documentos a fim de identificar semanticamente (ou morfológicamente) os atributos (palavras) mais importantes de um documento. É o caso de *Peter Anick* [ANI 97], que seleciona somente os substantivos das orações. Neste caso, a não ser que já se possua uma base de conhecimento sintático e semântico, e uma ferramenta para tal, o processo pode ser muito trabalhoso.

Até o momento não foi referenciada alguma técnica de normalização dos termos nos documentos. Adaptar o vocabulário, unificando-o, facilita a identificação de documentos similares. Isso porque existem muitas palavras de grafia diferente que possuem o mesmo significado (sinônimos). Neste caso, trocar as palavras de mesmo significado por uma única facilitaria o processo de agrupamento.

Do mesmo modo, é interessante eliminar as diferentes variações morfológicas de uma palavra (singular ou plural, por exemplo), utilizando somente seu radical. No trabalho de *Hwee Ng* [NG 97] pode ser encontrada uma técnica de pré-processamento que identifica os radicais das palavras, unificando o vocabulário. Porém, algumas vezes, pode tornar um texto muito abrangente, já que muitas palavras tornar-se-ão parecidas.

Todas estas técnicas de processamento são necessárias, porém, como no caso da análise sintática (citado anteriormente), criar ferramentas para tal processamento é demasiadamente caro, já que são necessários dicionários (morfológicos e de sinônimos) específicos da linguagem utilizada nos documentos. Devido a isso, nem todos os trabalhos analisados utilizam-nas.

Na maioria das vezes, assume-se a premissa de que documentos similares (com conteúdos semelhantes) possuem termos comuns. Portanto, mesmo que os termos apareçam poucas vezes em cada um dos documentos, e mesmo que sinônimos sejam usados eventualmente, haverá termos em comum por se tratarem de textos de um mesmo contexto (essa idéia de contexto é detalhada por *Janyce Wiebe* [WIE 96]).

## **2.4.2 Identificação de similaridades entre objetos – funções de similaridade**

A segunda etapa do processo de agrupamento consiste em analisar todos os objetos com o objetivo de identificar semelhanças entre eles. O grau de semelhança entre dois objetos é dado, em geral, por uma fórmula ou função de similaridade. Esta fórmula analisa todas as características (palavras) semelhantes que os objetos possuem e retorna um valor, um grau, indicando o quanto estes objetos são similares.

Novamente, existem várias classes de funções que servem para calcular o quanto um objeto está relacionado com outro. As principais classes são: *medidas de distância* e *medidas difusas (fuzzy)*. Maiores informações sobre estas e outras funções de similaridade podem ser obtidas em diversos trabalhos [SAL 83, CRO 94, KOR 97, KOW 97].

Algumas destas funções são consideradas binárias, pois somente utilizam valores que indicam se determinada característica (palavra) está presente ou não. Neste caso, se uma característica está presente utiliza-se o valor um (1), caso contrário, zero (0). Estes valores são atribuídos independente da palavra aparecer uma ou mais vezes no documento.

Outras, porém, são capazes de utilizar valores que informam o quanto determinada característica influi no objeto, ou seja, o quão discriminativa ela é, e levam em conta o número de aparições da palavra no documento (em alguns casos, levam em conta, também, o número de documentos em que a palavra aparece). Este valor geralmente é normalizado no intervalo [0, 1], onde zero (0) diz respeito a variáveis com nenhuma influência (ou importância) para o objeto, e um (1) significando uma variável na qual o objeto é totalmente dependente (pois o caracteriza significativamente).

### a) Medidas de Distância

As *medidas de distância* geralmente distribuem os objetos em um espaço euclidiano, onde cada ordenada corresponde a uma característica (uma palavra). A similaridade é, então, dada pela proximidade dos objetos neste espaço, conforme a FIGURA 2.7 a seguir.

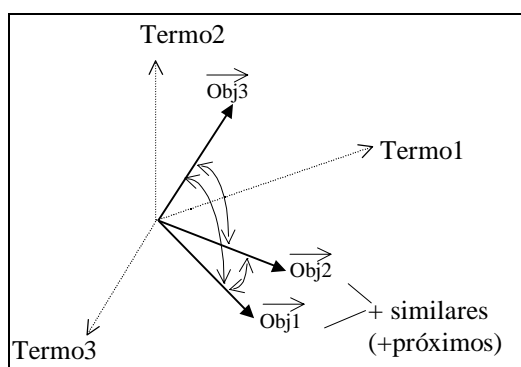


FIGURA 2.7 – Distância euclidiana

A princípio, grande parte das funções de distância costuma utilizar valores binários (palavra existente ou não). Porém, como a frequência das palavras oferece uma relação muito maior, muitas funções antigas foram adaptadas para aceitar valores intermediários, que condizem melhor com a realidade.

A FIGURA 2.8 apresenta um espaço com duas dimensões dadas pelos termos  $x$  e  $y$ . Neste caso, pode-se ver que o documento “A” contém ambas as palavras ( $1x, 1y$ ), e estas são consideradas completamente relevantes na descrição e identificação deste documento. Já o documento B, apesar de possuir ambas as palavras, não é considerado muito importante para nenhuma delas, pois sua relação com elas é de  $(0.4x, 0.65y)$ . O documento C encontra-se na coordenada  $(0.75x, 0y)$ , indicando que  $y$  não está presente no seu conteúdo.

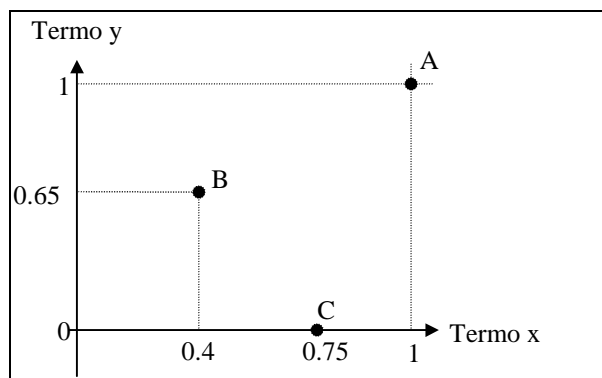


FIGURA 2.8 – Espaço com duas dimensões

Uma das funções de distância euclidiana mais conhecida é a função *coseno* (*cosine*), muito utilizada por *Gerard Salton* no sistema SMART, um SRI. Esta função calcula a distância vetorial euclidiana entre os dois objetos, utilizando valores contidos no intervalo [0, 1].

Pela função *cosine*, a distância (similaridade ou semelhança) entre dois objetos é dada pela fórmula apresentada na FIGURA 2.9, e é baseada na distância euclidiana (FIGURA 2.7).

$$\text{similaridade (Q,D)} = \frac{\sum_{k=1}^n w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^n (w_{qk})^2 \cdot \sum_{k=1}^n (w_{dk})^2}}$$

Onde: **Q** é o vetor de termos do documento *X*,  
**D** é o vetor de termos do documento *Y*  
**w<sub>qk</sub>** são os pesos dos termos do documento *X*, e  
**w<sub>dk</sub>** são os pesos dos termos do documento *Y*.

FIGURA 2.9 – Função de similaridade “cosine”

## b) Medidas Fuzzy

Existem também funções de origem *fuzzy*. Segundo *Henry Oliveira* [OLI 96], o termo *fuzzy* foi introduzido por volta dos anos sessenta em um estudo realizado por *Latfi Zadeh* [ZAD 73]. Neste estudo de *Zadeh* são apresentados os conjuntos difusos, onde se pode utilizar a lógica difusa (*fuzzy*). Pode-se dizer que a lógica *fuzzy* está para o raciocínio aproximado assim como a lógica tradicional está para o raciocínio preciso, conforme *Oliveira* [OLI 96].

Existem várias funções *fuzzy* que podem ser utilizadas. A função *fuzzy* mais simples é a de **inclusão simples** (*set theoretic inclusion*), relatada por vários autores, entre eles *Valerie Cross* [CRO 94], a qual avalia a presença de palavras nos dois elementos sendo comparados. Se o termo aparece nos dois elementos, soma-se o valor um (1) ao contador, caso contrário, zero (0). Ao final, o grau de similaridade é um valor *fuzzy* entre 0 e 1, calculado pela média, ou seja, o valor total do contador de termos

comuns dividido pelo número total de palavras nos dois elementos (sem contar repetidamente).

Este valor *fuzzy* representa o grau no qual um elemento está incluso no outro ou o grau de igualdade entre eles. Porém, há um problema nesta função, pois ela só dá importância para o fato de uma palavra aparecer em ambos os documentos. O fato de uma palavra ser mais importante em um ou outro documento, por aparecerem com frequências diferentes, não é levado em consideração.

Esse problema é resolvido, em parte, por outra função, defendida por *Oliveira* [OLI 96]. Esta função realiza a média por operadores *fuzzy*, sendo semelhante à anterior, porém, utilizando pesos para os termos. Assim, o fato de termos aparecerem com importâncias diferentes nos documentos é levado em consideração.

Neste caso, os pesos dos termos podem ser a frequência relativa ou um valor de discriminação. O valor de similaridade é calculado pela média entre os pesos médios dos termos comuns. Isto é, quando o termo aparece nos dois elementos, soma-se a média dos seus pesos, ao invés de se somar o valor um (1). Ao final, a média é calculada sobre o total de termos nos dois documentos.

Apesar de considerar os pesos dos termos nos documentos, esta última função de similaridade (contando a média nos pesos dos termos) pode trazer distorções nos cálculos. Por exemplo, dois pesos extremos darão o mesmo resultado que dois pesos médios, quando na verdade dois pesos extremos indicam que os termos (apesar de serem comuns nos dois elementos sendo comparados) têm importância diferente em cada documento.

Ao final desta etapa tem-se uma tabela (ver TABELA 2.1) com os valores de similaridade entre todos os objetos (documentos). Normalmente, quando adotadas formulas de medida de similaridade *fuzzy*, os valores apresentados na tabela possuem as seguintes peculiaridades:

- a) Os graus variam entre 0 (sem similaridade) e 1 (totalmente similar);
- b) Um objeto é totalmente similar a ele mesmo;
- c) Se o objeto  $x$  é 20% similar a  $y$ , então  $y$  também é considerado 20% similar a  $x$ .

Com isso, tem-se uma matriz triangular, onde somente os elementos acima da diagonal principal devem ser armazenados e considerados. Isso, facilita o processo de agrupamento, desde que os algoritmos aproveitem-se destas características.

TABELA 2.1 – Matriz de similaridade entre objetos

	<b>OBJ1</b>	<b>OBJ2</b>	<b>OBJ3</b>	<b>OBJ4</b>	<b>OBJ5</b>
<b>OBJ1</b>	1.0	0.3	0.2	0.7	0.6
<b>OBJ2</b>	0.3	1.0	0.5	0.3	0.9
<b>OBJ3</b>	0.2	0.5	1.0	0.3	0.4
<b>OBJ4</b>	0.7	0.3	0.3	1.0	0.8
<b>OBJ5</b>	0.6	0.9	0.4	0.8	1.0

## 2.4.3 Algoritmos de agrupamento

Após concluída a etapa de cálculo de similaridades (ou distância) entre os objetos, tem-se uma matriz que indica os valores de similaridade entre os objetos. A TABELA 2.1 apresenta uma tabela fictícia como exemplo. Esta tabela será utilizada para demonstrar o funcionamento dos algoritmos apresentados a seguir.

A partir desta tabela de similaridades, é possível identificar os grupos de objetos, especificando algum tipo de regra de relacionamento. As estratégias conhecidas como *graph-theoretical*, segundo Salton [SAL 83], utilizam uma notação gráfica a fim de facilitar a identificação de grupos (*clusters*). Neste caso, cada objeto é transformado em um nodo, e as ligações entre estes nodos representam os grupos. Objetos interligados por linhas pertencem a um mesmo grupo.

A FIGURA 2.10 apresenta esta idéia, onde um conjunto de objetos é analisado e seus grupos são identificados de acordo com sua similaridade. No caso, os objetos ligados por linhas pertencem a um mesmo grupo.

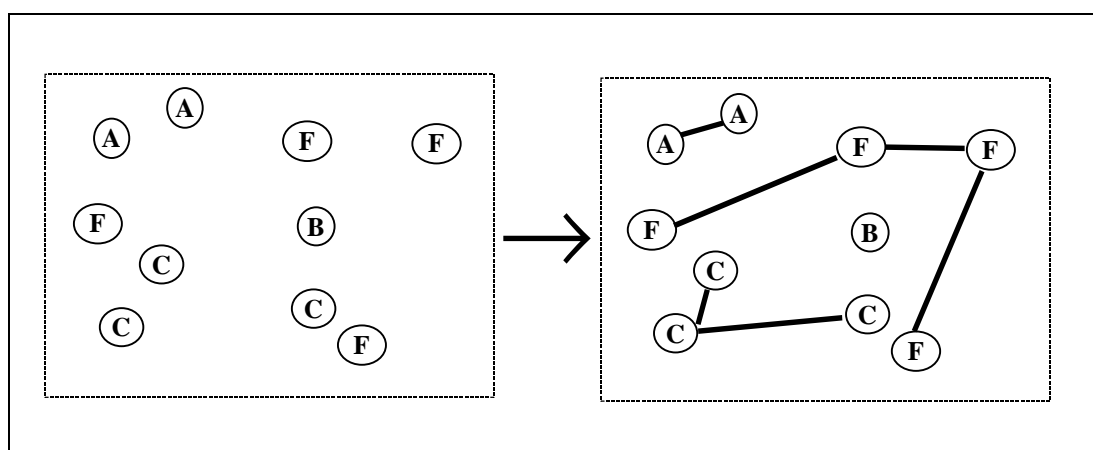


FIGURA 2.10 – Processo de Identificação de grupos

Os algoritmos pertencentes à classe graphic-theoretic, segundo Kowalski [KOW 97], são os seguintes: *Cliques*, *Stars*, *Connected Components* e *Strings*.

### a) Cliques

Exige que todos os objetos do *cluster* sejam similares entre si, com um fator mínimo de aceitação - *threshold*. Na FIGURA 2.11, pode ser visto o resultado de um agrupamento utilizando o algoritmo *cliques* correto e um incorreto. É o algoritmo que produz melhores resultados, porém, é o mais demorado já que todos os objetos têm que ser comparados uns com os outros.

O algoritmo consiste nos seguintes passos:

- 1) Selecionar próximo Objeto e adicioná-lo a um novo cluster;
- 2) Procurar objeto similar;
- 3) Se este objeto for similar a todos os objetos do cluster, adicioná-lo;
- 4) Enquanto houver objetos, voltar ao passo 2;
- 5) Voltar ao passo 1.

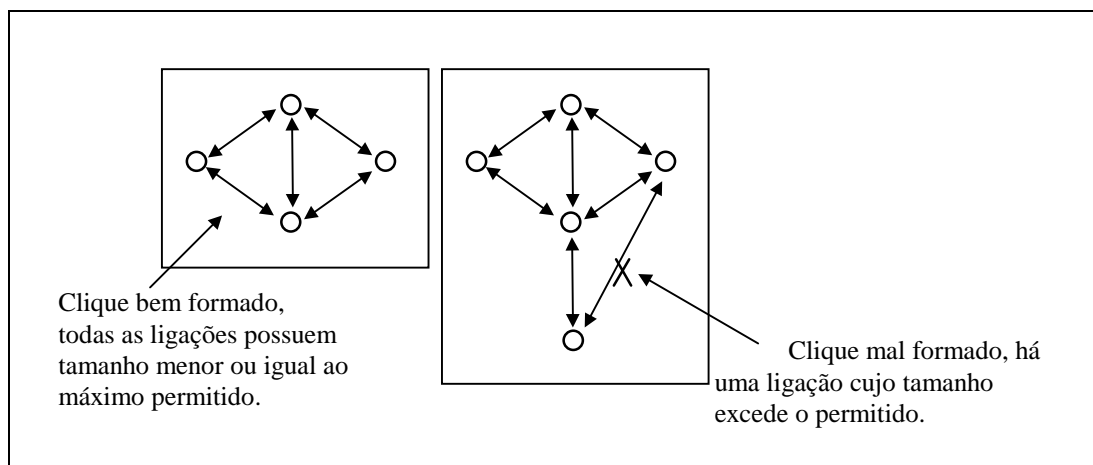


FIGURA 2.11 – Agrupamento pelo método “cliques”

Como exemplo, aplicando-se este algoritmo na TABELA 2.1 chega-se ao seguinte resultado (com  $threshold = 0.5$ ):

Grupo1: OBJ1, OBJ4, OBJ5

Grupo2: OBJ2, OBJ3

Grupo4: OBJ4, OBJ1, OBJ5

Grupo5: OBJ5, OBJ2

Se o usuário desejar este algoritmo pode ser disjuncto ou não (onde um elemento pode ser atribuído a mais de uma classe).

### b) Connected components (single link)

Neste caso, qualquer objeto que seja similar a outro de uma classe (não necessitando ser similar a todos, como no caso anterior) é adicionado à classe. Consiste em:

- 1) Selecionar o primeiro objeto sem classe e colocá-lo em uma nova classe;
- 2) Colocar na mesma classe todos os objetos similares;
- 3) Para cada um dos objetos adicionados à classe, realizar o passo 2;
- 4) Quando não forem identificados novos termos no passo 2, ir para passo 1.

Exemplo, utilizando  $threshold = 0.5$  e a TABELA 2.1:

Classe1: OBJ1, OBJ4, OBJ5, OBJ2

Classe2: OBJ3

Nota-se que não é possível que um objeto pertença a mais de um grupo. É considerado um algoritmo mais rápido do que o *Cliques*, por ser mais relaxado. Decorrente disso, produz grupos mais simples.

### c) Stars

Este algoritmo consiste em selecionar um elemento e identificar todos os elementos conectados a ele. Deste modo, tem-se uma figura muito parecida com uma estrela (daí o nome: *star* ou *estrela*), pois um item central conecta todos os outros componentes do grupo.

- 1) Selecionar 1 termo e colocar todos os similares na mesma classe;
- 2) Termos ainda não classificados são colocados como semente de classe (passo 1).

Aplicando-se este algoritmo à TABELA 2.1, chega-se ao seguinte resultado (com *threshold* = 0.5):

Classe1: OBJ1, OBJ4, OBJ5

Classe2: OBJ2, OBJ3

### d) Strings

A idéia deste algoritmo é construir vetores de objetos similares, como em um vetor de caracteres (String), onde o objeto *A* está conectado ao objeto *B*, este ao objeto *C*, e, assim, sucessivamente até que não existam mais conexões.

O método consiste em criar uma classe com o primeiro objeto. Após, é necessário localizar o próximo objeto similar e adicioná-lo na mesma classe. Selecionar o novo objeto e localizar o objeto mais similar a ele. Repetir o processo recursivamente. Passos:

- 1) Selecionar um objeto não classificado, adicionar em uma nova classe e selecioná-lo como NODO;
- 2) Localizar um objeto similar ao NODO e adicioná-lo na mesma classe;
- 3) O novo objeto passa a ser o NODO, repetir passo 2;
- 4) Se não houver objetos similares, voltar ao passo 1.

Aplicando-se este algoritmo à TABELA 2.1, chega-se ao seguinte resultado (com *threshold* = 0.6):

Classe1: OBJ1, OBJ4, OBJ5, OBJ2

Classe2: OBJ3

### e) Algoritmos heurísticos

Nos algoritmos anteriores o tempo de agrupamento é variável, dependente da quantidade de elementos existentes. Com isso, quanto mais elementos existirem, mais demorados eles se tornam. Isso se torna um problema quando o processo de agrupamento é utilizado em aplicações que necessitem de um tempo de resposta mais rápido ou imediato.

Em aplicações *on-line*, dificilmente os usuários teriam paciência de esperar um resultado cujo processamento demorasse mais do que alguns minutos. Um exemplo de aplicação de agrupamento, *on-line*, seria um sistema de agrupamento de páginas WEB por similaridade<sup>3</sup>. Este sistema, além de *on-line*, é *on-demand*, pois suas ações dependem do usuário para guiá-lo durante o processo (interativo). Ou seja, é executado quando o usuário necessita dele.

Atualmente, existem estudos sendo realizados a fim de criar algoritmos heurísticos que diminuam o tempo de agrupamento. Algoritmos como o "*k-means*" [BER 97] são capazes de processar mais rapidamente os grupos, porém, ainda não são ideais para aplicações reais e cotidianas.

O algoritmo *k-means* é um algoritmo muito conhecido e utilizado, o que não significa que ele seja o melhor. Seu processo consiste em estabelecer, *a priori*, um número máximo de grupos que se deseja obter (*k* grupos, por exemplo). O algoritmo busca então a melhor maneira de separar os documentos nestes *k* grupos.

É claro que este é um processo complexo, muitas vezes amenizado pelas técnicas que implementam este algoritmo. Estas técnicas, com o objetivo de diminuir o tempo necessário para o processo, acabam não obtendo a melhor separação (os melhores grupos), pois a iteração pode terminar antes que a melhor separação seja encontrada.

Outro algoritmo similar é citado por *Douglass Cutting* [CUT 92]: *Buckshot*. O *buckshot* também utiliza uma estratégia de partição fixa, onde o usuário estabelece um número fixo de grupos a serem gerados e o algoritmo distribui os objetos de acordo com este número. Nas experiências realizadas por *Cutting* [CUT 92, CUT 93], o processo é aplicado recursivamente e os *k* grupos encontrados são redistribuídos em *k* grupos, construindo assim uma estrutura hierárquica.

