UNIVERSIDADE FEDERAL FLUMINENSE – UFF
INSTITUTO DE COMPUTAÇÃO – IC
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CASSIOPEIA: UM MODELO BASEADO EM SUMARIZAÇÃO E APRENDIZADO AUTÔNOMO USADO EM AGRUPAMENTOS PARA DESCOBERTA DE CONHECIMENTO EM BASES TEXTUAIS

MARCUS VINICIUS CARVALHO GUELPELI

NITERÓI
2009

MARCUS VINICIUS CARVALHO GUELPELI

CASSIOPEIA: UM MODELO BASEADO EM SUMARIZAÇÃO E APRENDIZADO AUTÔNOMO USADO EM AGRUPAMENTOS PARA DESCOBERTA DE CONHECIMENTO EM BASES TEXTUAIS

Proposta de Tese de Doutorado apresentada ao Curso de Pós-Graduação em Computação da Universidade Federal Fluminense. Área de concentração: Otimização Combinatória e Inteligência Artificial

Orientadora: Ph.D. Ana Cristina Bicharra Garcia

Niterói (RJ), JANEIRO de 2009.

MARCUS VINICIUS CARVALHO GUELPELI

CASSIOPEIA: UM MODELO BASEADO EM SUMARIZAÇÃO E APRENDIZADO AUTÔNOMO USADO EM AGRUPAMENTOS PARA DESCOBERTA DE CONHECIMENTO EM BASES TEXTUAIS

Proposta de Tese de Doutorado apresentada ao Curso de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito para obtenção do Grau de Doutor. Área de concentração: Otimização Combinatória e Inteligência Artificial

BANCA EXAMINADORA:

_____

Ana Cristina Bicharra Garcia, Ph.D. – Orientadora
UFF – Universidade Federal Fluminense

_____

Flavia Cristina Bernardini, DSc.
UFF – Universidade Federal Fluminense

_____

Bianca Zadrozny, Ph.D
UFF – Universidade Federal Fluminense

_____

Alexandre Plastino de Carvalho, DSc.
UFF – Universidade Federal Fluminense

_____

Luis Miguel Parreira e Correia, Ph.D
UL-Universidade de Lisboa

PGC
universidade federal fluminense
instituto de computação
programa de pós-graduação em computação

Ata de Exame de Proposta de Tese de Doutorado

Ata da sessão de julgamento do Exame de Proposta de Tese de Doutorado em Computação, área de concentração: Otimização Combinatória e Inteligência Artificial, do aluno Marcus Vinicius Carvalho Guelpeli, realizada no dia 29 de janeiro de 2009.

Aos 29 dias do mês de janeiro de dois mil e nove, às 14h, na sala de seminários do Instituto de Computação reuniu-se a Banca Examinadora designada na forma regimental pela Coordenação do Curso, para julgar o Exame de Proposta de Tese de Doutorado, intitulada "Proposta de um modelo para descoberta e aprendizado de conhecimentos em bases textuais", prestado pelo aluno Marcus Vinicius Carvalho Guelpeli, como parte dos requisitos necessários à obtenção do grau de Doutor em Computação, área de concentração Otimização Combinatória e Inteligência Artificial. A Banca Examinadora foi presidida pela Profa. Ana Cristina Bicharra Garcia, tendo como membros efetivos os professores Luís Miguel Parreira e Correia, Flávia Cristina Bernardini,Bianca Zadrozny e Alexandre Plastino de Carvalho. Aberta a sessão pública, o aluno teve a oportunidade de expor o trabalho. Após a exposição, o aluno foi arguido oralmente pelos membros da Banca, tendo demonstrado suficiência de conhecimento e capacidade crítica para a condução de um trabalho de doutoramento. A Banca concluiu pela **aprovação** do aluno. E, nos termos do Regulamento Geral dos Cursos de Pós-Graduação desta Universidade, foi lavrada a presente ata, lida e julgada, conforme vai assinada pelos membros da Banca Examinadora e pelo candidato. Niterói, 29 de janeiro de 2009.

Banca Examinadora:

Profa. Dra.Ana Cristina Bicharra Garcia, UFF – Presidenta

Prof. Dr.Luís Miguel Parreira e Correia, Universidade de Portugal

Profa. Dra.Flávia Cristina Bernardini, UFF

Profa. Dra.Bianca Zadrozny, UFF

Prof. Dr.Alexandre Plastino de Carvalho, UFF

Aluno:

Marcus Vinicius Carvalho Guelpeli

RESUMO

O volume de informações não estruturadas vem crescendo de forma desordenada nos dias atuais. A internet, repositório destas informações em grandes partes textuais é o grande agente facilitador de novos conhecimentos. Esta estrutura, quase anárquica, trouxe consigo um grande problema de organização, surgido ante a dificuldade do ser humano de armazenar grande quantidade de informações e depois recuperá-las, ocasionando uma sobrecarga de informação. A área de Descoberta de Conhecimento em Textos, cuja principal finalidade é obter algum tipo de conhecimento em documentos textuais, torna explícito o conhecimento implícito.