O maior problema destes algoritmos é a imposição dos *k* grupos. Isso porque o usuário não tem como saber qual valor de *k* é ideal para distribuir os objetos, ou seja, quantos clusters são necessários para representar a organização natural dos elementos. No livro de *Michael Berry* [BER 97] há a seguinte afirmação:

**Se as variáveis utilizadas para descrever os objetos são independentes, nenhum cluster é encontrado. Porém, ao extremo oposto, se todas as variáveis são dependentes na mesma coisa (se são co-lineares), então todos os objetos devem fazer parte de um único cluster. Entre estes extremos não há como saber quantos clusters encontrar.**

Isso significa que o número de clusters (valor de *k*) é extremamente dependente do domínio (coleção de documentos) sendo processada. Se o usuário escolhe um valor arbitrário para *k* (o realmente acontece) a coleção é distribuída "à força" nestes *k* grupos.

Assim, não é possível saber se este número de clusters é realmente o ideal para o conjunto de documentos tratado. Com isso, a separação pode não levar em conta a organização natural dos documentos, já que eles acabam sendo modelados, abstraídos a um subconjunto fixo de espaços (assuntos).

---

<sup>3</sup> Como exemplo deste processo interativo *on-line* pode ser citado, novamente, o refinamento da ferramenta *Altavista*<sup>TM</sup>. Este refinamento apresenta graficamente os relacionamentos entre palavras identificadas nas páginas retornadas pela consulta do usuário.



Nestes casos, como solução, o usuário deve tentar outros valores, repetidamente, até encontrar a separação que mais lhe agrada. Os próprios algoritmos podem realizar testes, avaliando de alguma forma os resultados, e repetir o agrupamento, automaticamente, até que o melhor resultado seja encontrado.

## 2.4.4 Análise dos algoritmos

Para que o agrupamento possa ser realizado com os fins de recuperação de informações ou de descoberta de conhecimento, é necessário que os grupos constituídos pelos algoritmos tenham uma certa coesão entre seus objetos. Grupos com objetos muito diferentes não fariam sentido. O problema é que alguns dos algoritmos estudados não são tão restritivos quanto se espera, pois permitem que objetos com pouco grau de similaridade (em relação a todos os outros objetos do grupo) participem de um grupo só porque possuem forte relação com um dos objetos.

A seguir, são apresentadas análises dos algoritmos estudados, indicando se o algoritmo deve ou não ser adotado neste trabalho (de acordo com os fins de recuperação de informações ou descoberta de informações, definidos anteriormente). Em alguns casos, onde os algoritmos apresentam problemas mais simples, soluções alternativas são propostas. Em outros, onde os problemas são muito complicados, detalha-se o porquê deles não terem sido adotados.

### a) Strings

O algoritmo *Strings* não foi adotado porque não produz grupos de objetos coesos. Isso porque, verificando o funcionamento do algoritmo na matriz de similaridades, descobre-se que só é possível garantir o grau de similaridade mínima, e este grau, em um grupo de vários elementos, pode ser muito pequeno ou nem sequer existir. Um exemplo prático, utilizando a TABELA 2.2, é apresentado a seguir.

TABELA 2.2 – Matriz de similaridade entre objetos

	<b>OBJ1</b>	<b>OBJ2</b>	<b>OBJ3</b>	<b>OBJ4</b>	<b>OBJ5</b>
<b>OBJ1</b>	<b>1.0</b>	<b>0.6</b>	<b>0.2</b>	<b>0.7</b>	<b>0.4</b>
<b>OBJ2</b>	0.6	<b>1.0</b>	<b>0.6</b>	<b>0.3</b>	<b>0.9</b>
<b>OBJ3</b>	0.2	0.6	<b>1.0</b>	<b>0.7</b>	<b>0.4</b>
<b>OBJ4</b>	0.7	0.3	0.7	<b>1.0</b>	<b>0.8</b>
<b>OBJ5</b>	0.6	0.9	0.4	0.8	<b>1.0</b>

Como já citado anteriormente, o método consiste em selecionar o primeiro objeto da matriz e criar um grupo para ele. Após, o primeiro elemento similar a ele, ou seja, o elemento que possui, pelo menos, o grau mínimo de similaridade (limiar) estabelecido pelo usuário, é adicionado ao grupo. No caso que utiliza a matriz da TABELA 2.2, anterior, e com limiar de 0.6, o OBJ1 seria designado a o primeiro grupo. Em seguida, o OBJ2 seria adicionado ao mesmo grupo (primeiro), por possuir grau de similaridade (com o OBJ1) igual a 0.6.

O próximo passo é encontrar algum elemento similar ao último elemento adicionado (e não ao primeiro). Neste momento, esquece-se o OBJ1 e procura-se um objeto similar ao OBJ2 (o último adicionado). Com isso, o OBJ3 seria adicionado ao mesmo grupo, por possuir similaridade de 0.6 com o OBJ2. O processo é repetido até que não haja elementos com similaridade maior do que a mínima.

O resultado da aplicação deste algoritmo na tabela anterior seria equivalente ao apresentado na FIGURA 2.12.

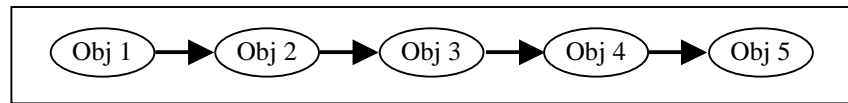


FIGURA 2.12 – Resultado do agrupamento pelo método “strings”

Neste caso, haveria um único grupo, composto de todos os elementos. Porém, nota-se que, apesar dos elementos possuírem graus de similaridade satisfatórios com seus vizinhos, não há um grau de similaridade satisfatório entre o primeiro e o último elemento.

Na FIGURA 2.13, mesmo que os objetos tenham um grau de similaridade mínimo de 90% com seus vizinhos, o grau de similaridade mínimo (supondo-se que a diferença entre os elementos seja de 10%, e que esta se agregue de elemento para elemento) entre o primeiro e o último seria de 50%. Com isso, quanto maior a *cadeia* (*string*), maior pode ser a diferença entre os objetos que estão em posições opostas da *string*.

É claro que quanto menor a *string*, maior é a chance dos documentos serem similares. O grau apresentado na figura acima diz respeito ao grau mínimo, mas é possível que este grau seja ainda maior. Porém, como não há como garantir esse valor, e esta diferença pode variar muito, o grupo pode tornar-se pouco coeso, sem muita utilidade prática para fins de descoberta de conhecimento, identificação de padrões e até mesmo recuperação de informações.

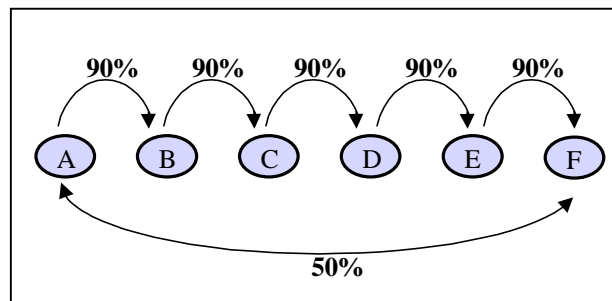


FIGURA 2.13 – Problema do agrupamento pelo método “strings”

### b) Connected Components (single link)

Este método não foi adotado porque, na verdade, consiste em uma estrela (*star*) que vai sendo agregada por outras estrelas. Com isso, vão formando-se cadeias de estrelas que possuem um certo grau de similaridade entre os componentes que ligam as estrelas. Porém, estas estrelas não mantêm, necessariamente, relação com as outras estrelas da cadeia, tornando o grupo pouco coeso (a FIGURA 2.14 ilustra este problema).

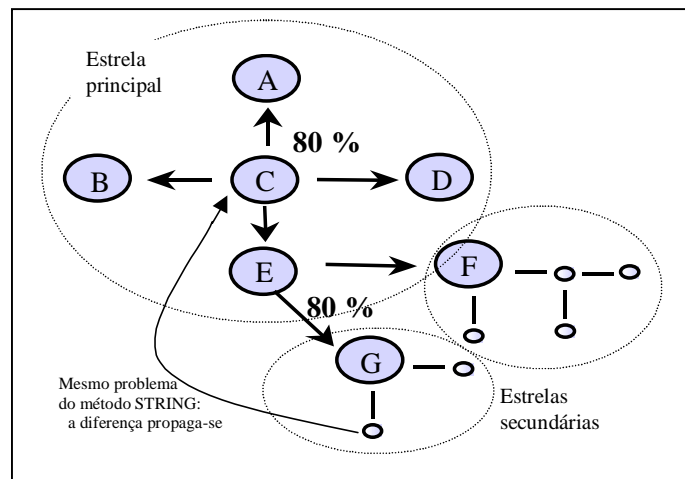


FIGURA 2.14 – Resultado fictício de um agrupamento pelo método “single link”

### c) Cliques

Segundo *Korfhage* [KOR 97] o algoritmo *Cliques*, dentre os apresentados, é o que oferece um melhor resultado, pois gera grupos de elementos muito coesos (se o grau de similaridade mínimo for elevado). Isso porque para cada elemento adicionado ao grupo, verifica-se seu grau de similaridade com todos os outros elementos do mesmo grupo. Caso este elemento não tenha grau de similaridade maior do que o mínimo, com um dos objetos, ele não é adicionado ao grupo. Porém, devido a esta verificação, este algoritmo torna-se mais demorado.

### d) Stars

Este algoritmo foi selecionado pois, além de oferecer uma boa coesão, é muito rápido, já que não necessita realizar análises complementares como no algoritmo *Cliques*. Além disso, é possível saber a similaridade mínima entre os elementos do grupo, e esta diferença não é propagada como no algoritmo *Strings*.

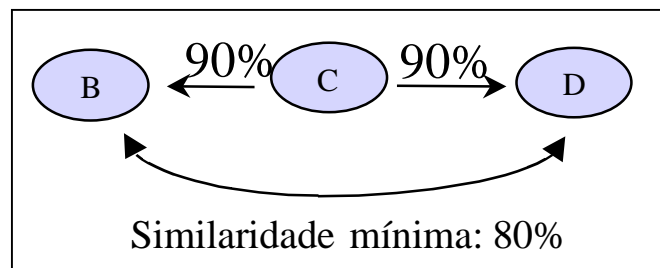


FIGURA 2.15 – Similaridade mínima no algoritmo “stars”

Porém, ele apresenta dois problemas. O primeiro diz respeito ao fato dos objetos serem atribuídos ao primeiro grupo cujo grau de similaridade seja maior do que o mínimo exigido. E, já que os objetos atribuídos a um grupo não são mais processados, não há garantia de que um objeto tenha sido colocado no grupo de maior afinidade (similaridade). Além disso, devido a isso, caso a ordem dos elementos na matriz de similaridades seja trocada, o resultado do agrupamento pode variar.

Outro problema diz respeito ao fato do processo não indicar todos os grupos que o objeto poderia fazer parte. Em documentos é possível que existam textos que tratem de mais de um assunto. No algoritmo *stars* o documento seria atribuído ao primeiro assunto que satisfizesse a restrição do grau mínimo de similaridade (portanto o algoritmo é disjunto).

Visando solucionar estes problemas, foram elaborados dois algoritmos: o *best-star* e o *full-stars*, descritos a seguir.

### e) Best star

O algoritmo *best-star* (melhor estrela) foi desenvolvido com o intuito de solucionar o problema do algoritmo *stars* que atribui um objeto ao primeiro grupo cujo grau de similaridade satisfaça a restrição do grau mínimo. Isso porque podem existir grupos onde o objeto teria um grau de afinidade maior, e, portanto, mais adequado. No caso, os objetos identificados como sendo mais similares a determinado grupo, mesmo que já agrupados, são reorganizados e atribuídos ao grupo de maior afinidade.

Neste caso, o documento que possua relação com mais de um assunto, seria colocado no assunto (grupo) com que possuir maior afinidade (relação). Outra vantagem deste algoritmo diz respeito ao fato de encontrar uma relação mais natural entre os objetos, colocando-os em grupos mais coerentes com a realidade (porque os objetos são colocados automaticamente no grupo de objetos que ele possui relação mais forte), sem que o usuário precise preocupar-se em estabelecer um grau de similaridade mínima adequado. Os testes apresentados na **Seção 5 – Estudos de caso** – explicam e detalham melhor este fato.

### f) Full stars

Todos os outros algoritmos adotados são considerados *disjuntos*, ou seja, atribuem elementos a um único grupo. Isso porque eles não indicam todas as relações (todos os grupos) que determinado documento pode ter. Neste caso, não há como o usuário saber quais são todos os assuntos que determinado documento se encontra.

É claro que eles poderiam ser adaptados de modo que todo o elemento, mesmo que já processado, pudesse ser alojado em outro grupo. O algoritmo *full-stars* desenvolvido faz justamente isso, ou seja, atribui os documentos a todos os grupos cujo grau de similaridade ultrapassa o valor mínimo estabelecido. Seu funcionamento é idêntico ao funcionamento do algoritmo *stars*, porém, com a diferença de que os documentos já atribuídos continuam no processo, não são ignorados.

### 3 Agrupamento textual proposto

Para poder realizar o agrupamento de informações textuais, aplicou-se a metodologia de agrupamento de informações detalhada na seção anterior. Todas as etapas descritas (seleção de características, cálculo de similaridades e agrupamento) foram modeladas e refinadas. Com isso, foram identificados alguns problemas e propostas alternativas como solução.

A técnica de agrupamento proposta consiste nas seguintes etapas:

- a) Identificação de palavras;
- b) Remoção de palavras negativas (stopwords);
- c) Cálculo de frequência relativa;
- d) Remoção de palavras com frequência inferior a um limiar estabelecido pelo usuário. Esta frequência mínima pode ser ignorada pelo usuário;
- e) Utilização das  $x$  palavras mais frequentes – método de *truncagem* (onde  $x$  é estabelecido pelo usuário, podendo ser todas as palavras);
- f) Utilização de uma fórmula de similaridade *fuzzy* definida especialmente para o processo (uma das contribuições do trabalho);
- g) Aplicação de um algoritmo de agrupamento, inclusive sendo apresentados dois novos algoritmos.

Nota-se que em nenhum momento realiza-se alguma etapa de pré-processamento de linguagem natural (citadas na seção 2.4.1). Portanto, considera-se que os documentos já tenham sido corrigidos por um corretor ortográfico. Além disso, é interessante que os textos tenham sofrido algum tipo de processamento de normalização de termos. Porém, estas etapas não são obrigatórias. Não foi realizado nenhum estudo comprovando que sua utilização torna o processo mais eficiente.

A única exigência do processo diz respeito ao formato do texto. Todos os documentos necessitam estar no formato ASCII padrão, sem caracteres de controle adicionais ou comandos. Aconselha-se que, pelo menos, os textos sejam corrigidos em relação a sua ortografia.

#### 3.1 Identificação de palavras

A técnica de identificação de palavras adotada é muito simples. Foram definidos dois tipos de dados básicos: palavras e números. Em primeiro lugar, todos os caracteres são convertidos para seu respectivo código ASCII maiúsculo. Depois disso, as palavras são reconhecidas. Neste caso, palavra é toda seqüência de caracteres (*string*) que possui um ou mais dos seguintes caracteres: “abcdefghijklmnopqrstuvwxyzàáéíóúâêôûäëïöüç”.

Especificação formal da expressão:

$$P \rightarrow S1 S2$$

$$S1 \rightarrow SS1|SS2$$

$$SS1 \rightarrow a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|x|y|z$$

$$SS2 \rightarrow \grave{a}|\acute{a}|\acute{e}|\acute{i}|\acute{o}|\acute{u}|\grave{a}|\hat{e}|\hat{i}|\hat{o}|\hat{u}|\tilde{a}|\tilde{e}|\tilde{i}|\tilde{o}|\tilde{u}|\c$$

$$S2 \rightarrow S1|S2|S3|\phi$$

$$S3 \rightarrow 1|2|3|4|5|6|7|8|9|0$$

De acordo com a expressão acima, também é possível utilizar uma palavra que contenha números em posições intermediárias da seqüência, desde que não seja a primeira posição.

São considerados números as seqüências de caracteres que não contenham os caracteres anteriores e contenham somente números (caracteres “1234567890”) e, eventualmente, entre estes números, possuam o caracter “.” (ponto) ou “,” (vírgula).

Especificação formal da expressão:

$$N \rightarrow S1 S2$$

$$S1 \rightarrow 1|2|3|4|5|6|7|8|9|0$$

$$S2 \rightarrow S2|S1| S3 S1|\phi$$

$$S3 \rightarrow .|,$$

Todos os demais caracteres e seqüências são ignorados. Os delimitadores de palavras, são, então, qualquer caracter diferente dos caracteres identificados.

Os números e palavras identificados são colocados em uma lista de atributos (palavras) relevantes ao objeto (texto).

## 3.2 Remoção de palavras negativas

Como dito anteriormente, existem palavras inerentes à linguagem utilizada ou até mesmo inerentes ao contexto sendo tratado na coleção de documentos sendo processada. Nestes casos, é comum excluir estas palavras (chamadas de palavras negativas ou *stopwords*) pois elas influenciam no processo, além de torná-lo mais demorado, já que com elas há um número maior de características para serem analisadas.

No processo adotado, o usuário deve indicar quais são as palavras negativas que devem ser excluídas do processo. Deste modo, já que coleções de documentos podem exigir listas de palavras negativas específicas, o processo torna-se mais adaptável a diferentes domínios e linguagens.

As palavras identificadas como negativas pelo usuário são excluídas da lista de atributos de cada objeto, quando encontradas, reduzindo o número de características a serem utilizadas no processo de agrupamento.

### 3.3 Cálculo de frequência relativa

Até esta etapa todas as palavras do documento são consideradas características importantes (com exceção das palavras negativas). Porém, de alguma forma, é necessário identificar quais características caracterizam melhor cada objeto. A frequência relativa, detalhada no capítulo anterior, é uma das formas utilizadas para identificar o quanto cada característica é importante em cada objeto.

Ela é adotada porque, além de ser simples, *Willet* [WIL 88] comenta que o resultado das técnicas de agrupamento não é muito influenciado pela função escolhida, mas sim, pelo algoritmo de agrupamento utilizado.

A frequência relativa baseia-se na frequência absoluta, que consiste em contar o número de vezes que determinada palavra aparece em um documento. Após, a frequência absoluta é dividida pelo número total de características do documento, dando o grau de afinidade da característica com o documento, levando em conta também as outras características do documento (ou seja, indica o grau de porcentagem ou influência da característica no documento em questão).

### 3.4 Determinação de frequência mínima

A fim de tornar o processo mais rápido, o usuário pode indicar o número mínimo de vezes que as palavras devem aparecer para que façam parte do processo de agrupamento (para que sejam consideradas características relevantes).

Isso é feito para diminuir o número de características utilizadas no processo, já que quanto maior o número de características a ser comparado, mais demorado é o processo. Isso pode ser feito, em alguns casos, sem que haja uma perda muito grande de qualidade. Em experimentos realizados por *Schütze* [SCH 97], a utilização de cinquenta (50) características produziu resultados muito similares aos obtidos com a utilização de todas as características.

A teoria por trás do processo é de que as palavras com pouca frequência não são relevantes para o documento, podendo ser ignoradas.

### 3.5 Seleção de características

Adotou-se como seleção de termos a técnica de *truncagem*, já que, além de ser implementada mais facilmente, não influi negativamente nos resultados, conforme testes realizados por *Schütze* [SCH 97], além de oferecer um ganho de performance no algoritmo. Além disso, *Schütze* constata que a *LSI* mostra-se mais efetiva para o agrupamento de termos, no contexto de recuperação de informações, não sendo o objetivo principal deste trabalho.

Na *truncagem*, estabelece-se um número máximo de características a serem utilizadas para caracterizar cada objeto. Para tanto, é necessário que as características estejam ordenadas de acordo com seu grau de relevância (ou importância). Assim, somente as primeiras  $x$  características são utilizadas. Aqui, a mesma hipótese da etapa anterior é utilizada: *palavras com pouca frequência não caracterizam fortemente o objeto, e são consideradas irrelevantes*.

É claro que estabelecer o grau mínimo ou o número mínimo de características relevantes é um processo complicado e difícil, que pode até mesmo variar de coleção para coleção. Porém, conforme já citado, os experimentos de *Schütze* [SCH 97] indicam que na grande maioria dos casos um número de 50 características oferece um resultado satisfatório.

### 3.6 Cálculo de similaridades

Depois de identificadas as características relevantes de cada objeto, parte-se para a análise de similaridade entre estes objetos. Todo o processo de agrupamento está baseado em algum tipo de similaridade entre os objetos, pois os agrupa (ou separa-os) em grupos de objetos que possuam alguma semelhança (similaridade) entre si.

Esta é a etapa mais crucial do processo, e sua eficiência depende muito das características identificadas como relevantes. Caso as etapas anteriores não tenham identificado características realmente relevantes todo o processo pode ser comprometido, já que o resultado pode não ser o ideal para a coleção de documentos sendo processada.

É possível afirmar que quanto maior o número de características utilizadas nesta etapa do processo, mais confiável é o grau de similaridade entre os objetos. Como em todo o processo de abstração, quanto menos se abstrai do mundo real mais condizente com a realidade consegue-se ser (e melhor é a descrição deste mundo, pois ela é mais real). Quanto mais se abstrai de algum objeto, menos características podem ser informadas (ou armazenadas), e pior é a qualidade de representação. Isso é válido para o agrupamento.

Infelizmente, o tempo necessário para obter um agrupamento de qualidade é diretamente proporcional ao número de características utilizadas. Não há como obter um desempenho muito rápido com qualidade, a não ser que as características escolhidas como relevantes sejam realmente as que mais transmitem informações do objeto. Por isso, as etapas anteriores são críticas e devem trabalhadas com cuidado pois, ao menos na literatura consultada, não há um algoritmo de detecção de características que funcione adequadamente em qualquer contexto.

Novamente, existem várias fórmulas ou funções capazes de identificar a similaridade entre objetos. Neste trabalho deter-se-á em um único tipo, bastante utilizado atualmente: as funções *fuzzy*. No trabalho de *Oliveira* [OLI 96] há um estudo bem detalhado sobre funções *fuzzy*. Baseando-se no estudo de *Oliveira* é possível definir uma fórmula para cálculo de similaridades que leve em conta as diferenças e as semelhanças de cada documento, utilizando operadores *fuzzy* adequados para cada situação. Todo o referencial teórico referente à função definida pode ser encontrado no referido estudo, não sendo necessário validar sua eficiência e aplicação neste trabalho.

A função apresentada por *Oliveira* [OLI 96] é denominada *média por operadores fuzzy*. Baseando-se nesta idéia, foi criada a função para o cálculo de similaridades entre documentos, apresentada na Figura 3.1, seguinte.



$$gs(X, Y) = \frac{\sum_{h=1}^k gi_h(a, b)}{n}$$

onde:

**gs** é o grau de similaridade entre os documentos **X** e **Y**;

**gi** é o grau de igualdade entre os pesos do termo **h** (peso **a** no documento **X** e peso **b** no documento **Y**);

**h** é um índice para os termos comuns aos dois documentos;

**k** é o número total de termos comuns aos dois documentos;

**n** é o número total de termos nos dois documentos (sem contagem repetida).

FIGURA 3.1 – Fórmula da média por operadores “fuzzy”

Basicamente, a função apresentada na FIGURA 3.1, anterior, utiliza um contador que vai acumulando pontos toda vez que um termo é encontrado em ambos os documentos sendo comparados. O valor utilizado para atualizar este contador é dado por uma outra fórmula, definida por [PED 93], que identifica o grau de igualdade entre os termos comuns.

Esta outra fórmula é necessária, porque, apesar da palavra aparecer em ambos os documentos, ela pode ter graus de importância diferentes em cada documento. Esta outra função, apresentada a seguir, leva em conta a frequência relativa do termo em ambos os documentos (na realidade, sua média). Os termos que não aparecem em nenhum dos dois documentos sendo comparados contribuem com o valor zero.

$$gi(a, b) = \frac{1}{2} [(a \rightarrow b) \wedge (b \rightarrow a) + (\bar{a} \rightarrow \bar{b}) \wedge (\bar{b} \rightarrow \bar{a})]$$

onde,

$$\bar{x} = 1 - x$$

$$a \rightarrow b = \max\{c \in [0,1] \mid atc \leq b\}, t = \text{produto}$$

$$\wedge = \min$$

FIGURA 3.2 – Fórmula do cálculo do grau de igualdade entre pesos

Assim, se um termo aparece em ambos os textos, porém, com pesos muito diferentes, o grau de igualdade torna-se baixo. Por outro lado, quando os valores dos pesos são próximos, o grau de igualdade torna-se mais alto. Caso um termo apareça somente em um dos documentos, a similaridade diminui (já que este termo contribui com um grau zero). Com isso, obtém-se um valor mais realista para ser considerado na média (similaridade) final.

Ao final desta etapa tem-se a matriz de similaridade triangular (uma propriedade desta fórmula), onde os valores variam no intervalo [0,1]. Um grau zero (0) indica documentos totalmente díspares; já um grau um (1) indica documentos similares. Além

disso, um documento é igual a ele mesmo, recebendo, portanto, grau um (1). Outro fator importante: a matriz é simétrica. Com isso, se um elemento  $A$  possui um grau de similaridade  $x$  com o elemento  $B$ , então, o elemento  $B$  também possui o mesmo grau de similaridade com  $A$ .

Há, porém, uma advertência: documentos são considerados similares por possuir palavras similares. Um grau de similaridade de 100% (1) não significa que os documentos sejam iguais, mas sim, que possuem as mesmas palavras (excluindo as palavras negativas, as palavras infreqüentes e as demais palavras ignoradas no processo de agrupamento). Estas palavras, mesmo que comuns, não são obrigadas a estar na mesma ordem (de aparição) em ambos os documentos.

### **3.7 Agrupamento**

No capítulo anterior, viu-se que o agrupamento em si consiste em definir algum tipo de restrição que será aplicada na matriz de similaridades, gerada na etapa anterior. Cada algoritmo possui um tipo de restrição diferente, conforme descrito na seção anterior. Com isso, os objetos (documentos) são então separados em grupos que satisfazem estas restrições.

## 4 Implementação

Para fins de análise e comparação dos algoritmos estudados foi realizada a implementação de um protótipo de ferramenta de agrupamento de informações (objetos) textuais. A ferramenta possui várias opções e parâmetros que podem ser definidos pelo usuário, além de apresentar diversas formas de análise dos resultados. Porém, por ser um protótipo, possui algumas limitações que devem ser ajustadas e refinadas em algum trabalho futuro.

A ferramenta foi batizada de *Eurekha*, já que permite que o usuário obtenha conhecimento (padrões, relacionamentos) nos textos de forma interativa. Foi implementada utilizando a linguagem C++, com algumas características de orientação a objetos. O ambiente de programação adotado é o *Borland CBuilder 3.0*, devido às suas facilidades de construção de interfaces.

Adotou-se a plataforma Windows95 ou WindowsNT como plataforma alvo da implementação. Como requisitos mínimos do sistema são necessários:

- a) um ambiente ou sistema operacional gráfico padrão Windows 32 bits;
- b) um processador Pentium™ ou compatível;
- c) 16 Megabytes de memória;
- d) 4 Megabytes disponíveis no disco rígido.

Porém, com objetivos de aumentar a eficiência (o tempo de processamento), recomenda-se 64Mb de memória RAM disponível. O espaço em disco depende do tamanho e número de documentos utilizados no processo.

Optou-se pela adoção do idioma Inglês para a construção da interface, já que as coleções de teste são, em sua grande maioria, elaboradas neste idioma. Além disso, os resultados de outros estudos são utilizando estas coleções também são publicados em Inglês. Portanto, para conseguir uma abrangência maior, publicando resultados em revistas e congressos internacionais (com exemplos de utilização da ferramenta), este idioma foi utilizado.

### 4.1 Interface

A interface da ferramenta é constituída de 3 tabelas ou *orelhas*, cada uma correspondendo a uma etapa do processo:

- a) *Stopwords* – Definição e manipulação de palavras negativas;
- b) *Collections* – Definição e manipulação de coleções de documentos, definição dos atributos para o processo de agrupamento e início do processo de agrupamento (cálculo de similaridades e geração de matriz de similaridades);
- c) *Clusters* – Permite agrupar as coleções cujas matrizes de similaridades já tenham sido calculadas na etapa anterior. Apresenta os algoritmos de agrupamento disponíveis na ferramenta, que podem ser selecionados pelo usuário, e permite que os documentos da coleção selecionada sejam agrupados conforme o algoritmo selecionado.

### 4.1.1 Tabela de “stopwords”

Na primeira das orelhas (FIGURA 4.1) o usuário manipula as classes de palavras negativas (stopwords) que devem ser ignoradas no processo de agrupamento. Nesta tabela (orelha) o usuário pode definir tipos (grupos) de palavras negativas (stopwords).

Na região esquerda, identificada pelo número um (1), encontra-se a lista de tipos de palavras negativas. Na região direita, dois (2), encontra-se a lista de palavras negativas do tipo selecionado.

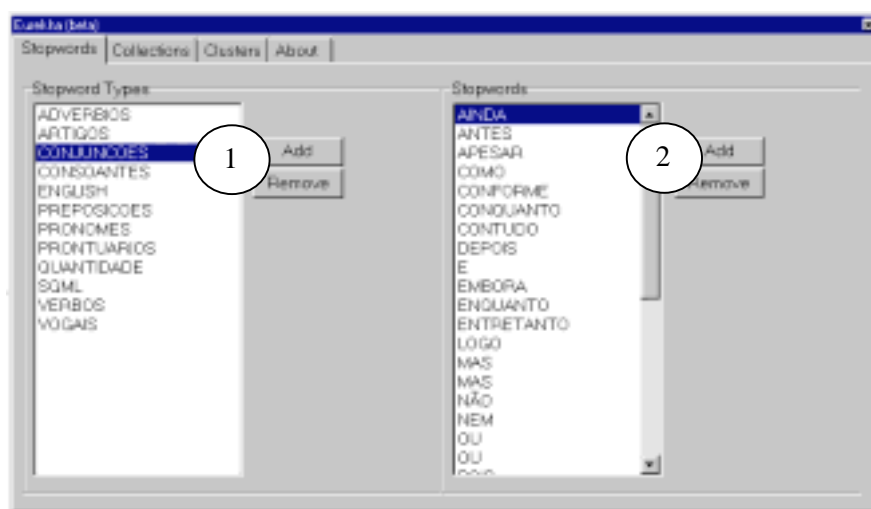


FIGURA 4.1 – Tabela de manipulação de palavras negativas

Caso o usuário deseje simplesmente verificar as palavras negativas pertencentes a determinado grupo, basta clicar no nome do tipo na região esquerda e as palavras correspondentes a este grupo apareçam na lista da direita. Por exemplo, na FIGURA 4.1 a categoria *Conjunções* está selecionada, e a lista de palavras da categoria (*ainda, antes, apesar...*) são listadas no lado direito da janela.

Para criar um novo tipo (categoria), basta clicar no botão *Add* (adicionar) esquerdo. Com isso, surge uma caixa de diálogo requisitando o nome da nova categoria (FIGURA 4.2). Após indicar o nome e pressionar a tecla [ENTER], a categoria será adicionada.

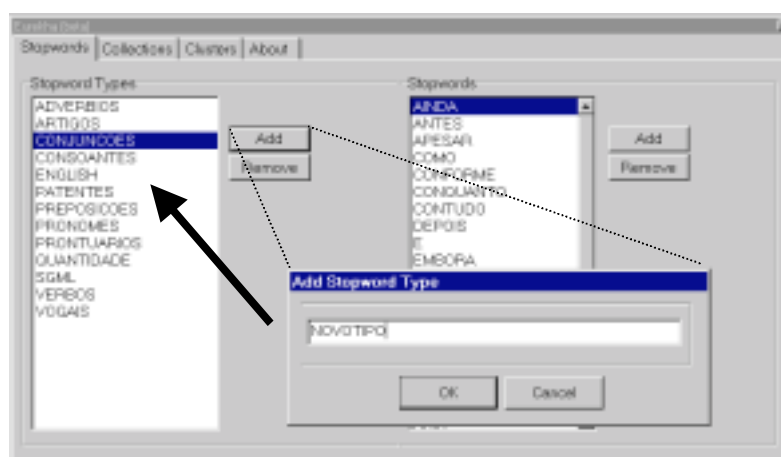


FIGURA 4.2 – Caixa de diálogo para adicionar categoria

Para incluir novas palavras em uma categoria, basta selecionar a categoria (lista esquerda) e pressionar no botão *Add* (adicionar) que está ao lado da lista direita. Novamente, uma caixa de diálogo surge e as palavras da respectiva categoria podem ser inseridas (FIGURA 4.3). Neste caso, ao pressionar-se a tecla [ENTER] (ou o botão *OK*), a caixa de diálogo permanece ativa para que várias palavras possam ser inseridas em seqüência. A caixa de diálogo fecha-se somente quando se clica no botão *Cancel*.

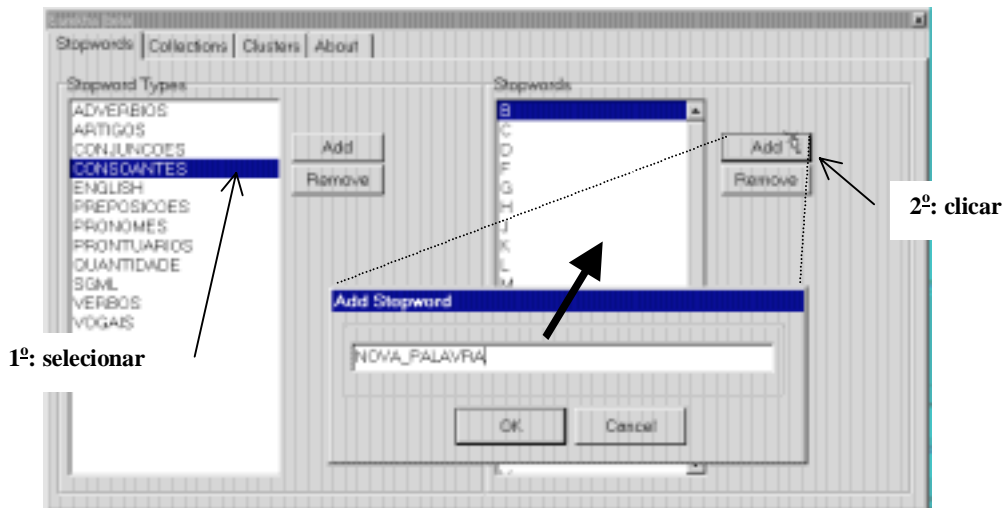


FIGURA 4.3 – Caixa de diálogo para adicionar palavras negativas

## 4.1.2 Tabela de coleções

Na ferramenta implementada é possível realizar vários testes com diversos conjuntos de documentos. Para cada conjunto de documentos a testar deve-se criar uma *coleção*. As coleções são criadas e manipuladas na *Orelha de Coleções (Collections)*, apresentada na FIGURA 4.4.

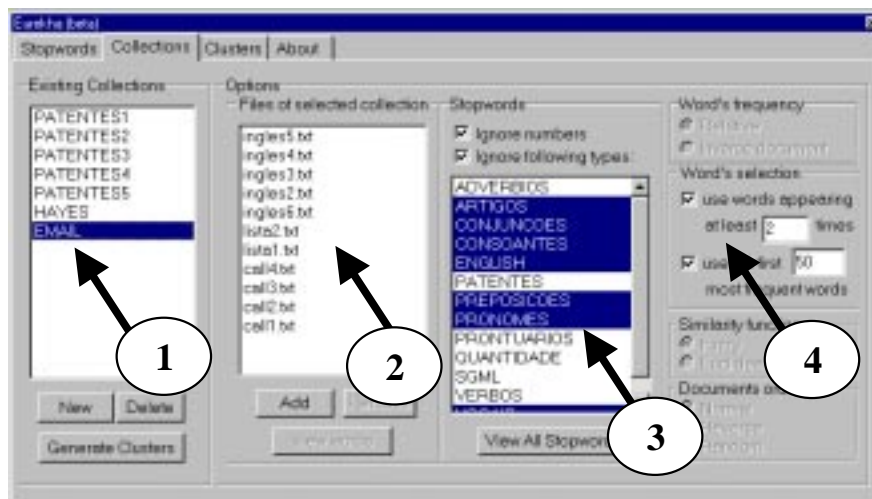


FIGURA 4.4 – Orelha de manipulação de coleções

Uma coleção é, portanto, um conjunto de documentos que irá ser tratado pela ferramenta. A coleção é a unidade que o programa utiliza para separar os documentos. As coleções é que são utilizadas no processo de agrupamento, e os documentos da coleção são separados, agrupados em sub-coleções ou sub-grupos. Uma coleção não

influencia em outra coleção. Assim, é possível trabalhar com diversas coleções simultaneamente.

A *Orelha de Coleções* possui quatro regiões, detalhadas a seguir:

### a) Manipulação de coleções

A primeira região (região 1 da FIGURA 4.4) permite com que o usuário crie uma nova coleção, selecione e/ou remova uma coleção já existente. Para criar uma nova coleção basta clicar no botão *new* situado abaixo desta região. Neste instante, surge uma caixa de diálogo requisitando o nome da nova coleção.

Para selecionar uma coleção basta selecioná-la com o *mouse*. Co- isso, todas as outras regiões serão alteradas, refletindo as configurações da coleção selecionada.

Após selecionada, uma coleção pode ser excluída pressionando-se o botão *delete*. É importante salientar que os arquivos-texto correspondentes não são removidos, mas sim, somente o arquivo de definições da coleção.

A definição e a adição dos arquivos que fazem parte de uma coleção é detalhada a seguir, mas para isso a coleção na qual deseja-se adicionar arquivos deve estar selecionada.

Há ainda um último botão denominado *Generate Clusters*. Este botão serve para iniciar o processo de geração da matriz de similaridades. Só deve ser pressionado após todas as outras regiões estarem devidamente configuradas. De qualquer forma, o botão permanece inativo até que os arquivos que fazem parte da coleção tenham sido definidos (as outras regiões, com exceção da definição dos arquivos, não necessitam ser alteradas, já que possuem valores *default*).

Quando este botão (*generate clusters*) é pressionado, surge a janela de progresso, apresentada na figura seguinte. Nesta janela, é possível verificar a ação em andamento, o nome do arquivo sendo processado, o tempo de processamento já decorrido e o tempo estimado de processamento.

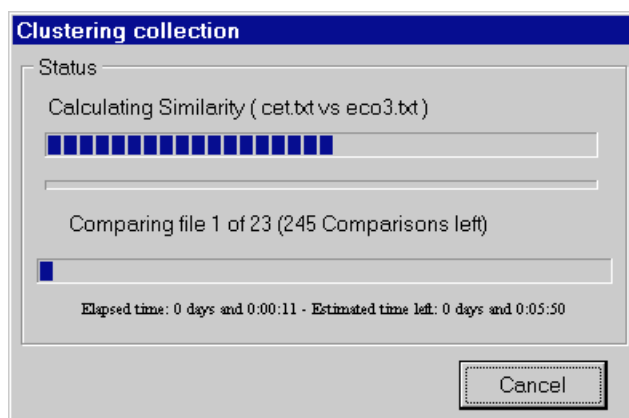


FIGURA 4.5 – Janela de progresso do processamento

## b) Definição de arquivos

A segunda região (região 2, FIGURA 4.4) serve para definir os arquivos que fazem parte de uma coleção. Ao selecionar uma coleção automaticamente os respectivos arquivos são listados. Para adicionar arquivos na coleção basta clicar no botão *Add* desta região. Com isso, surge uma caixa de diálogo similar à apresentada na FIGURA 4.6 seguinte.

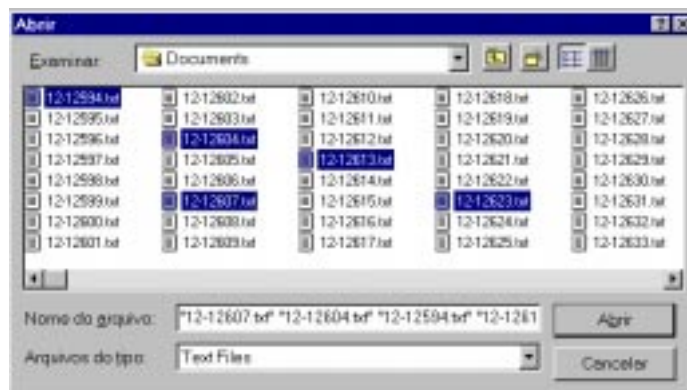


FIGURA 4.6 – Seleção de arquivos de uma coleção

Estando a caixa de diálogo ativa, basta selecionar o nome dos arquivos. Mantendo-se pressionada a tecla *Control*, é possível selecionar mais de um arquivo. Também é possível utilizar máscaras (*wildcards*), como por exemplo a máscara *\*.txt*, que seleciona todos os arquivos de extensão *.txt*.

Um aspecto importante diz respeito ao fato da ferramenta somente poder utilizar arquivos no formato ASCII (.txt). Portanto, arquivos de outros formatos (*doc*, *ps*, *rtf*, ...) devem ser convertidos previamente. Além disso, só podem ser utilizados documentos que estejam dentro do subdiretório *Documents*, que pode ser encontrado no diretório de instalação da ferramenta (geralmente: *C:\Arquivos de programas\Eureka\Documents*). Documentos que estejam fora deste diretório não podem ser utilizados, e devem ser copiados ou movidos para este diretório.

O botão *Remove* serve para remover um arquivo da coleção atual (selecionada). Ao ser pressionado, este botão retira este arquivo da lista de arquivos da coleção. Porém, o respectivo arquivo não é eliminado do diretório. Com isso, várias coleções podem utilizar os mesmos arquivos, de forma independente.

Nesta região é possível visualizar as palavras (características) de um documento qualquer da coleção. Basta selecionar um documento e clicar no botão *view words*. Este botão abre uma nova janela contendo todas as palavras encontradas no documento (FIGURA 4.7). Além das palavras, a janela apresenta ainda a frequência relativa e a frequência absoluta de cada palavra.

Na listagem de palavras de um documento há ainda uma informação adicional: um ícone posicionado a esquerda de cada palavra. Este ícone indica se ela será utilizada ou não no processo de agrupamento, de acordo com as configurações estabelecidas nas demais regiões.

Um ícone vermelho, contendo um sinal SW cortado, indica que a palavra foi identificada como *stopword*, e, portanto, não será utilizada no processo. Um ícone contendo o sinal de aprovação verde indica que a palavra não apresenta restrições,

podendo ser utilizada no processo. Já o ícone semi-transparente, indica que a palavra não atingiu a frequência mínima (definida pelo usuário na região de configurações de pré-processamento), e não será utilizada.

Word	Absolute Freq.	Relative Freq.
18	1	1
19	2	2
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1
36	1	1
37	1	1
38	1	1
39	1	1
40	1	1
41	1	1
42	1	1
43	1	1
44	1	1
45	1	1
46	1	1
47	1	1
48	1	1
49	1	1
50	1	1
51	1	1
52	1	1
53	1	1
54	1	1
55	1	1
56	1	1
57	1	1
58	1	1
59	1	1
60	1	1
61	1	1
62	1	1
63	1	1
64	1	1
65	1	1
66	1	1
67	1	1
68	1	1
69	1	1
70	1	1
71	1	1
72	1	1
73	1	1
74	1	1
75	1	1
76	1	1
77	1	1
78	1	1
79	1	1
80	1	1
81	1	1
82	1	1
83	1	1
84	1	1
85	1	1
86	1	1
87	1	1
88	1	1
89	1	1
90	1	1
91	1	1
92	1	1
93	1	1
94	1	1
95	1	1
96	1	1
97	1	1
98	1	1
99	1	1
100	1	1
101	1	1
102	1	1
103	1	1
104	1	1
105	1	1
106	1	1
107	1	1
108	1	1
109	1	1
110	1	1
111	1	1
112	1	1
113	1	1
114	1	1
115	1	1
116	1	1
117	1	1
118	1	1
119	1	1
120	1	1
121	1	1
122	1	1
123	1	1
124	1	1
125	1	1
126	1	1
127	1	1
128	1	1
129	1	1
130	1	1
131	1	1
132	1	1
133	1	1
134	1	1
135	1	1
136	1	1
137	1	1
138	1	1
139	1	1
140	1	1
141	1	1
142	1	1
143	1	1
144	1	1
145	1	1
146	1	1
147	1	1
148	1	1
149	1	1
150	1	1
151	1	1
152	1	1
153	1	1
154	1	1
155	1	1
156	1	1
157	1	1
158	1	1
159	1	1
160	1	1
161	1	1
162	1	1
163	1	1
164	1	1
165	1	1
166	1	1
167	1	1
168	1	1
169	1	1
170	1	1
171	1	1
172	1	1
173	1	1
174	1	1
175	1	1
176	1	1
177	1	1
178	1	1
179	1	1
180	1	1
181	1	1
182	1	1
183	1	1
184	1	1
185	1	1
186	1	1
187	1	1
188	1	1
189	1	1
190	1	1
191	1	1
192	1	1
193	1	1
194	1	1
195	1	1

FIGURA 4.7 – Listagem de palavras (características) de um documento

A janela apresenta ainda informações sobre o número total de palavras no documento, a frequência total das palavras, o número de *stopwords* encontradas, o número de palavras ignoradas por terem frequências mais baixas do que o limiar estabelecido pelo usuário (ver alínea D desta seção, a seguir) e o número de palavras candidatas (que podem ser utilizadas), tudo isso levando em conta as demais configurações.

### c) Configuração de palavras negativas

A região de palavras negativas (*stopwords*) é a terceira região em destaque na FIGURA 4.4. É nesta região que o usuário pode selecionar as classes de palavras negativas (definidas na orelha de palavras negativas). Para tanto, basta indicar que se deseja ignorar palavras negativas, ativando a opção *ignore following types*. Com isso, a lista de tipos de palavras negativas torna-se ativa e é possível então selecionar, com o mouse, aquelas que não devem ser consideradas no processo. É possível, também, indicar que números sejam ignorados. Neste caso, a opção *ignore numbers* deve ser ativada.



FIGURA 4.8 – Área de seleção de palavras negativas



A FIGURA 4.8, anterior, apresenta a região de palavras negativas com as opções de *ignore following types* (*ignorar os seguintes tipos*) e *ignore numbers* (*ignorar números*) ativas. Isso significa que os números serão considerados palavras negativas no processo de agrupamento, e, portanto, ignorados. Nesta figura também estão selecionadas as classes de *artigos*, *consoantes*, *english*, *patentes* e *sgml*, entre outras, que serão ignoradas. É importante salientar que, mesmo que estas estejam marcadas, a opção de *ignorar os tipos seguintes* (*ignore following types*) deve estar ativa. Caso esta opção seja desativada as respectivas classes passarão a ser utilizadas, isto é, não serão mais consideradas palavras negativas.

O botão *View all Stopwords* serve para listar todas as palavras negativas, de acordo com as classes selecionadas.

#### d) Configurações de pré-processamento

A quarta e última região (região 4, FIGURA 4.4) encontrada na *orelha* de coleções apresenta uma série de opções que podem ser ajustadas para que o agrupamento torne-se mais rígido ou mais relaxado (aumentando ou diminuindo o tempo de processamento), de acordo com os resultados que se deseja obter.

A maioria destas opções ainda não foi implementada, mas foram previstas para implementações futuras. No momento, somente é possível definir a frequência absoluta mínima (uma etapa de pré-processamento indicada por *Schütze* [SCH 97]) e o número máximo de características (palavras) que podem ser utilizadas em cada documento (técnica de truncagem).

Para definir a frequência mínima que as palavras devem ter para serem selecionadas como características relevantes de um documento deve-se ativar a opção *use words appearing at least  $X$  times*, onde  $X$  é o grau de frequência mínima definido pelo usuário.

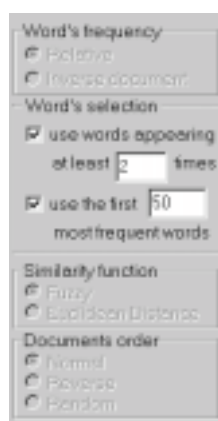


FIGURA 4.9 – Área de configuração de opções de pré-processamento

O limite máximo de características (truncagem) permitido para cada documento é definido na opção *use the first  $X$  most frequent words*, onde  $X$  é o número máximo de palavras. Com isso, somente as  $X$  características (palavras) mais importantes são utilizadas, fator que torna o processo de cálculo de similaridades mais rápido, porém, já que menos características são utilizadas, menos rígido. Ou seja, os grupos podem não ser tão precisos.

### 4.1.3 Tabela de agrupamento

Na tabela ou orelha anterior (seção 4.1.2) define-se coleções e geram-se matrizes de similaridades entre os elementos de uma coleção. O próximo passo na etapa de agrupamento é, justamente, agrupar estes elementos, cujas similaridades são especificadas na matriz, de acordo com alguma restrição.

A última tabela da ferramenta implementada contém justamente as etapas finais do processo: os algoritmos de agrupamento. Pode ser visto na FIGURA 4.10 que existem nove regiões disponíveis.

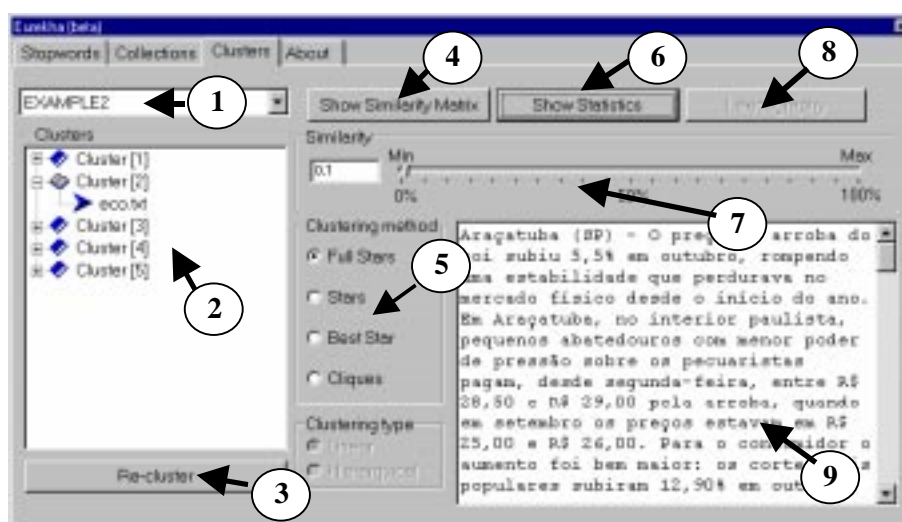


FIGURA 4.10 – Orelha de agrupamento

#### a) Selecionando uma coleção

Na FIGURA 4.10, a região identificada pelo número um (1) serve para que o usuário selecione a coleção que ele deseja trabalhar. Ao clicar com o mouse na pequena seta existente à direita desta região, abre-se uma lista com todas as coleções cuja matriz de similaridade já tenha sido calculada. Portanto, não é possível visualizar nesta etapa coleções que ainda não tenham sofrido o processo de cálculo de similaridades.

Quando selecionada, uma coleção é agrupada automaticamente, utilizando as configurações atuais das demais regiões. Após isso, caso alguma configuração seja alterada, é necessário pressionar o botão de re-agrupamento (*re-cluster*) para que os grupos sejam atualizados de acordo.

#### b) Visualizando grupos e documentos

Na região 2 da FIGURA 4.10 são mostrados os grupos de documentos constituídos a partir da coleção selecionada. Um ícone em forma de livro indica que

determinado grupo (cluster) está fechado. Para abri-lo, e descobrir os documentos que foram atribuídos a ele, basta clicar duplamente com o mouse em cima do nome do grupo. Com isso, o ícone passa a representar um livro aberto (indicando que o grupo está aberto) e todos os documentos atribuídos ao grupo são listados.

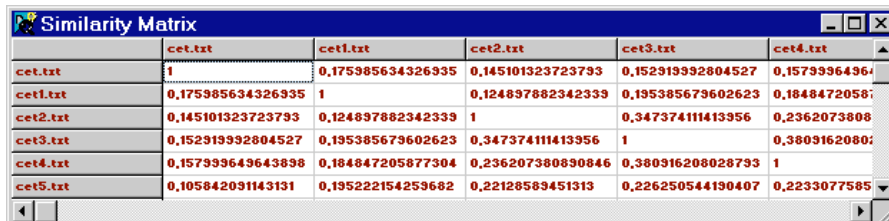
Quando um documento é selecionado, a região nove (9) apresenta seu conteúdo. Portanto, é possível visualizar, um-a-um, todos os documentos de determinado grupo. Com isso, consegue-se lê-los e descobrir o porquê de terem sido agrupados.

### c) Selecionando e configurando o método de agrupamento

Foram implementados quatro métodos de agrupamento: o método *cliques*, *stars*, *full-stars* e *best-star*. Atualmente, não é possível construir hierarquias entre grupos, ficando esta opção para trabalhos futuros. O algoritmo de agrupamento desejado pode ser escolhido na quinta região (ver FIGURA 4.10).

Cada algoritmo de agrupamento possui um método próprio, estes métodos já foram detalhados em capítulos anteriores. Todos estes métodos baseiam-se na matriz de similaridades. Portanto, a ferramenta oferece a opção de mostrar a matriz de similaridades, o que pode ser feito pressionando-se o botão denominado *show similarity matrix*, indicado como sendo a quarta (4<sup>a</sup>) região.

Nesta matriz, cujo exemplo é apresentado na FIGURA 4.11, o usuário pode verificar o quanto cada documento relaciona-se com os outros, o que facilita a tarefa de compreensão de o porquê de determinado documento ter sido atribuído a determinado grupo.



	cet.txt	cet1.txt	cet2.txt	cet3.txt	cet4.txt
cet.txt	1	0,175985634326935	0,145101323723793	0,15291992804527	0,157999649643898
cet1.txt	0,175985634326935	1	0,124897882342339	0,195385679602623	0,184847205877304
cet2.txt	0,145101323723793	0,124897882342339	1	0,347374111413956	0,2362073808
cet3.txt	0,15291992804527	0,195385679602623	0,347374111413956	1	0,3809162080
cet4.txt	0,157999649643898	0,184847205877304	0,236207380890846	0,380916208028793	1
cet5.txt	0,105842091143131	0,195222154259682	0,22128583451313	0,226250544190407	0,2233077585

FIGURA 4.11 – Matriz de similaridades

Além da matriz, outro fator importante na identificação dos grupos é o grau mínimo de similaridade. A sétima região permite que o usuário estabeleça este grau. Estabelecer este grau é tarefa complicada. Este valor depende muito de cada coleção, e não há um valor padrão.

Geralmente, utiliza-se a matriz de similaridades para se ter uma idéia da média de valores (similaridades) entre os documentos. O grau mínimo deve então ser estabelecido de acordo com esta média, que no momento não é calculada pela ferramenta. É claro que estabelecer este valor depende muito do usuário e do resultado que ele espera. Pode-se afirmar que quanto maior este grau, maior será a exigência por parte do algoritmo, o que leva à constituição de grupos mais coesos. Porém, neste caso, os documentos devem ser muito similares, com muitas palavras em comum, para que sejam atribuídos ao mesmo grupo.

Quanto menor o grau de similaridade (mais próximo de 0%), menos palavras em comum os documentos precisam ter, o que pode acarretar em grupos menos coesos.

É importante salientar que qualquer alteração realizada na configuração (tipo de agrupamento, valor de similaridade mínima) não afeta o agrupamento atual. Para que as alterações façam efeito, é necessário pressionar o botão de re-agrupamento.

#### d) Visualizando informações estatísticas

A ferramenta apresenta duas janelas auxiliares com informações adicionais sobre os grupos constituídos. Estas informações servem para facilitar o processo de descoberta de conhecimento.

A primeira facilidade oferecida pela ferramenta é a janela contendo um gráfico de barras que relaciona a quantidade de documentos por grupo encontrado. Esta janela pode ser visualizada pressionando-se o botão indicado como área seis (6) na FIGURA 4.10. Esta janela, que apresenta a percentagem de documentos por grupo, possui a aparência da FIGURA 4.12.

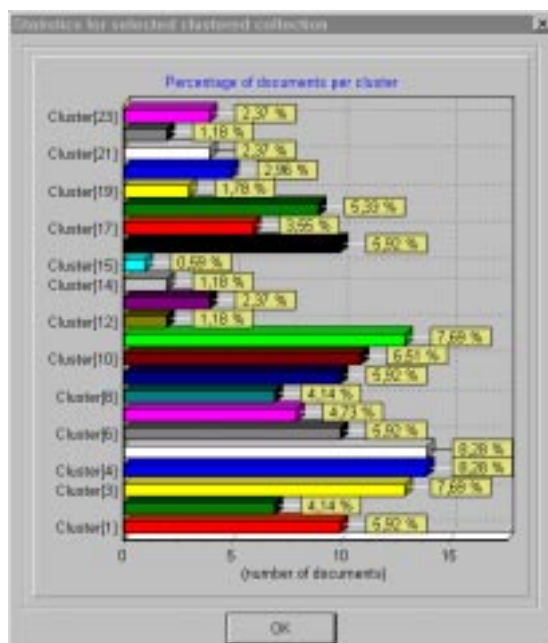


FIGURA 4.12 – Janela de informações estatísticas

Outra informação útil é a listagem de palavras importantes de um grupo. O botão *lexicografia (lexicography)* – área 8, FIGURA 4.10 – abre a janela apresentada na figura seguinte, contendo informações sobre as palavras que mais aparecem em determinado grupo. Além de listar as palavras, a ferramenta apresenta informações sobre a quantidade de documentos em que a palavra aparece e a quantidade de vezes (soma das freqüências absolutas) que ela aparece em todos os documentos.

Estas palavras podem ser consideradas o *centróide* do grupo, ou seja, as características principais do grupo. Com isso, tem-se uma idéia geral do assunto abordado por todos os documentos do grupo, o que facilita a recuperação de informações (já que o usuário pode escolher o grupo cujo assunto seja mais relevante para ele) e, também, a descoberta de conhecimento, já que estas palavras principais constituem o conhecimento abordado pelo grupo.

Word	Documents	Frequency
EM	5	7
HOJE	4	4
NOITE	4	4
TEM	3	3
RIO	3	4
COMO	3	4
SUA	3	3
INTER	3	6
DOS	3	4
JOGADORES	3	3
BRASILEIRO	2	2
ALÉM	2	2
ESPECIAL	2	2
JANEIRO	2	2
TAMBÉM	2	2
ES	2	3
SE	2	2
GRANDE	2	2
TRATAMENTO	2	2
EMPRESAS	2	3

FIGURA 4.13 – Listagem de palavras principais de um documento

## 4.2 Restrições

A implementação não é considerada a principal contribuição do trabalho, apesar disso, a ferramenta mostra-se bastante útil. Porém, por ter sido elaborada em um pequeno espaço de tempo e, a princípio, sem visar lucros comerciais, possui alguns problemas e restrições. Planeja-se, como trabalho futuro, corrigir estes erros e ampliar suas funcionalidades.

Dentre as funcionalidades inexistentes, incluem-se as fases consideradas de limpeza de dados: correção ortográfica, normalização de termos e substituição de pronomes por seus respectivos substantivos. Neste caso, crê-se que os documentos já tenham sido tratados (limpos) quando adicionados ao diretório de documentos da ferramenta.

Estas fases de limpeza não são realmente obrigatórias, mas ajudam no processo, pois documentos normalizados (com termos de um mesmo domínio) e corretos tendem a ser mais coesos. A fase de limpeza, quando não realizada, pode tornar o resultado errôneo, já que a ferramenta não tem como saber se determinada palavra (por estar diferente ortograficamente) corresponde à mesma palavra que aparece em outro documento.

Outro problema diz respeito ao fato de todas as partes do texto serem igualmente utilizadas e, portanto, igualmente importantes. Não é possível delimitar uma sub-região, como por exemplo o título e as linhas iniciais do texto, limitando assim a área de atuação, o que poderia tornar o algoritmo mais rápido. A delimitação de textos também poderia servir para que o usuário excluísse regiões consideradas irrelevantes ou que contivessem informações que pudessem *influenciar* no processo, de acordo com sua intuição.

A princípio, esta funcionalidade (de poder delimitar regiões) não foi implementada porque o objetivo inicial do trabalho é trabalhar com textos genéricos, onde, nestes casos, todo o texto pode vir a contribuir com seu entendimento.

Outra funcionalidade não implementada corresponde ao fato de as datas serem *quebradas* em números, ou seja, ao encontrar a seqüência 10/10/1997, as barras (“/”)

são retiradas pelo pré-processamento, fazendo com que as seqüências 10, 10 e 1997 sejam reconhecidas como números isolados. Por isso, não é possível realizar um agrupamento por datas específicas. Porém, textos que contenham a mesma data possuirão os mesmos números, o que pode levá-los a ser atribuídos ao mesmo grupo.

Similarmente, a ferramenta implementada não faz distinção entre palavras escritas com letras maiúsculas e palavras escritas com letras minúsculas, todas são convertidas para um único formato: letras maiúsculas. Em alguns casos o usuário pode achar interessante realizar esta distinção, principalmente quando tratando com nomes de pessoas e de países (diferenciando entre o nome comum e o nome próprio). Futuramente, seria interessante adicionar uma opção onde o usuário pudesse escolher se quer ou não ignorar a diferenciação entre letras maiúsculas e minúsculas – opção sensível ao tipo (*case sensitive*).


Outras funcionalidades, presentes na interface mas não implementadas, incluem a utilização de outras fórmulas de cálculo de importância (frequência) de características. Assim, seria possível realizar agrupamentos utilizando mais de uma fórmula de frequência, comparando os resultados. Outras fórmulas de similaridade, como por exemplo a função *cosine* não foram implementadas, e são igualmente interessantes.

O tipo de agrupamento hierárquico ainda não é permitido no estágio atual da ferramenta. Porém, com a utilização de *centróides* e com a aplicação recursiva dos algoritmos existentes, seria possível adaptá-la a este tipo de agrupamento.

Além disso, não é possível modificar a ordem de análise dos documentos na matriz de similaridades. Deste modo, não é possível verificar se determinados algoritmos produziram resultados diferentes utilizando inicialmente, por exemplo, os documentos finais da matriz. Em algoritmos como o *Stars* isso pode acontecer se um documento pertencer a mais de um grupo. Já o algoritmo *Best-Star* elimina este problema, já que adiciona o elemento ao grupo cuja similaridade for maior, criando grupos mais coesos, independente da ordem utilizada.

## 5 Estudos de caso

Para poder comparar uma técnica de agrupamento com outra, a fim de determinar a técnica superior, é necessário identificar qual delas oferece melhor resultado e desempenho. Infelizmente, devido à diversidade de métodos de agrupamento existentes e ao número excessivo de aplicações para as quais estes métodos são utilizados, não há uma medida de eficiência e eficácia padrão para que estes métodos possam ser comparados.

Devido a isso, os primeiros estudos na área de agrupamento ou classificação de objetos textuais resolveram adotar duas medidas muito utilizadas pelas técnicas de recuperação de informações – a *abrangência* (*recall*) e a  *precisão* (*precision*).

A *Abrangência* serve para indicar se a quantidade de documentos recuperados em uma consulta é igual à quantidade de documentos que tratam do assunto solicitado pelo usuário, existentes na coleção, ou seja, a proporção de documentos retornados em relação à quantidade de documentos que deveriam ser retornados. Para tanto, deve-se saber previamente a quantidade de documentos que podem ser recuperados por uma consulta a determinado assunto.

No caso da classificação de documentos, esta métrica consiste em dividir a quantidade de categorias atribuídas pela quantidade de categorias que deveriam ser atribuídas. Ou ainda, a quantidade de documentos atribuídos a uma categoria dividida pela quantidade de documentos que deveriam ser atribuídos a esta categoria.

A *Precisão* é uma medida que indica se todos os documentos recuperados tratam realmente do assunto solicitado, ou seja, a proporção de documentos retornados que são realmente relevantes. Neste caso, em relação à classificação, a precisão é calculada dividindo-se a quantidade de categorias *corretamente* atribuídas pela quantidade de categorias atribuídas. Ou ainda, a quantidade de documentos atribuídos corretamente a uma categoria dividida pela quantidade de documentos atribuídos a esta mesma categoria.

É claro que para cada grupo de documentos formado há então um número  $x$  de documentos que tratam do mesmo assunto (são da mesma categoria) e devem fazer parte do mesmo grupo. Como podem existir muitos grupos e muitos documentos por grupo, é necessário, de alguma forma, utilizar uma medida mais genérica que leve em conta todos os grupos simultaneamente.

Com base nestas medidas (*recall* e *precision*), e levando em conta que o método de agrupamento deve realizar diversas comparações para decidir a que grupo pertence determinado documento, podem ser utilizadas outras duas medidas, conhecidas por *Microaveraging* e *Macroaveraging*.

*Microaveraging*, segundo David Lewis [LEW 91], é uma técnica que trata todas as possibilidades (decisões) como se pertencessem a um único grupo, calculando a *precisão* ou a *abrangência* global. Já a *Macroaveraging* é uma técnica que calcula a *precisão* (ou *abrangência*) de cada grupo separadamente. Após, estas medidas intermediárias são somadas e sua média é calculada. Esta média é tomada como

resultado da eficiência do método. Em grande parte dos estudos realizados na área de classificação de informações textuais a *Microaveraging* é utilizada<sup>4</sup>.

Por outro lado, estas métricas só podem ser utilizadas quando se tem esse conhecimento prévio sobre os assuntos ou classes de cada documento. Como uma das vantagens do agrupamento é justamente separar documentos sem que se tenha esse conhecimento prévio dos documentos, elas tornam-se um pouco inadequadas.

Porém, como o objetivo é testar a eficiência do método, pode-se (e deve-se) utilizar uma coleção de documentos conhecida e padrão para que estas métricas possam ser aplicadas. Após comprovada a eficiência do método as métricas podem ser dispensadas e este pode ser aplicado em coleções de documentos desconhecidos.

Deste modo, neste trabalho buscou-se identificar uma coleção de referência que fosse utilizada por outros estudos e que pudesse, então, ser utilizada para avaliar o método proposto.

Encontra-se disponível na Internet uma coleção de documentos elaborada especificamente para fins de classificação e agrupamento de informações. Esta coleção é distribuída pela agência de notícias *Routers* e pode ser utilizada livremente, desde que sejam feitas menções à sua fonte. Esta coleção foi recuperada e utilizada em alguns dos testes realizados com a ferramenta descrita no capítulo anterior. Os resultados destes testes são analisados na seção 5.3.

Além do teste de eficiência em relação ao resultado obtido pelo método (coesão dos grupos), procurou-se testar que tipo de influência as mais variadas opções e parâmetros do método causam no resultado do agrupamento e no tempo de agrupamento. Estes outros testes são apresentados na seção seguinte, e foram realizados utilizando uma outra coleção, não padrão, que se aproxima da aplicação real para qual o método foi desenvolvido.

A TABELA 5.1 apresenta as listas de palavras definidas como negativas, ignoradas pelos processos de agrupamento realizados. Estas listas foram construídas fazendo-se análises manuais em textos, páginas WEB e em consultas na Internet. Geralmente são compostas de palavras ou letras que aparecem com muita frequência nos textos, como por exemplo, preposições, conjunções, pronomes e alguns verbos. Algumas são específicas de domínios utilizados em um ou outro estudo de caso.

---

<sup>4</sup> Para efeitos de classificação a métrica de *Microaveraging* é adotada por ser aplicada mais facilmente do que a *Macroaveraging*. Isso porque na classificação os documentos são apresentados ao método de classificação e este retorna (ou não) a classe a que este documento (supostamente) pertence. Com isso, a *Microaverage Recall* consiste em calcular o resultado da divisão da quantidade de documentos classificados pelo número de documentos não classificados (isso é feito rapidamente). A *Microaverage Precision* é calculada dividindo-se o número de documentos classificados pelo número de documentos corretamente classificados.



TABELA 5.1 – Listas de palavras negativas utilizadas

TIPO	PALAVRAS
CONSOANTES	B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Y, Z
PREPOSIÇÕES	A, À, ANTE, AO, APOS, ATE, ATÉ, COM, CONTRA, DA, DE, DESTA, DO, NA, NO, PARA, PERANTE, POR, SEM, SOB, SOBRE, TRAS, DURANTE, COMO, CONFORME, EXCETO, MEDIANTE, AFORA, ENTRE, COMO, PER
VOGAIS	A, E, I, O, U
ARTIGOS	A, AS, O, OS, UM, UMA, UMAS, UNS
PRONOMES	COMIGO, CONTIGO, DELE, DELES, ELE, ELES, EU, ME, MEU, MI, NOS, NÓS, NOSSAS, NOSSOS, SEU, TEU, TU, VOS, VÓS, ELA, ELAS, ME, TE, O, A, SE, LHE, MIM, TI, SI, NOS, OS, AS, LHES, COMIGO, CONTIGO, CONSIGO, CONOSCO, CONVOSCO, VOCE, VOCÊ, VOCES, VOCÊS, SENHOR, SENHORA, VOSSA, MEU, SEU, NOSSO, VOSSO, SEU, MINHA, TUA, SUA, NOSSA, VOSSA, SUA, MEUS, TEUS, SEUS, NOSSOS, VOSSOS, SEUS, MINHAS, TUAS, SUAS, NOSSAS, VOSSAS, SUAS, ESTE, ESTA, ISTO, ESSE, ESSA, ISSO, AQUELE, AQUELA, AQUILO, MESMO, PRÓPRIO, PRÓPRIO, SEMELHANTE, TAL, ALGUÉM, ALGUEM, NINGUÉM, NINGUEM, TUDO, NADA, ALGO, OUTREM, NENHUM, OUTRO, UM, CERTO, QUALQUER, QUAISQUER, ALGUM, CADA, QUEM, QUAL, ESTE, QUANTOS, QUE, CUJO, QUAIS, CUJA, CUJOS, CUJAS, QUANTO, QUANTA, QUANTOS, QUANTAS, ONDE
CONJUNÇÕES	E, MAS, OU, QUE, QUANDO, PORQUE, OU, NEM, NÃO, MAS, PORÉM, POREM, CONTUDO, TODAVIA, ENTRETANTO, SENÃO, SENAO, LOGO, POIS, PORTANTO, COMO, QUANTO, EMBORA, CONQUANTO, APESAR, AINDA, CONFORME, SEGUNDO, TAL, TÃO, TAO, TANTO, QUANDO, DEPOIS, ANTES, ENQUANTO
PATENTES	ABI, ABSTRACT, ACCESS, AUTHORS, CODES, COMPANIES, DATE, GEO, GLOBAL, INFORM, ISSN, JOURNAL, NO, PLACES, PROQUEST, REPRINT, SUBJECTS, TITLE, VOL
INGLÊS/ENGLISH	ABOUT, ABOVE, ACCORDING, ACCOUNT, ACTUAL, ACTUALLY, ADDED, ADDING, ADDITIONAL, ADDITIONS, ADDRESSED, ADDRESSES, AFTER, AFTERWARDS, AGAIN, AGAINST, AHEAD, ALL, ALMOST, ALONE, ALONG, ALREADY, ALSO, ALTHOUGH, ALWAYS, AMONG, AMOUNT, AN, AND, ANOTHER, ANSWER, ANYONE, ANYTHING, APPARENT, APPEARED, APPEARS, APPLIED, APPLY, ARE, AREA, AREAS, AREN, ASIDE, ASKED, ASKING, ASPECTS, ASSUME, ASSUMED, ASSUMES, AT, AVOID, AWARE, AWAY, BACK, BE, BECAME, BECAUSE, BECOME, BECOMES, BEEN, BEFORE, BEGAN, BEGIN, BEGINS, BEGUN, BEHOLD, BEING, BELOW, BESIDE, BESIDES, BETWEEN, BEYOND, BOTH, BROUGHT, BY, BYTE, BYTES, CALL, CALLED, CALLING, CALLS, CAME, CAN, CANNOT, CAUSE, CAUSED, CAUSES, CAUSING, CEASE, CEASED, CEASES, CERTAIN, CERTAINLY, CHANGE, CHANGED, CHANGES, CHANGING, CHOOSE, CIRCA, CLEAR, CLEARLY, COME, COMES, COMING, COMMON, COMMONLY, COMPARE, COMPLETE, CONCEPT, CONCEPTS, CONSIDER, CONSIDERED, CONSIDERS, CONSISTING, CONSISTS, CONSTANT, CONTAINED, CONTAINS, CONTENTS, CONTINUE, COULD, COULDN, CREATE, CREATED, CREATING, CURRENT, DATES, DAYS, DEALING, DEFAULT, DEFINE, DEFINED, DEGREE, DEPEND, DEPENDING, DEPENDS, DESCRIBE, DESCRIBED, DESCRIBES, DIDN, DIFFERENCE, DIFFERENCES, DIFFERENT, DIFFERENTLY, DIRECTLY, DISCUSS, DO, DOCUMENT, DOCUMENTS, DOES, DOESN, DOING, DONE, DOUBT, DURING, EACH, EARLIER, EARLY, EASIER, EASILY, EASY, EIGHT, EIGHTY, EITHER, ELEVEN, ELEVENTH, ELSE, EMAIL, ENDED, ENDING, ENDS, ENOUGH, ENTIRE, ENTIRELY, EQUAL, ESPECIALLY, EVEN, EVENTUALLY, EVER, EVERY, EVERYONE, EVERYTHING, EXACT, EXACTLY, EXAMPLE, EXCEPT, EXCEPTED, EXCEPTION, EXIST, EXISTED, EXISTENCE, EXISTING, EXPANSION, EXPECT, EXPECTS, EXTEND, EXTENDING, EXTENDS, EXTENSION, EXTENT, EXTRA, FACT, FACTS, FEATURE, FEATURES, FIFTH, FINAL, FINALLY, FIND, FINDING, FINDS, FINISH, FIRST, FIVE, FOLLOW, FOLLOWED, FOLLOWING, FOLLOWS, FORM, FORMER, FORMS, FORTY, FOUND, FOUR, FOURTH, FREQUENT, FREQUENTLY, FROM, FRONT, FULL, FULLY, FURTHER, GAVE, GENERAL, GETS, GETTING, GIVE, GIVEN, GIVES, GIVING, GOES, GOING, GONE, GOOD, GREAT, GREATER, GREATLY, HAS, HAST, HATH, HAVE, HAVEN, HAVING, HE, HEAR, HEARD, HEARS, HELD, HELP, HELPS, HENCE, HERE, HEREIN, HERSELF, HIGH, HIGHER, HIGHLY, HIMSELF, HOLDS, HOME, HOPE, HOUR, HOWEVER, HREF, HTML, HTTP, I, IDEA, IDEAS, IF, IGNORED, IMAGE, IMAGINED, IMPLIED, IMPLY, IMPROVE, IN, INCLUDE, INCLUDES, INDICATE, INFER, INFO, INITIAL, INPUT, INSERT, INSIDE, INSTALL, INSTEAD, INTO, IS, IT, ITEM, ITEMS, ITS, ITSELF, JUST, KEEP, KEEPING, KEEPS, KEPT, KIND, KINDS, KNEW, KNOW, KNOWING, KNOWN, KNOWS, LACK, LARGE, LARGELY, LARGER, LAST, LASTED, LASTLY, LATE, LATELY, LATER, LATTER, LEAD, LEAST, LEAVE, LEAVING, LEFT, LESS, LESSER, LEVEL, LEVELS, LIES, LIFE, LIGHT, LIKE, LIKED, LIKELY,

TABELA 5.1 – Listas de palavras negativas utilizadas (continuação)

TIPO	PALAVRAS
INGLÊS/ENGLISH	LIKENED, LIKES, LIKING, LINES, LINK, LINKED, LINKS, LIST, LISTS, LITERAL, LITERALLY, LITTLE, LONG, LONGER, LOOK, LOOKED, LOOKING, LOOKS, LOTS, LOWER, MADE, MAIL, MAILTO, MAIN, MAINLY, MAKE, MAKES, MAKING, MANNER, MANY, MARK, MARKER, MARKS, MAYBE, ME, MEAN, MEANING, MEANS, MEANT, MEET, MEETS, MENU, MENUS, MERE, MERELY, MIDDLE, MIDST, MIGHT, MINE, MORE, MOST, MOSTLY, MOVED, MUCH, MULTI, MUST, MYSELF, NAME, NAMED, NAMELY, NAMES, NATURALLY, NEARLY, NEAT, NEATLY, NECESSARILY, NECESSARY, NEED, NEEDED, NEEDING, NEEDS, NEITHER, NEVER, NEWS, NEXT, NICE, NICELY, NINE, NINTH, NOBODY, NONE, NOONE, NORMAL, NOTE, NOTED, NOTES, NOTHING, NOTICE, NOTING, NOWHERE, OCCUR, OCCURS, OF, OFFER, OFTEN, ON, ONCE, ONES, ONLY, ONTO, OR, OTHER, OTHERS, OTHERWISE, OUGHT, OURS, OUTSIDE, OVER, OVERLY, OWED, OWES, OWING, PAGE, PAGES, PAIR, PAIRED, PART, PARTLY, PARTS, PASS, PASSED, PAST, PERHAPS, PIECES, PLACE, PLACED, PLANS, PLUS, POINT, POINTS, POSSIBLE, POSSIBLY, PRECEDES, PRECEDING, PREFACE, PREFER, PRESENT, PRESENTED, PRESENTS, PRESERVE, PREVIOUS, PREVIOUSLY, PRIMARILY, PROBABLE, PROBABLY, QUITE, QUOT, QUOTE, QUOTED, QUOTES, RARELY, RATHER, READ, READILY, READING, READS, READY, REAL, REALLY, REASON, REASONS, RECEIVE, REFERENCES, REFERENCING, REFERRED, REFERRING, REFERS, REGARD, REGARDLESS, RELATED, RELATIVE, RELEVANT, REMAIN, REMAINED, REMAINING, REMAINS, REPEATED, REPLACE, REQUIRE, REQUIRED, REQUIRES, REQUIRING, REST, RESULT, RESULTANT, RESULTED, RESULTING, RESULTS, REVISED, RIGHT, RISE, RULE, RULES, SAID, SAME, SAVE, SAVED, SAYING, SAYS, SEARCH, SECOND, SEEING, SEEK, SEEM, SEEMED, SEEMS, SEEN, SEES, SELF, SELVES, SEND, SENDING, SENDS, SENSE, SENT, SEPARATE, SEPARATELY, SET, SETTING, SEVEN, SEVENTH, SEVERAL, SHALL, SHALT, SHE, SHOULD, SHOULDN, SHOW, SHOWED, SHOWING, SHOWN, SHOWS, SIMPLE, SIMPLY, SINCE, SINGLE, SIXTH, SLIGHT, SLIGHTLY, SMALL, SO, SOLELY, SOME, SOMEBODY, SOMEONE, SOMETHING, SOMETIME, SOMETIMES, SOMEWHAT, SOMEWHERE, SOON, SOONER, SORT, SOUNDS, SPEAK, SPECIFIC, SPECIFICALLY, SPECIFY, SPECIFYING, STANDS, START, STARTED, STARTS, STATE, STATED, STATES, STATING, STATUS, STILL, STOPS, SUCH, SUPPOSE, SUPPOSED, SURE, SURELY, TABS, TAKE, TAKEN, TAKES, TAKING, TALK, TALKED, TALKS, TELL, TELLING, TELLS, TEND, TENDED, TENDING, TENDS, TENTH, TERM, TERMS, TEXT, THAN, THAT, THE, THEE, THEIR, THEIRS, THEM, THEME, THEMSELVES, THEN, THENCE, THERE, THEREBY, THEREFORE, THEREIN, THEREOF, THESE, THEY, THINE, THING, THINGS, THINK, THINKS, THIRD, THIS, THOSE, THOU, THOUGH, THREE, THROUGH, THROUGHOUT, THUS, TILL, TIME, TIMES, TO, TOGETHER, TOLD, TOOK, TOTAL, TOWARDS, TREATS, TRUE, TRUTH, TRYING, TURN, TURNED, TURNS, TWELFTH, TWELVE, TWENTY, TYPE, TYPES, TYPICAL, UNDER, UNDERSTAND, UNDERSTANDS, UNDERSTOOD, UNLESS, UNLIMITED, UNTIL, UNTO, UPON, URLs, USED, USEFUL, USER, USERS, USES, USING, USUAL, USUALLY, VARIETY, VARIOUS, VERY, VIEW, WANT, WANTED, WANTS, WASN, WATCH, WAYS, WELL, WENT, WERE, WHAT, WHATEVER, WHEN, WHERE, WHEREAS, WHEREBY, WHEREIN, WHETHER, WHICH, WHILE, WHOLE, WHOM, WHOSE, WILL, WILT, WITH, WITHIN, WITHOUT, WORSE, WORTH, WOULD, WOULDND, WRITE, WRITER, WRITES, WRITING, WRITTEN, WROTE, YOU, YOUR, YOURS, YOURSELF, ZERO
SGML	BODY, DATELINE, REUTER, TEXT, TITLE

## 5.1 Coleção de mensagens eletrônicas

Para dar início aos testes do método de agrupamento proposto, optou-se por uma coleção de documentos pequena e simples. Procurou-se identificar uma coleção de testes que se aproximasse mais dos experimentos cotidianos, para os quais a ferramenta de agrupamento pode ser utilizada.

O *correio eletrônico* é um serviço muito utilizado pela maioria das pessoas que acessam a *Internet*. Por esse motivo, como primeiro caso, foram selecionadas onze (11) mensagens de correio eletrônico, sendo que, destas, cinco são escritas em inglês e seis em português. As mensagens em português subdividem-se em duas classes: uma

contendo quatro (4) mensagens no estilo *call for papers* e duas (2) relativas a uma lista de discussão de episódios de um seriado (portanto com um total de 3 grupos).

Para esta coleção foram adotados os seguintes parâmetros: *ignorar números, ignorar as palavras negativas* pertencentes às classes *Artigos, Conjunções, Consoantes, English, Preposições, Pronomes e Vogais*. Além disso, optou-se por não limitar o número de palavras utilizadas, ou seja, não houve pré-processamento em nível de truncagem, a fim de obter um melhor resultado. O tempo de cálculo da matriz de similaridades foi de 23 minutos e 30 segundos em um microcomputador Pentium™ 133 com 96MB de memória. O número total de palavras processadas (encontradas em todos os documentos) corresponde a mil seiscentas e oito (1.608) palavras, com uma frequência total de onze mil quinhentas e dezessete (11.517) aparições (esse número não leva em conta as palavras negativas, pois foram ignoradas durante o processo).

O objetivo deste estudo de caso é identificar o melhor algoritmo de agrupamento (dentre os algoritmos implementados). Para tanto, o método consiste em calcular o grau de abrangência e precisão de cada grupo em cada um dos algoritmos, calculando sua média final. Esse tipo de análise corresponde à identificação da *macroaverage recall* e da *macroaverage precision*. Os três grupos de documentos identificados *a priori* foram utilizados como referenciais para cálculo destas duas métricas, ou seja, o algoritmo que identificar todos os grupos com seus respectivos documentos obtém valores de 100% em ambas as métricas.

Na seção anterior, apresentou-se a nível conceitual as métricas de *macroaverage recall* e *precision*. Estas, segundo *David Lewis*, calculam a abrangência ou a precisão de cada grupo e, após, apresentam a média entre estes valores obtidos em cada grupo. Neste estudo de caso a fórmula utilizada para obter esta métrica de *macroaverage recall*, de acordo com o conceito dado por *David Lewis*<sup>5</sup>, é apresentada na FIGURA 5.1.

$$\text{MACRO recall} = \frac{\sum_{i=1}^n R_i}{n}$$

FIGURA 5.1 – Fórmula de “macroaverage recall”

Nessa fórmula  $n$  representa o número de grupos e  $R$  a *abrangência* de cada um desses grupos.  $R$  é calculado utilizando-se a fórmula tradicional de *recall* que consiste em identificar o número de documentos atribuídos ao grupo e dividir este número pelo número de documentos que deveriam ser atribuídos ao grupo.

A fórmula de *macroaverage precision* é similar à formula anterior. Porém, neste caso, são utilizados os valores de *precisão* de cada grupo intermediário, e não os valores de *abrangência*. A *precisão* ( $P$ ) é calculada dividindo-se o número de documentos corretamente atribuídos ao grupo pelo número total de documentos atribuídos ao grupo.

$$\text{MACRO precision} = \frac{\sum_{i=1}^n P_i}{n}$$

FIGURA 5.2 – Fórmula de “macroaverage precision”

<sup>5</sup> *David Lewis* não apresenta as fórmulas de *macroaveraging* e *microaveraging*. Ele apresenta somente seu conceito.

Além das métricas, é necessário estabelecer um grau de similaridade mínimo (GSM), que será utilizado por todos os algoritmos. Preferiu-se adotar mais de um grau de similaridade mínima, identificando, assim, vários pontos (em locais diferentes) para análise. Assim, não há o perigo de ser utilizado um grau arbitrário, onde determinado algoritmo tenha melhor desempenho do que os outros e, ao mesmo tempo, ficando impossível saber se não há outro ponto, onde este mesmo algoritmo obteria resultados inferiores.

Os pontos de similaridade mínima escolhidos foram 0, 0.05, 0.1, 0.15, 0.2, 0.5 e 1, ou seja, valores bem distribuídos. O melhor algoritmo é aquele que obtém o melhor valor de *macroaverage recall* e *precision* em todos os pontos (a média dos valores).

O primeiro algoritmo a ser testado foi o algoritmo *Best-star*. Os resultados dos testes com este algoritmo podem ser visualizados no gráfico apresentado na FIGURA 5.3. Observa-se neste gráfico que, inicialmente, o algoritmo *Best-star* obtém um bom grau de abrangência (0.6 pontos). Isso, porque o algoritmo consegue identificar relações de igualdade entre os elementos, agrupando-os em grupos de objetos similares, mesmo quando o usuário utiliza um GSM de zero por cento (0%). Isso não ocorre com nenhum dos outros algoritmos, conforme pode ser visto na FIGURA 5.5. Por isso, diz-se que este algoritmo consegue identificar os relacionamentos naturais entre os objetos, sem que o usuário tenha que se preocupar em encontrar o grau de similaridade mínima mais adequado para a coleção.

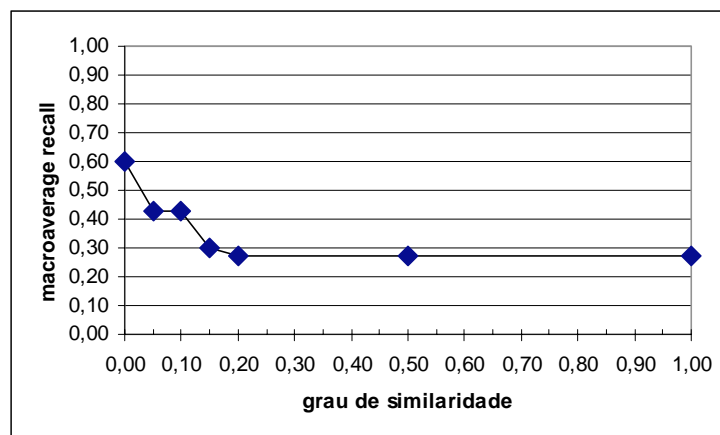


FIGURA 5.3 – Resultados de “macroaverage recall” utilizando o método “best-star”

Esse gráfico, assim como os outros (Figuras 5.4 e 5.5), indica que quanto maior for o GSM, ou seja, quanto maior a exigência de similaridade entre os objetos, mais difícil torna-se a identificação de relações entre os objetos. Além disso, identifica-se que quando o GSM é de 20% (0.2) o gráfico estabiliza-se, e nenhum dos algoritmos consegue encontrar relações entre os objetos (os objetos tendem a separar-se completamente). Este fato pode ser comprovado analisando-se os grupos identificados, onde se constata que cada grupo contém somente um documento. Isso significa que o grau máximo de similaridade entre os objetos encontra-se entre os valores 0.15 e 0.20.

Quanto à *precisão*, para qualquer valor de similaridade os grupos constituídos contêm somente documentos de um mesmo assunto. Com isso, o grau de *macroaverage precision* para todos os casos é de 100%.

Realizando-se uma análise dos grupos após o primeiro agrupamento descobre-se que, na verdade, o número de classes (assuntos) existentes é maior do que três. O

conjunto de documentos em inglês, por exemplo, é constituído de 3 documentos que tratam da ferramenta *Altavista*<sup>TM</sup> e 2 documentos diversos. Levando-se em conta estes dados e refazendo-se o agrupamento (com 0% de similaridade mínima) obtém-se um grau de *macroaverage recall* maior, de 0.8 pontos (80%). Isso indica que a análise *a priori*, onde foram identificadas três classes, é incorreta.

Com isso, fica explícito o fato da ferramenta poder ser utilizada com fins de descoberta de conhecimento. Pois, no caso, descobriu-se que o domínio é constituído de sub-áreas (subgrupos) diferentes dos imaginados *a priori*. No caso dos *e-mails*, o conjunto de documentos deveria ter sido dividido nos seguintes grupos: três documentos descrevendo a ferramenta *Altavista*<sup>TM</sup> (em inglês), dois documentos genéricos em inglês, quatro documentos contendo *chamadas* de artigos para congressos e duas relativas a uma lista de discussão.

Utilizando estes novos dados, os testes foram refeitos e novos resultados obtidos. Estes resultados são ilustrados na FIGURA 5.4, e apresentam o mesmo comportamento anterior (a mesma tendência), pois a linha resultante possui o mesmo aspecto. Porém, os graus de *macroaverage recall* são mais altos, e, portanto, melhores.

Aplicando-se o algoritmo *Stars* neste novo conjunto de dados obtém-se os valores indicados na FIGURA 5.5, que compara todos os algoritmos. Este algoritmo obtém resultados inferiores aos obtidos pelo algoritmo *best-stars*. Isso, porque, primeiramente, o algoritmo não separa os documentos quando o GSM utilizado é de zero por cento (0%), alocando-os em um único grupo. Devido a isso, o valor de *macroaverage recall* e *macroaverage precision* inicial é de zero pontos.

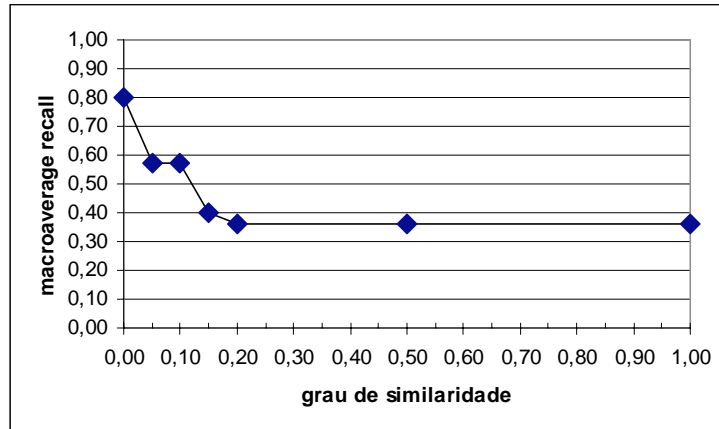


FIGURA 5.4 – Novos resultados de “macroaverage recall” utilizando o método “best-star”

Com um grau de similaridade mínimo de 0.05 (5%), o grau de *macroaverage recall* sobe para 0.6 pontos. Porém, o grau de *macroaverage precision* diminui para 0.93 pontos (o valor anterior de *precision* estava estimado em 1.0 ponto, conforme já informado anteriormente), pois documentos de assuntos diferentes são alocados em um mesmo *cluster*. Esse é o único caso em que isso acontece. Todos os demais graus de similaridade obtêm, neste algoritmo, *macroaverage precision* de 100%.

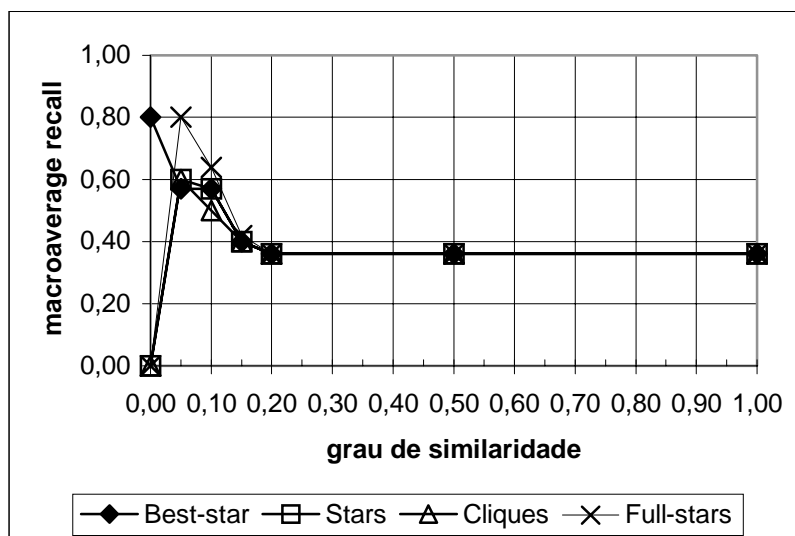


FIGURA 5.5 – Comparação dos resultados de “macroaverage recall”

O algoritmo *Cliques* apresenta resultados similares ao algoritmo *Stars*, porém, inferiores. Isso se deve ao fato deste algoritmo ser mais exigente, alocando elementos a um *cluster* somente se este também é similar a todos os outros elementos do mesmo *cluster*.

Conforme citado anteriormente, nesta coleção, os algoritmos que utilizam GSMs acima de 20% separam todos os documentos em grupos diferentes. A *macroaverage recall* mínima em todos estes casos é de 0.36 pontos. Aos 15% todos os algoritmos obtêm uma *macroaverage recall* de 0.4 pontos. Porém, com 10% o algoritmo *full-stars* já se destaca, obtendo o melhor resultado, já que cria grupos com todas as possibilidades de combinações entre documentos (esse desempenho torna-se melhor ainda, aos 5%). Estes dados podem ser confirmados na FIGURA 5.5, anterior. As informações sobre a quantidade de grupos identificados por cada algoritmo em cada GSM utilizado podem ser obtidas na FIGURA 5.6, seguinte.

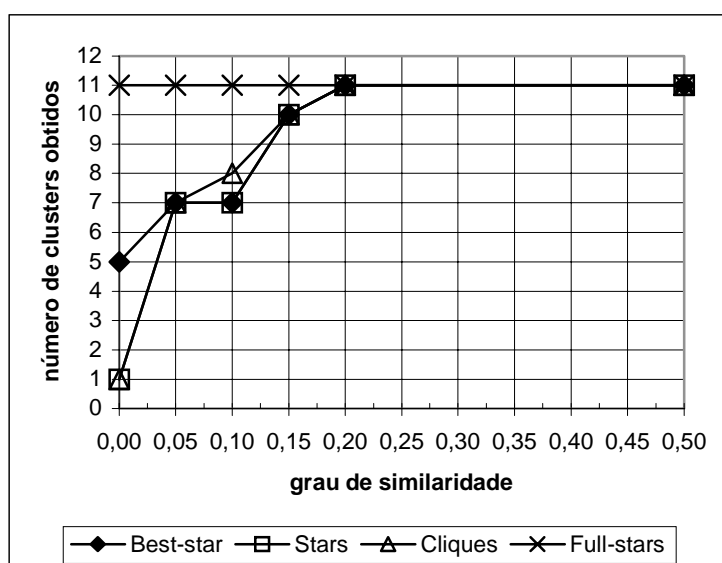


FIGURA 5.6 – Quantidade de “clusters” obtidos em cada algoritmo

De acordo com os resultados, o algoritmo *cliques* é o que obtém o pior desempenho, com uma *abrangência* de 0.5 pontos. Porém, estas métricas de *abrangência e precisão* não conseguem medir a quantidade de conhecimento que pode ser adquirido com o agrupamento, visto que são baseadas em análises *a priori* da coleção.

Em aplicações que tem como objetivo adquirir conhecimento, o algoritmo *cliques* é bem mais indicado, pois este algoritmo é mais restritivo, exigindo com que todos os documentos sejam similares entre si e constituindo grupos mais coesos. Porém, de acordo com os resultados obtidos, o algoritmo *Cliques* não é recomendado para fins de recuperação de informações.

Em geral o algoritmo *Full-stars* apresenta o melhor desempenho, principalmente em GSM pequenos. Porém, a quantidade de grupos identificados por este algoritmo é bem maior e muitos documentos são atribuídos a mais de um grupo simultaneamente. Justamente, por esse motivo, seu desempenho é maior (em muitos casos, o mesmo conjunto de elementos é alocado em mais de um grupo, porém, com ordem invertida).

A FIGURA 5.7 apresenta um gráfico comparativo contendo o desempenho médio de cada algoritmo em todos os pontos de GSM utilizados. Nesta figura, fica claro que o algoritmo que apresenta o melhor desempenho médio é o algoritmo *Best-star*, seguido do algoritmo *Full-stars*. O algoritmo *Stars* encontra-se na terceira posição, seguido do algoritmo *Cliques*, o último colocado.

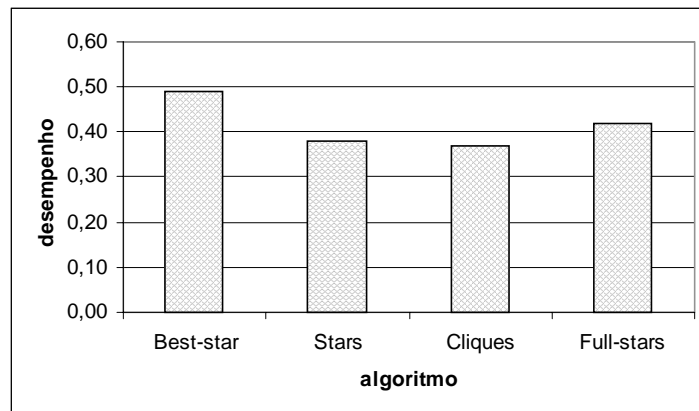


FIGURA 5.7 – Desempenho médio geral

É importante salientar que ao realizar-se uma análise mais detalhada dos grupos, considerando-se suas palavras principais (o centróide), consegue-se descobrir conhecimento a respeito *do porquê* dos grupos terem sido constituídos de uma ou outra forma. Com isso, verifica-se, algumas vezes, que os grupos aparentemente diferentes dos esperados possuem certa relação e são igualmente importantes. Deste modo, de maneira interativa, o usuário pode ir descobrindo novos conhecimentos (novos fatos) sobre os objetos (documentos) que ele está trabalhando.

Este é um dos pontos principais da ferramenta, ou seja, permitir que o usuário descubra novas relações e padrões (conhecimento) nos objetos que está trabalhando. É claro que este tipo de conhecimento adquirido é muito difícil de ser medido, tornando difícil a avaliação da ferramenta sob este aspecto. Devido a isso, a ferramenta foi avaliada somente em relação a sua capacidade de recuperação de informações.

## 5.2 Coleção de patentes

Além da coleção de documentos de *e-mail*, buscou-se simular um agrupamento de informações coletadas da *Internet*, retornadas como resposta de uma consulta a um SRI. Esse é mais um exemplo de utilização da ferramenta (e do método de agrupamento), uma etapa adicional à fase de recuperação de informações. Neste caso, os documentos retornados pela ferramenta de recuperação de informações (FRI) são agrupados de acordo com sua similaridade, facilitando a tarefa de identificação de documentos relevantes para o usuário.

A base de dados consultada pode ser encontrada na Internet no endereço <http://www.umi.com/proquest>, em um serviço denominado *ABI-Form*<sup>6</sup>. A consulta realizada visava obter informações sobre patentes na área de *Inteligência Competitiva na Empresa*. As palavras utilizadas como consulta foram: *Competitive Intelligence*, e 120 registros foram recuperados.

Cada um dos registros foi armazenado em um arquivo texto individual (num total de 120 arquivos-texto), cujos nomes variam entre CI9401.txt (primeiro registro) e CI94121.txt (ultimo registro).

Os registros possuem uma semi-estruturação (através de marcas – *tags*) que indica, entre outras informações, o *título do documento*, a *data de registro*, o *assunto*, o *autor* e um *resumo* descrevendo a patente registrada. Para que estas informações (*tags*) não influenciassem no processo de agrupamento, foram criadas duas categorias de *stopwords* específicas, contendo-as (categorias: *Patentes* e *SGML*). O objetivo do agrupamento, neste caso, é identificar os registros que tratam do mesmo assunto.

Além disso, buscava-se saber se os parâmetros de agrupamento oferecidos pela ferramenta são capazes de influenciar negativamente o processo de agrupamento. Neste caso, os parâmetros só puderam ser analisados em relação ao tempo de processamento, já que não havia uma análise *a priori* sobre os sub-conjuntos de dados existentes na coleção, inviabilizando o cálculo de *abrangência* e *precisão*. Nessa coleção a única informação que se tinha é que os documentos pertenciam ao assunto *Inteligência Competitiva*, e pretendia-se descobrir se havia algum padrão ou relacionamento entre os registros (documentos).

Os parâmetros testados foram: *uso de palavras negativas*, *uso de números*, *truncagem* e *filtragem*.

### 5.2.1 Análise dos diferentes parâmetros

Na TABELA 5.2 são listadas as diferentes opções (ou parâmetros) utilizados e os seus respectivos resultados, inclusive o tempo de processamento. Este tempo pode variar, tornando-se menor ou maior, de acordo com o equipamento utilizado<sup>7</sup>. Porém, como todos os testes foram realizados em um único equipamento, foi possível comparar os resultados, verificando se havia alguma mudança significativa quando os parâmetros eram modificados.

---

<sup>6</sup> Os dados utilizados nesse teste foram coletados e fornecidos pela professora Lília Maria Vargas do PPGA/UFRGS.

<sup>7</sup> O equipamento utilizado para realizar esses testes constitui-se de um microcomputador Pentium™ 133 com 96MB de memória.



TABELA 5.2 – Parâmetros e seus respectivos tempos de processamento

Coleção	Ignora		Pré-processamento	Tempo de processamento da Matriz			
	Números	Stopwords		Dias	Horas	Minutos	Segundos
Patentes1	SIM	SIM	SIM	0	0	13	27
Patentes2	NÃO	SIM	SIM	0	0	13	50
Patentes3	NÃO	NÃO	SIM	0	0	30	47
Patentes4	SIM	SIM	NÃO	0	6	23	15
Patentes5	NÃO	NÃO	NÃO	1	5	52	58

O primeiro teste foi realizado com a intenção de descobrir se a utilização das etapas de pré-processamento (truncagem e filtragem) afetaria significativamente o tempo de processamento.

Analisando-se a coleção Patentes5, que utiliza todas as palavras (já que não ignora números, palavras negativas e não há filtragem – ou seja, não há pré-processamento), nota-se que o tempo de processamento é demasiadamente grande<sup>8</sup>. Realizando-se a comparação deste tempo com o tempo de processamento da coleção Patentes3, que também utiliza números e palavras negativas mas realiza o pré-processamento, identifica-se que há uma melhora significativa em relação ao tempo de processamento. Neste caso, há uma diferença de vinte e nove (29) horas e trinta (30) minutos (mais alguns segundos), já que o número de palavras processadas diminuiu drasticamente, conforme a TABELA 5.3.

TABELA 5.3 – Quantidade de palavras por coleção

Coleção	Quantidade de palavras	Frequência total
Patentes1	31	5826
Patentes2	31	5836
Patentes3	29	5978
Patentes4	2784	516583
Patentes5	3770	962576

Com isso, conclui-se que as etapas de pré-processamento oferecem um ganho de desempenho (em relação ao tempo) muito grande nos testes realizados. Acredita-se que este ganho seja notado em outras coleções, desde que existam palavras a serem excluídas pelo pré-processamento. Com isso, um número bem menor de palavras (características) é utilizado para descrever e comparar os objetos.

Porém, devido ao fato das palavras negativas terem sido utilizadas em ambos os experimentos citados, e estas nada tinham a ver com o assunto abordado nos documentos (afetando negativamente a identificação de grupos), optou-se pela realização de mais dois testes (na mesma coleção), visando identificar se o uso de palavras negativas influencia no tempo de processamento.

Comparando-se o tempo de processamento do teste Patentes5 com o teste Patentes4, identifica-se que a utilização de palavras negativas torna o processo mais demorado, já que, como antes, um número maior de características deve ser analisado.

<sup>8</sup> Este tempo, dependendo do objetivo do agrupamento, pode ser irrelevante quando se obtêm resultados satisfatórios.

Com isso, conclui-se que o uso de palavras negativas torna o processo de agrupamento (cálculo de similaridades) mais demorado. Isso porque, com elas, há um número de características maior, o que aumenta o tempo de comparação e, conseqüentemente, o tempo de processamento.

Além do fato das palavras negativas tornarem o processo mais demorado, as palavras negativas são características que influenciam de forma negativa o processo de agrupamento. Com isso, aconselha-se que as palavras negativas sejam ignoradas em qualquer processo de agrupamento.

Os números também podem ser descartados em algumas coleções. Nestes casos a sua utilização pode afetar os resultados. Nos testes realizados, verificou-se que a utilização de números aumenta o tempo de processamento (ver Patentes1 e Patentes2), assim como as palavras negativas. Porém, nesta coleção, este fator não influenciou de forma significativa pelo fato de existirem poucos números com frequência suficiente para serem processados (já que as etapas de pré-processamento identificaram os números como sendo palavras menos freqüentes e eles foram ignorados).

## 5.2.2 Análise dos algoritmos

A TABELA 5.4 apresenta os resultados obtidos pelos quatro algoritmos implementados, utilizando um GSM de zero por cento (0%). A coleção utilizada neste teste foi a *patentes4*, que ignora palavras negativas.


Verifica-se nesta tabela que todos os algoritmos, com exceção do *Best-star*, não conseguem separar os documentos, colocando-os no mesmo grupo. O algoritmo *Full-stars* obteve 120 grupos de 120 documentos, pois identificou todas as possibilidades de grupos (como, neste caso, o GSM utilizado foi de 0%, o que significa que os documentos poderiam ser completamente diferentes, todos os grupos continham todos os documentos). Mais uma vez, como no estudo de caso anterior, os resultados mostram que o algoritmo *best-star* consegue identificar as melhores relações entre os objetos, não sendo necessário que o usuário tenha que preocupar-se com a identificação de um  GSM específico para a coleção. Ambos os casos mostram que basta deixar o GSM em zero por cento para obter resultados satisfatórios.

TABELA 5.4 – Resultados dos diferentes algoritmos

Algoritmo	GSM	Grupos obtidos	Média de documentos por grupo
Full Stars	0	120	120
Best Star	0	46	2.6
Cliques	0	1	120
Stars	0	1	120

Utilizando-se um grau de similaridade mais exigente, de cinquenta por cento (50%), obtêm-se os resultados apresentados na TABELA 5.5.

O algoritmo *Full-stars* obteve quatro grupos com mais de um elemento. São eles: grupo 46 (CI9430 e CI9435), grupo 47 (CI9431 e CI9436), grupo 50 (CI9435 e CI9430) e grupo 51 (CI9436 e CI9431). O grupo 46 e o grupo 50 são idênticos, com exceção da ordem dos elementos no grupo (ambos os grupos contém os mesmos documentos). Com os grupos 47 e 51 acontece o mesmo. Nestes dois grupos o tema

principal é o software *intelliseek* (este fato foi verificado analisando-se o *centróide* do grupo).

TABELA 5.5 – Resultados para GSM de cinquenta por cento

Algoritmo	GSM	Grupos obtidos	Quantidade de documentos por grupo
Full Stars	0.5	120	1.033
Best Star	0.5	118	1.017
Cliques	0.5	119	1.008
Star	0.5	118	1.017

Os demais algoritmos também identificaram os mesmos grupos de elementos (porém não duplicados). Um grupo contendo os elementos CI943 e CI9435, e outro com os elementos CI9436 e CI9431. Todos os demais grupos possuem um único elemento.

Isso demonstra que a relação existente entre os elementos desta coleção é muito baixa (em torno de 10%, conforme pode ser verificado analisando-se a matriz de similaridades da coleção), já que poucos grupos foram identificados com *GSM* superiores a este valor. Todos os testes realizados indicam que os graus de similaridade costumam ser muito baixo para qualquer coleção. Isso porque dificilmente encontram-se elementos com muita similaridade, mesmo que tratem do mesmo assunto, pois as palavras utilizadas costumam variar muito.

### 5.3 Coleção “reuters”

A coleção *Reuters* é uma coleção de documentos provenientes de artigos da *reuters newswire/1987*. Foi originalmente criada por funcionários do grupo *Reuters Ltd.* e *Carnegie Group Inc.*, que classificaram manualmente os artigos organizando-os em diversas categorias predefinidas.

Por volta de 1990, a coleção foi disponibilizada à comunidade científica. Nesta época a coleção denominava-se *Reuters-22173* pois era constituída de 22.173 documentos. Com o passar dos anos a coleção passou a ser utilizada por diversos pesquisadores. Observando a possibilidade de poder comparar seus resultados, a coleção tornou-se padrão entre os pesquisadores que poderiam então validar seus estudos e algoritmos.

Em 1996 *David Lewis* realizou uma série de alterações na coleção, refinando e reorganizando a coleção. Com isso, alguns documentos foram excluídos e várias anomalias corrigidas. A partir de então a coleção passou a chamar-se *Reuters-21578*, correspondendo à nova quantidade de documentos.

A coleção pode ser livremente distribuída e utilizada para fins de pesquisa e estudo, desde que a publicação que a utilize indique seu nome (*Reuters-21578, distribution 1.0*) e o local onde pode ser encontrada (<http://www.research.att.com/~lewis>).

### 5.3.1 Formato da coleção

A coleção constitui-se de vinte e dois arquivos de dados no formato SGML, cada qual contendo 1.000 artigos (com exceção do último que possui 578 artigos). Além disso, seis arquivos descrevendo as categorias utilizadas para indexar os artigos também estão inclusos na coleção.

Os artigos estão dispostos em ordem cronológica, cada um possuindo um identificador próprio. Das diversas *marcações (TAGs)* SGML que os arquivos contém as mais importantes para o presente trabalho são apresentadas a seguir. As demais TAGs podem ser obtidas no arquivo *readme.txt* que acompanha a coleção.

h) `<Reuters TOPICS = ?? LEWISSPLIT = ?? CGISPLIT = ?? OLDID = ?? NEWID = ??> ... </Reuters>` - Identificador de início de documento (artigo).

Os parâmetros são os seguintes:

- TOPICS – Indica se o documento foi categorizado manualmente ou não. Valores: *yes* (o documento foi atribuído a, pelo menos, uma categoria), *no* (o documento não foi atribuído a uma categoria, talvez pelo fato de não ser relevante a uma das categorias predefinidas) e *bypass* (o documento não foi indexado);
- LEWISSPLIT – Indica se o documento faz ou não parte dos experimentos realizados por *David Lewis*. Valores: *Training* (utilizado na coleção de treinamento), *Test* (utilizado na coleção de teste do experimento) e *Not-used* (não foi utilizado);
- CGISPLIT – Serve para indicar os documentos que foram utilizados nos experimentos de *Hayes*. Valores: *Training-set* (utilizado no treinamento) e *Published-testset* (utilizado nos testes);
- OLDID – O número de identificação do documento na coleção *Reuters* anterior;
- NEWID – O número de identificação atual do documento;

Os demais atributos podem ser desconsiderados.

i) `<TOPICS> ... </TOPICS>`, `<PLACES> ... </PLACES>`, `<PEOPLE> ... </PEOPLE>`, `<ORGS> ... </ORGS>`, `<EXCHANGES> ... </EXCHANGES>` - Indicam a lista de categorias do documento (se existirem). Cada categoria é definida pela *tag* `<D> ... </D>`.

j) `<COMPANIES> ... </COMPANIES>` - Mesmo que as *tags* anteriores, porém não há categorias de companhias (portanto aparecem adjacentes).

k) `<TEXT Type = ??> ... </TEXT>` - Delimitam o material textual dos artigos. Possui o atributo TYPE que indica:

- NORM – O texto possui uma estrutura normal;
- BRIEF – O texto é um pequeno comentário de 2 ou 3 linhas;
- UNPROC – Texto não usual, o que dificulta sua estrutura.

- 1) <BODY> ... </BODY> - Corpo do texto do documento.

### 5.3.2 Utilizando a coleção

Para a coleção *Reuters* existem 5 conjuntos diferentes e predefinidos de categorias (*Topics, Exchange, Orgs, People e Places*). Os documentos foram indexados e enquadrados manualmente nestas categorias. Dos cinco conjuntos de categorias existentes, o mais utilizado, nos testes de classificação de documentos realizados até o momento, é o conjunto *TOPICS*. Este conjunto abrange assuntos de interesse econômico, num total de 39 assuntos ou subcategorias. As subcategorias deste conjunto são descritas nos arquivos *all-topics-strings.lc.txt* e *cat-descriptions\_120396.txt* que acompanham a coleção.

A maioria dos estudos que utilizam a coleção *Reuters* realiza classificação. Como já dito anteriormente, a classificação de informações exige duas etapas: uma de aprendizado (ou treinamento) e uma de classificação propriamente dita (também chamada de fase de testes). Com isso, para que o método de classificação não seja influenciado, geralmente adota-se um conjunto de documentos específicos para a fase de treinamento e outro conjunto de documentos para a fase de testes (o objetivo disso é tornar os testes mais realistas, já que o algoritmo é testado em uma coleção diferente da coleção que o treinou).

A coleção *Routers*, por ter sido utilizada em vários experimentos, já possui subdivisões ou sub-coleções elaboradas visando às etapas do processo de classificação. Como alguns estudos importantes já foram realizados com esta coleção, recomenda-se que estudos subsequentes utilizem uma das subdivisões já analisadas (é importante salientar que estudos só podem ser comparados se utilizarem o mesmo conjunto de dados para treinamento e teste). As subdivisões mais conhecidas e trabalhadas da coleção *routers* são três: LEWIS SPLIT, APTE SPLIT e HAYES SPLIT. Os detalhes de como cada uma destas subdivisões foram obtidas são descritos no arquivo *readme.txt* que acompanha a coleção.

A sub-coleção adotada neste trabalho é a HAYES SPLIT pois apresenta o menor subconjunto de dados para teste<sup>9</sup>. Além disso, esta sub-coleção foi utilizada em um sistema de classificação chamado *Construe*, desenvolvido pelo grupo *Carnegie Group Inc* a pedido do próprio grupo *Routers*. Os resultados são apresentados no trabalho de *David Lewis* [LEW 91], e podem ser comparados.

Esta sub-coleção constitui-se de 20,856 documentos de treinamento e 722 documentos para teste. Como o agrupamento não necessita de uma fase de treinamento (pois este não é seu objetivo), somente o conjunto de testes foi utilizado.

Por possuírem uma semi-estruturação, dada pela utilização das *Tags SGML* apresentadas anteriormente, os arquivos da coleção *Routers* podem ser analisados mais facilmente. Estruturas de detalhe, como por exemplo as estruturas iniciais de descrição do documento (contendo o assunto a que pertence e número de identificação), podem e foram ignoradas no estudo realizado. Isso porque a utilização destas estruturas pode vir a influenciar nos resultados, já que a palavra que identifica o assunto que o documento trata aparece nestas estruturas. Devido a isso, somente o texto demarcado pela *tag*

<sup>9</sup> Tentou-se realizar alguns testes com outras subcoleções, mas, devido ao excessivo número de documentos, os testes tornaram-se inviáveis. Optou-se por utilizar estas outras subcoleções em trabalhos futuros, que venham a refinar os algoritmos e a ferramenta, tornando-os mais eficazes.

“*Body*” (*corpo do texto*) foi utilizado nos testes. As demais estruturas foram retiradas dos arquivos de teste.

O algoritmo de agrupamento utilizado nos testes foi o *Best-star* utilizando um grau de similaridade mínima correspondente à zero, já que os resultados dos estudos de caso anteriores demonstram que este algoritmo utilizando este parâmetro obtém os melhores resultados. Os grupos de palavras negativas utilizados neste experimento correspondem às classes: *consoantes*, *english*, *sgml* e *vogais*.

### 5.3.3 Análise dos resultados

Os resultados obtidos no sistema CONSTRUE, citado por Lewis [LEW 91], são obtidos através das métricas de *microaverage recall* e *microaverage precision*. Deste modo, para que os resultados aqui obtidos possam ser avaliados, estas métricas também foram adotadas neste experimento.

Para a métrica de *microaverage*, conforme David Lewis, considera-se a coleção como um todo, não realizando os cálculos intermediários de *abrangência* ou *precisão* para obter o valor final. Em casos de classificação de informações a *microaverage recall* é calculada dividindo-se o número de documentos que o método conseguiu classificar pelo número de documentos que deveriam ser classificados. Neste estudo de caso este valor é obtido dividindo-se o número de documentos agrupados pelo número de documentos que deveriam ser agrupados. Considerar-se-á um documento como *agrupado* quando ele for alocado a um grupo de diversos documentos. Todo o documento que for alocado a um grupo isolado será considerado como não agrupado (não classificado)<sup>10</sup>.

Já a *microaverage precision* é obtida dividindo-se o número de documentos atribuídos corretamente pelo número total de documentos efetivamente atribuídos (independente do fato de terem sido atribuídos corretamente ou não).

Nos experimentos realizados, um documento é considerado atribuído corretamente a um grupo, quando ele for encontrado em um grupo cuja maioria dos elementos pertençam ao mesmo assunto (à mesma categoria). O elemento que diferir do tópico indicado pela maioria dos elementos é considerado *mal atribuído*. Além disso, adotou-se que um elemento atribuído a um grupo de somente um elemento (ou seja, ele mesmo), ao mesmo tempo em que seu tópico abrange mais elementos, é considerado *mal atribuído*. Os elementos que não possuem tópico, mas são atribuídos a classes de um único elemento, são considerados corretamente atribuídos. Porém, quando atribuídos a uma classe com mais de um elemento também são considerados *mal atribuídos*.

De um total de 722 documentos, 266 grupos foram obtidos (com uma média de 2,7 documentos por grupo). O tempo de processamento do agrupamento pelo método *Best-star* (utilizado nos testes) foi de uma (1) hora e sete (7) minutos. Este tempo já é mais aceitável em aplicações reais que utilizem uma grande quantidade grande de objetos (documentos), principalmente porque a quantidade de conhecimento (descoberta de associações e padrões) que pode ser obtida é muito grande e compensadora (conforme será demonstrado a seguir).

<sup>10</sup> Para esta regra há uma exceção: se o documento atribuído a um grupo isolado pertencer a uma classe que contenha somente um elemento (ou seja, ele próprio) então ele deve ser considerado como agrupado. Além disso, caso o documento isolado não tenha sido atribuído a uma classe pelo processo manual ele também é considerado como agrupado (pois permanece isolado, sem classe ou grupo).

A TABELA 5.6 apresenta o tempo de processamento (cálculo) da matriz de similaridades para os 722 documentos da coleção. Esse tempo de processamento é bastante elevado, o que dificulta a utilização da ferramenta em aplicações que exijam processamento imediato ou de tempo real. Apesar disso, uma vez realizada esta etapa a matriz de similaridades fica armazenada, podendo ser utilizada, várias vezes, por diversos métodos de agrupamento (e que utilizem diversos parâmetros). Com isso, as etapas tornam-se independentes e a matriz pode ser calculada previamente.

TABELA 5.6 – Tempo de processamento da matriz de similaridades

Coleção	Ignora		Pré-processamento	Tempo de processamento da Matriz			
	Números	Stopwords		Dias	Horas	Minutos	Segundos
HAYES1	SIM	SIM	SIM	0	13	25	44

O grau de *microaverage recall* obtido nos experimentos foi de 0.95 pontos (687 documentos agrupados). Esse valor é bem maior do que o valor obtido pelo sistema *CONSTRUE* que obteve 0.89 pontos. Já o grau de *microaverage precision* obtido foi de 0.43 pontos, um valor bem menor do que o obtido pelo sistema *CONSTRUE*, de 0.92 pontos.

Supõe-se que esta queda no grau de *microaverage precision* seja ocasionada pelo método adotado para cálculo desta medida, já que esta medida é mais apropriada para medir desempenho de classificação de informações (na classificação só há um fator a ser considerado: ou o documento é classificado corretamente ou não), e neste caso a coleção pode ser considerada como um todo onde se consegue identificar facilmente os documentos classificados e os documentos não classificados.

Já no agrupamento, os documentos são separados em grupos de documentos similares, o que dificulta a abordagem da coleção como um todo. Neste caso são necessárias regras ou restrições que identifiquem quando um documento foi classificado e ainda se esta classificação foi correta.

A abordagem tomada neste estudo para considerar um documento como sendo classificado não causa maiores problemas, tanto que o grau de abrangência foi elevado. Porém, identificar o grau de precisão (definir se um documento foi classificado corretamente ou não) não é tarefa trivial, pois é necessário identificar se os documentos foram colocados em grupos corretos.

Nos testes realizados, convencionou-se que o grupo correto seria aquele em que todos os documentos pertenceriam à mesma categoria (definida manualmente *a priori*). Por outro lado, muitas destas categorias são muito similares, pertencendo ao mesmo assunto, mas foram consideradas diferentes pelas pessoas que indexaram os documentos. Para exemplificar, nos experimentos realizados vários documentos do assunto *gás natural* (*natural gas*) foram agrupados com documentos do assunto *combustível* (*fuel*). Dependendo do nível de abrangência que se está procurando, pode ser interessante que os documentos que tratam do assunto *gás natural*, *gasolina*, *álcool* ou *óleo diesel* sejam retornados em um mesmo grupo (ou seja, sejam considerados da mesma classe). Em se tratando do assunto combustível, todos pertencem à mesma classe. Porém, caso deseje-se identificar os documentos que tratem dos diferentes *tipos de combustível*, a separação destes documentos pode ser necessária.

Estes níveis de abrangência ou detalhamento não são levados em conta pela indexação manual, e esse fator influenciou muito nos testes realizados. Além disso, a

ferramenta descobriu uma associação (conhecimento) interessante: gás natural é um combustível.

Outro fator que influi negativamente no grau de *microaverage precision* obtido é a identificação de muitos grupos com documentos não indexados (ou seja, não categorizados *a priori*). Cerca de 139 documentos não indexados foram encontrados em diversos grupos de documentos pertencentes a alguma categoria. E, neste caso, conforme as restrições impostas, os documentos sem categoria são considerados incorretamente classificados, diminuindo a precisão. Apesar deste fato influir negativamente na métrica utilizada, ele demonstra que a ferramenta é capaz de identificar relacionamentos que a indexação manual não foi capaz de identificar, já que elementos não indexados foram agrupados em algumas das categorias existentes.

Com isso, apesar da ferramenta ter obtido um baixo grau de *microaverage precision*, seu método não deve ser desmerecido. Além de descobrir relacionamentos entre categorias de documentos (descoberta de conhecimento associativo), o método conseguiu agrupar documentos que não possuíam categorias (documentos não indexados *a priori*), identificando sua categoria. Estes dois fatores são extremamente importantes pois auxiliam no processo identificação de classes de documentos e descoberta de conhecimento.



## 6 Conclusões

Neste capítulo apresentar-se-ão as contribuições oferecidas pelo estudo realizado, assim como se discutirá as aplicações que podem ser dadas ao método elaborado, suas restrições e sugestões para que este possa ser melhorado.

Como maior contribuição, este trabalho apresenta um estudo da aplicação de técnicas de agrupamento de objetos em informações (documentos) textuais. Todo o levantamento bibliográfico, aqui apresentado, é de grande utilidade e valia, e serve de embasamento para que futuros trabalhos sobre o mesmo tema possam se aprofundar.

Como resultado deste estudo, elaborou-se um método capaz de aplicar a técnica de agrupamento em documentos (informações textuais). Este método, de acordo com os resultados obtidos nos estudos de caso, apresenta resultados satisfatórios e pode ser utilizado tanto para fins de recuperação de informações (organização de documentos por similaridade) quanto para a descoberta de conhecimento (descoberta de associações entre documentos e assuntos, identificação de padrões e elaboração de regras).

Além do método, outra grande contribuição deste trabalho é a ferramenta de agrupamento desenvolvida (apesar deste não ter sido o objetivo principal deste trabalho). De acordo com o levantamento bibliográfico realizado, existem poucas ferramentas de *agrupamento de documentos* (uma destas ferramentas é apresentada em [CUT 92]). Estas ferramentas, apesar de realizarem o agrupamento, realizam-no de forma restrita, requisitando que o usuário indique o número de grupos que ele deseja obter ou realizando somente o método de agrupamento hierárquico.

Com este estudo, descobriu-se também que grande parte das ferramentas já desenvolvidas (e não só as ferramentas, mas também os estudos) trabalha com o enfoque de classificação de documentos – uma etapa posterior ao processo de agrupamento. Estas ferramentas possuem o seu mérito, porém, existem aplicações que necessitam realizar uma análise da coleção de documentos sem que se tenha conhecimento prévio desta coleção. Isso só é possível no agrupamento, já que a classificação exige conhecimento prévio sobre as classes nas quais os documentos podem ser classificados. Além disso, a classificação não prevê (na maioria dos casos) a existência de documentos que não se encontrem em alguma das classes existentes. Os documentos só podem ser classificados em uma das classes existentes.

O agrupamento permite com que novas classes sejam descobertas, já que consegue agrupar documentos mesmo que estes não pertençam a assuntos conhecidos. Isso porque não há essa necessidade de conhecimento prévio sobre os assuntos (ou os possíveis assuntos dos documentos). Os assuntos ou as classes dos objetos (documentos) sendo agrupados são descobertos após o agrupamento, em um processo de análise dos grupos obtidos.

Ainda em relação à ferramenta, mesmo que outras ferramentas sejam encontradas em estudos não analisados (em decorrência do tempo disponível para este estudo, ou da dificuldade de obtenção dos respectivos dados), ela apresenta um diferencial interessante: ela não necessita que o usuário especifique um número  $x$  de grupos a serem encontrados. Os grupos de documentos vão sendo encontrados no momento da análise, sem restrições (ou melhor, a única restrição é o grau mínimo de similaridade). Além disso, analisando-se os estudos de caso, a ferramenta dispõe do método de agrupamento *best-star* (outra das contribuições deste estudo, a ser analisada

posteriormente), que não necessita de um grau de similaridade mínimo para identificar relações entre os documentos. Inclusive, nos dois primeiros casos estudados, este fator fez com que os melhores grupos de documentos tivessem sido identificados.

Além disso, a ferramenta permite que o usuário utilize todo o texto no processo de análise, e não somente algumas partes ou estruturas (como, por exemplo, títulos e resumos). Isso faz com que o resultado seja mais preciso, já que todas as características do texto podem ser utilizadas. Este fator, porém, torna o processo de agrupamento mais demorado (já que uma quantidade maior de características deve ser analisada).

Outra característica, similar a anterior, corresponde ao fato da ferramenta poder utilizar todas as palavras (todo o vocabulário) de um documento, e não somente algumas palavras de um domínio específico (predefinido ou controlado).

Caso estas opções sejam consideradas irrelevantes ou inadequadas para determinado experimento, elas podem ser ignoradas ou modificadas. Isso porque a ferramenta permite que sejam definidos conjuntos de palavras a serem excluídos do processo (as palavras negativas). Estas podem ser palavras relativas à estrutura do documento ou palavras relativas ao vocabulário (contexto) dos documentos sendo analisados, e podem ser excluídas. É possível também limitar o número de características utilizadas (diminuindo o tempo e a qualidade do agrupamento), utilizando um limiar de limite máximo (quantidade de palavras) ou um limiar de importância mínima (grau de importância – frequência – da palavra no documento).

Outra grande contribuição deste trabalho é o estudo comparativo dos algoritmos estudados. No capítulo de estudo de casos, há alguns experimentos que testam os diferentes algoritmos estudados (*Cliques*, *Stars*, *Full-stars* e *Best-star*) e indicam o comportamento destes algoritmos, quando aplicados em objetos textuais. Apesar de existirem diferenças sutis em muitos dos resultados, identifica-se que o algoritmo *best-star* (outra das contribuições deste trabalho) apresenta melhor desempenho dentre os algoritmos estudados.

O algoritmo desenvolvido (*Best-star*) é o único capaz de identificar grupos de elementos similares quando o GSM escolhido pelo usuário é de zero por cento. Além disso, neste caso, obtém o melhor desempenho (0.8 pontos de *recall*) no estudo de caso realizado (coleção de documentos de correio eletrônico). Com isso, apresenta-se à comunidade científica um algoritmo de agrupamento que não necessita que o usuário especifique um limiar de similaridade mínima. Ou seja, o algoritmo não necessita que o usuário preocupe-se em manipular parâmetros de agrupamento nem especifique um número  $x$  de grupos a serem identificados (dois grandes problemas dos métodos de agrupamento existentes).

Assim, o usuário não necessita ter conhecimento prévio algum da coleção ou do método de agrupamento sendo empregado (uma das proposições do agrupamento é justamente não requisitar conhecimento prévio dos objetos para separá-los ou agrupá-los). Portanto, mesmo com a utilização de um grau de similaridade zero como limiar (onde todos os documentos são considerados iguais), as relações naturais entre os elementos são encontradas.

A separação das etapas de agrupamento também é uma grande contribuição, já que minimiza o tempo de agrupamento. Isso porque existem etapas que após serem realizadas uma vez não necessitam ser realizadas novamente. Esse é o caso da etapa de cálculo da matriz de similaridades. Após esta matriz ter sido calculada, basta aplicar o algoritmo de agrupamento (restrições na matriz). Esse processo é muito mais rápido,

quase que instantâneo. A matriz só precisa ser recalculada se os parâmetros iniciais foram alterados (pré-processamento e stopwords, por exemplo). Por outro lado, se o processo de agrupamento for considerado como um todo, ele é demorado.

Por último, a fórmula *fuzzy* de cálculo de similaridade de objetos que foi aplicada em documentos também é considerada uma contribuição. Na verdade este trabalho utilizou-se de duas fórmulas *fuzzy*: uma para cálculo de similaridade entre documentos e outra para identificação da importância de cada palavra no documento. A segunda fórmula é utilizada pela primeira para que esta possa identificar os graus de similaridade. A junção das duas fórmulas, assim como sua aplicação com os fins de agrupamento de objetos, são as contribuições deste trabalho.

Até o momento foram discutidas as questões técnicas do método desenvolvido, não sendo feita nenhuma referência aos aspectos práticos. Na seção seguinte são discutidos estes aspectos práticos, onde se comenta como este método (e a ferramenta) desenvolvido pode ser utilizado em aplicações práticas.

Na seção seguinte discute-se como o método e a ferramenta desenvolvidos podem ser aplicados aplicações práticas.

## 6.1 Aplicações possíveis

As aplicações descritas nesta seção demonstram a importância do agrupamento de objetos textuais. O objetivo básico do agrupamento é separar documentos em grupos de documentos similares. Logo, toda a aplicação que necessita deste tipo de organização é auxiliada pelo agrupamento. O agrupamento de documentos facilita a organização de informações (visando à recuperação de informações) e a análise textual (visando a descoberta de conhecimento associativo, por exemplo).

A recuperação de informações é facilitada porque o método desenvolvido consegue processar uma grande quantidade de documentos (de assuntos diversos) e agrupá-los em *clusters* de documentos de assuntos similares. Estando organizados em grupos de documentos de assuntos similares, o processo de localização (e recuperação) de informações torna-se facilitado. O escopo (nível de abrangência e precisão) de cada grupo pode ser definido pelo usuário, e corresponde ao limiar GSM.

Um dos grandes problemas do GSM é que ele varia de acordo com a coleção sendo trabalhada (grande parte dos limiares variam de acordo com o experimento), não existindo um GSM genérico. Portanto, o usuário deve realizar diversos experimentos, modificando o limiar, até identificar a melhor organização (os melhores grupos) para a coleção que ele está trabalhando (por outro lado, mostrou-se que com o algoritmo *best-star* esse limiar não é tão importante, podendo ser sempre zero). É possível afirmar que quanto menor for este limiar, menos similares os documentos precisam ser, ocasionando um aumento na abrangência do grupo (aumentando o número de documentos), diminuindo porém sua precisão (documentos de outros assuntos passam a entrar no mesmo grupo). Do mesmo modo, quanto maior for o limiar mais exigente o processo torna-se, diminuindo a abrangência e aumentando a precisão. Um limiar muito alto pode fazer com que cada documento seja atribuído em um grupo diferente (ou seja, cada documento tornar-se-á um *cluster*).

Um exemplo prático de organização de informações é idealização de uma ferramenta de manipulação de mensagens eletrônicas (email). É possível utilizar o

método de agrupamento em um conjunto de mensagens eletrônicas com o objetivo de agrupá-las por assunto. Um programa *leitor de e-mails* com esta capacidade é capaz de identificar o assunto de cada mensagem recebida e colocá-la em seu respectivo *folder* (pasta). Estes *folders* podem ser predefinidos pelo usuário de acordo com os assuntos de seu interesse ou com a organização que ele preferir (por exemplo: congressos, informações institucionais, lista de discussão). A recuperação destas informações torna-se mais fácil, já que permanecem organizadas de acordo com os assuntos que o próprio usuário estabelecer.

Da mesma forma, uma ferramenta específica de recuperação de informações que se utilize da técnica de agrupamento é capaz de recuperar informações e organizá-las de forma que os documentos sejam apresentados para o usuário dispostos em grupos que tratam do mesmo assunto. Com isso, o usuário pode deter-se no grupo cujo assunto for mais aproximado do assunto que ele está necessitando, facilitando a recuperação. O grupo selecionado pode ser reorganizado (num novo processo de agrupamento local), onde subcategorias são encontradas e o processo de identificação do grupo mais adequado é repetido. Esse processo pode ser aplicado recursivamente até que o usuário encontre o documento ou o conjunto de documentos mais adequado à sua necessidade.

Nestes dois exemplos vê-se que o entendimento dos documentos (ou melhor, a identificação do assunto dos documentos) torna-se mais prática. Essa facilidade permite com que técnicas de *análise textual* sejam aplicadas.

Técnicas de análise textual, segundo *Moscarola* [MOS 98], servem para facilitar a pesquisa e a descoberta de conhecimento, conseqüentemente facilitando a leitura e favorecendo a interpretação de textos muito grandes ou conjuntos de textos. Esse tipo de técnica oferece, portanto, uma redução considerável de esforço.

A *sumarização* é um exemplo de técnica de análise textual voltada à descoberta de conhecimento. Através de um sumário (resumo) contendo as palavras estatisticamente mais importantes de um grupo de documentos (o *centróide*, por exemplo), é possível identificar o assunto abordado por este conjunto de documentos sem que seja necessário ler (efetivamente) cada documento do grupo. Estas técnicas partem do princípio de que as palavras mais importantes oferecem uma visão geral do conteúdo de um texto ou conjunto de documentos (Este princípio também é utilizado por aqueles que pregam a eficiência da leitura dinâmica). Neste caso, as palavras consideradas mais importantes são aquelas que aparecem em maior quantidade e, portanto, aquelas estatisticamente mais relevantes.

Estas palavras mais importantes (o *centróide*) são consideradas como a *tendência do cluster*, e indicam quais as características mais marcantes de determinado grupo de documentos – o padrão de um grupo. Com essa idéia de *padrão*, a técnica de agrupamento somada a técnica de análise textual pode ser aplicada em diversos campos com o objetivo de identificação de características marcantes em determinado conjunto.

Para exemplificar, a aplicação destas técnicas em um contexto jurídico, onde existe uma quantidade muito grande de processos para serem analisados, seria de grande valia. Isso porque o tempo que um Juiz leva para analisar um processo é geralmente muito grande. Com o método de agrupamento vários processos similares são agrupados em um único grupo. Neste caso, os detalhes mais insignificantes (que muitas vezes tornam o processo mais demorado) podem ser ignorados (basta limitar o número de características e pegar somente aquelas mais importantes – dois parâmetros facilmente manipuláveis no processo de agrupamento), tornando o processo mais prático.

Após, com uma técnica de *sumarização*, o *Juiz* é capaz de identificar o padrão deste conjunto de processos (identificar suas características mais marcantes) e verificar que realmente tratam de problemas similares. Com isso, bastaria aplicar a mesma sentença a todos os processos. É claro que este exemplo é muito otimista e extremista. A tarefa do agrupamento poderia ser minimizada (por questões éticas) e o processo utilizado somente para que o *Juiz* localizasse processos similares ao que ele está julgando e então, através das características principais, encontrar mais rapidamente uma solução para o caso.

Similarmente ao caso jurídico, médicos podem utilizar o método de agrupamento para enquadrar prontuários médicos em classes de prontuários similares. Com isso, o conjunto de pacientes com a mesma doença do caso que o médico está tratando é identificado e localizado. Consequentemente, o mesmo tratamento (exames e remédios) utilizado no conjunto de prontuários encontrado pode ser aplicado, já que são da mesma classe.

Outro exemplo, puxando para o lado comercial, é a possibilidade de aplicação do método de agrupamento em empresas. Neste caso as técnicas de análise textual seriam aplicadas com o objetivo de identificar as atividades desenvolvidas pela concorrência, além de auxiliar no processo de monitoração estratégica e competitiva da empresa. Para obter estas informações qualquer documento disponível pode ser utilizado (folhetos de propaganda, páginas *WEB*, bancos de dados). Essa é uma atividade muito em evidência atualmente, já que a ação de qualquer empresa deve ser globalizada. Neste caso, segundo *Moscarola* [MOS 98], toda informação (que os outros não possuem) é uma vantagem.

Em todos os exemplos citados há um objetivo comum utilizando uma metodologia básica. Essa metodologia comum realiza, em todos os casos, a identificação de grupos de objetos similares para que se possa descobrir suas características comuns (padrões). Neste caso, o conhecimento descoberto é a própria associação entre os objetos (que *a priori* não era conhecida) e também as características comuns entre os objetos. Todo esse processo, uma vez dominado, pode ser aplicado em qualquer campo do saber humano.

## 6.2 Sugestões e trabalhos futuros

Durante o desenvolvimento deste estudo, alguns pontos deixaram de ser cobertos. Isso ocorre não só neste, mas em todo trabalho científico, pois nem todos os problemas podem ser resolvidos de uma única vez. Alguns destes problemas devem ser deixados de lado para que outros, mais marcantes, sejam resolvidos.

Grande parte dos aspectos não resolvidos neste trabalho diz respeito à implementação do método desenvolvido. Muitas facilidades e refinamentos não foram implementados devido à complexidade do método, optando-se pelos métodos mais rápidos de implementação sem objetivar a otimização dos algoritmos. Mas, como ferramenta de apoio à comprovação do método elaborado, o programa desenvolvido pôde ser utilizado (isso significa que a ferramenta deve ser refinada para que possa ser utilizada com fins comerciais).

Antes de analisar os detalhes técnicos de implementação que devem ser aprimorados, uma sugestão importante é a aplicação efetiva do método elaborado em uma das aplicações sugeridas na seção anterior. Um exemplo seria aplicar a ferramenta

em documentos de uma empresa, visando à identificação de algum conhecimento que possa tornar esta empresa mais competitiva. Estas são situações cotidianas reais que servem para comprovar tanto a eficiência quanto à utilidade real do método.

Outra sugestão importante é a adaptação do método para que ele possa ser utilizado em nível de classificação de informações. Neste caso, depois de identificados os grupos de documentos similares, novos documentos poderiam ser agrupados nos grupos já existentes. Aqui, há uma preocupação muito importante: os grupos devem ser mantidos coesos durante sua manutenção, ou seja, quando um novo documento for adicionado, ele não deve alterar o enfoque do grupo. Caso sua relação com um dos grupos existentes não seja encontrada, um grupo adicional deve ser criado. No trabalho de Charikar [A 97] é apresentado um algoritmo que busca realizar a manutenção de grupos já existentes, levando em conta muitas destas considerações.

Em relação às facilidades que a ferramenta poderia possuir, além de sugestões e alternativas de implementação, destacam-se os seguintes aspectos:

- a) Permitir que outros formatos de texto (e não só o ASCII) sejam utilizados. Isso porque em aplicações reais os documentos existentes estão em padrões muito mais avançados (*Postscript, MSWord, Acrobat, HTML*) e isso deve ser transparente para o usuário;
- b) A ferramenta foi elaborada visando informações não estruturadas. Porém, em alguns casos, o usuário possui textos que contém delimitadores ou campos conhecidos como títulos ou seções específicas. Nestes casos, o usuário poderia optar pelo agrupamento de documentos que utilizasse somente uma destas estruturas (campos), ao invés do documento inteiro. Isso, não só diminuiria o tempo de processamento, como, também, evitaria que certas informações contidas no texto influenciassem de forma negativa no processo;
- c) Permitir que o usuário selecione diferentes fórmulas de cálculo de similaridade entre objetos. Essa opção pode dificultar o uso da ferramenta para usuários finais, mas pode auxiliar um pesquisador a determinar o melhor método (permitindo comparações);
- d) Do mesmo modo, implementar outros métodos de agrupamento e também a forma hierárquica;
- e) Estudar e implementar outras formas de cálculo e seleção de características importantes, diferentes da frequência relativa. Uma sugestão seria levar em conta a posição sintática da palavra (substantivos e verbos seriam considerados mais relevantes) ou a sua localização no texto (em títulos ou citações). Com isso, talvez sejam identificadas características mais marcantes nos objetos, o que poderia vir a diminuir o número de características que descrevem cada objeto, tornando o processo de agrupamento (cálculo da matriz de similaridades) mais rápido;
- f) Criar outras formas de visualização de grupos e adicionar mais informações sobre os grupos identificados. Como exemplo, a informação sobre o grau

médio de similaridade entre os objetos de um *cluster* seria útil, pois indicaria o grau de coesão do grupo. Gráficos e informações estatísticas mais detalhadas também são uma sugestão importante.

- g) Modificar a visualização de *centróide* dos grupos (botão *lexicography*), oferecendo mais opções de manipulação léxica (manipulação de palavras) e adicionando uma visualização gráfica do relacionamento entre as palavras (como pode ser feito na opção *refine* da ferramenta *Altavista*<sup>TM11</sup>);
- h) Refinar os algoritmos existentes, melhorando sua performance;
- i) Realizar experimentos com outras coleções de dados, reavaliando o método e aprimorando-o, inclusive publicando os resultados em âmbito científico.

---

<sup>11</sup> Endereço Internet: <http://altavista.digital.com>

**Anexo 1 – Hyperdictionary: A knowledge discovery  
tool to help information retrieval**



**Anexo 2 – Recuperação de informações usando a  
expansão semântica e a lógica difusa**

## Bibliografia

- [ANI 97] ANICK, Peter; VAITHYANATHAN, Shivakumar. Exploiting clustering and phrases for context-based information retrieval. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1997. **Proceedings...** New York: Association for Computing Machinery, 1997. p.314-323.
- [BER 97] BERRY, Michael J. A; LINOFF, Gordon. **Data mining techniques: for marketing, sales and customer support.** New York : John Wiley & Sons, 1997. p.187-215.
- [CHA 98] CHÁVEZ, Edgar. **Algoritmos para detección de cúmulos en nubes de datos.** [S.l.:s.n.], 1998. Curso sobre Recuperacion de Informacion oferecido nas Jornadas Iberoamericanas de Informatica, 4., Bolívia, 1998.
- [CHA 97] CHARIKAR, Moses et al. Incremental clustering and dynamic information retrieval. In: SYMPOSIUM ON THEORY OF COMPUTING, STOC, 1997. **Proceedings...** New York: Association for Computing Machinery, 1997. p.626-635.
- [CHE 96] CHEN, Hsinchun et al. **A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the worm community system.** Tucson, Arizona: Management Information System Department Department, University of Arizona, 1996. Disponível por WWW em <http://ai.bpa.arizona.edu/papers/wcs96.html> (12/02/1999).
- [COW 96] COWIE, Jim; LEHNERT, Wendy. Information extraction. **Communications of the ACM**, New York, v.39, n.1, p.80-91, 1996.
- [CRO 94] CROSS, Valerie. Fuzzy information retrieval. **Journal of Intelligent Information Systems**, Boston, v.3, n.1, p.29-56, 1994
- [CUT 92] CUTTING, Douglass et al. Scatter/Gather: a cluster-based approach to browsing large document collections. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1992. **Proceedings...** New York: Association for Computing Machinery, 1992. p.318-329.
- [CUT 93] CUTTING, Douglass et al. Constant interaction-time scatter/gather browsing of very large document collections. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1993. **Proceedings...** New York: Association for Computing Machinery, 1993. p.126-134.
- [EVE 74] EVERITT, Brian. **Cluster Analysis.** New York: John Wiley & Sons, 1974.
- [FEL 97] FELDMAN, Ronen; HIRSH, Haym. Exploiting background information in knowledge discovery from text. **Journal of Intelligent Information Systems**, Boston, v.9, n.1, p.83-97, July/Aug 1997.
- [HAN 96] HAN, Jiawei et alli. Intelligent query answering by knowledge discovery techniques. **IEEE Transactions on Knowledge and Data Engineering**, Washington, v.8, n.3, p.373-390, July 1996.

- [KOW 97] KOWALSKI, Gerald. **Information retrieval systems: theory and implementation**. Boston : Kluwer Academic Publishers, 1997. 282p.
- [KOR 97] KORFHAGE, Robert R. **Information retrieval and storage**. [S.l.]: John Wiley & Sons, 1997.
- [KUP 95] KUPIEC, Julian at al. A trainable document summarizer. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1995. **Proceedings...** New York: Association for Computing Machinery, 1995. p.68-73.
- [LEW 91] LEWIS, David Dolan. **Representantion and Learning in Information Retrieval**. Amherst: University of Massachusetts, Department of Computer and Information Science, 1991. Phd Thesis.
- [MOS 98] MOSCAROLA, Jean et al. **Technology watch via textual data analysis**. France: Université de Savoie, Le Sphinx Développement, 1998. (Note de Recherche, n. 98-14).
- [NG 97] NG, Hwee et al. Feature selection, perceptron learning, and a usability case study for text categorization. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1997. **Proceedings...** New York: Association for Computing Machinery, 1997. p.67-73.
- [OLI 96] OLIVEIRA, Henry M. **Seleção de entes complexos usando lógica difusa**. Porto Alegre: Instituto de Informática da PUC-RS, 1996. Dissertação de mestrado.
- [PED 93] PEDRYCZ, Witold. Fuzzy neural networks and neurocomputations. **Fuzzy Sets and Systems**, Washington, v.56, n.1, p.01-28, 1993.
- [RIJ 79] RIJSBERGEN, C. van. **Information retrieval**. 2.ed. London: Butterworths, 1979.
- [SAL 83] SALTON, G.; MCGILL, M. J. **Introduction to modern information retrieval**. New York: McGraw-Hill, 1983.
- [SCH 97] SCHÜTZE, Hinrich; SILVERSTEIN, Craig. projections for efficient document clustering. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1997. **Proceedings...** New York: Association for Computing Machinery, 1997. p.74-81.
- [SIL 97] SILVERSTEIN, Craig; PEDERSEN, Jan. Almost-constant-time clustering of arbitrary corpus subsets. In: SPECIAL INTEREST GROUP ON INFORMATION RETRIEVAL, SIGIR, 1997. **Proceedings...** New York: Association for Computing Machinery, 1997. p.60-66.
- [WIE 96] WIEBE, Janyce; HIRST, Graeme; HORTON, Diane. Language use in context. **Communications of the ACM**, New York, v.39, n.1, p.102-111, Jan. 1996.
- [WIL 88] WILLET, Peter. Recent Trends In Hierarchic Document Clustering: A Critical Review. **Information Processing & Management**, [S.l.], v.24, n.5, p.577-597, 1988.
- [WIV 96] WIVES, Leandro K. **Um Modelo de Hiperdicionário: Estudo de Caso em Prontuários Médicos**. Pelotas: UCPEL, 1996. Trabalho de diplomação.

- [WIV 97] WIVES, Leandro K. **Um Estudo Sobre Técnicas de Recuperação de Informações com ênfase em Informações Textuais**: Trabalho Individual. Porto Alegre: CPGCC da UFRGS, 1997. (TI-672).
- [WIV 98a] WIVES, Leandro K; LOH, Stanley. Hyperdictionary: a knowledge discovery tool to help information retrieval. In: STRING PROCESSING AND INFORMATION RETRIEVAL – A SOUTH AMERICAN SYMPOSIUM, SPIRE, 1998. **Proceedings...** Washington: IEEE Press, 1998.
- [WIV 98b] WIVES, Leandro K; LOH, Stanley. Recuperação de informações usando a expansão semântica e a lógica difusa. In: CONGRESO INTERNACIONAL EN INGENIERIA INFORMATICA, ICIE, 1998. **Proceedings...** Buenos Aires: Universidad de Buenos Aires, 1998.
- [ZAD 73] ZADEH, Lotfi A. Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transactions on Systems, Man and Cybernetics**, Washington, v.SMC-3, n.1, p.28-44, Jan. 1973.