Este trabalho propõe a criação de um modelo denominado *Cassiopeia,* cujo objetivo é proporcionar a descoberta de conhecimento em bases textuais em domínios distintos e/ou antagônicos usando um *Clusterizador,* que utiliza a Sumarização e o Aprendizado Autônomo para a obtenção destes conhecimentos ao longo do tempo. Com a proposta do modelo *Cassiopeia,* pretende-se obter a melhora da coesão dos agrupamentos e viabilização do problema de alta dimensionalidade na descoberta de conhecimento em bases textuais.

O modelo *Cassiopeia* é composto por três módulos: Sumarização, *Clusterização* e Aprendizado Autônomo.

O módulo de Sumarização é baseado *em perfis pragmáticos.* Este módulo usa um algoritmo *perfil* que enfatiza a importância da palavra dentro de cada sentença como índice de desenvolvimento gramatical. Desta forma consegue determinar o nível de compressão automaticamente para a geração do sumário e a diminuição do volume de palavras para a *clusterização*.

Outro módulo que integra o modelo *Cassiopeia* é o módulo de *Clusterização,* que utiliza metodologia de Similaridade Difusa para minimizar a questão de ambiguidades da linguagem natural. Através da análise do grau de similaridade dos textos a serem *clusterizados,* pode-se determinar pertinência de cada palavra em cada índice que compõe os agrupamentos. Desta forma, consegue-se definir o melhor momento que o algoritmo deve parar.

Por fim, o módulo de Aprendizado Autônomo, que usa o Processo Decisório de Markov, composto por técnicas de aprendizado por reforço e um algoritmo de aprendizado de máquina (*Q Learning*). O objetivo desse módulo é aprender ao longo do processo, o índice de compressão do sumário e o número de vezes que ocorre a *reclusterização*.

ABSTRACT

The volume of non-structured informations has been growing in nowadays because internet is an information warehouse and a large provider of new knowledge. However, the internet is growing unordered, and a huge part of this information is textual. This structure almost anarchical brought an organization problem when human being tries to store this information and after recover it. This problem is called overload information. Text mining, specifically in knowledge discovery context, aims to obtain some kind of knowledge in textual documents, turning implicit into explicit knowledge.

This work proposes a model called *Cassiopeia* whose main purpose is produce the knowledge discovery in textual databases using in distinct or antagonics using clusterizing that uses the Summarization Autonomous Learning to get knowledge for a long time. Through of *Cassiopeia* model the objective is get improved cohesion of groups and make viable the problem about high dimensionality in knowledge discovery in textual databases.

The *Cassiopeia* model is composed of three modules: Summarization, Categorization and Autonomous Learning.

The Summarization module purpose is to summarize the texts based on pragmatic profiles. This module uses an algorithm, called *Perfil* that emphasizes the importance of each word into each sentence as an index of grammatical development. By this way, this algorithm can determine the compression level of the summary and further volume decrease of words to categorization. The Categorization module is responsible to categorize the texts into categories. This module uses fuzzy similarity to minimize ambiguities in natural language. After, the algorithm responsible to construct

the categories determines the number of words into each *cluster* through the similarity degree analysis of the texts to be categorized, defining its cut point. The Autonomous Learning module uses Hidden Markov Models to statistical analysis, and a learning algorithm to discover the knowledge in textual documents along the process. In this proposal, we intend to obtain learning curves, improving the recall and precision measures of the categories, obtained as a result of the application of the *Cassiopeia* model in knowledge discovery from text bases process.

The proposed model aims to show, through simulations, the obtained gain when using the integrated approaches of knowledge discovery (poli-structured model) that compose the *Cassiopeia* model, mentioned before: Summarization, Categorization and Autonomous Learning modules.

SUMÁRIO

LISTA DE ABREVIATURAS 6

LISTA DE FIGURAS 7

LISTA DE TABELAS 1

1. INTRODUÇÃO 1

1.1. MOTIVAÇÃO 4

1.2. PROBLEMA 6

1.3. HIPÓTESE 6

1.4. CONTRIBUIÇÕES 6

1.5. METODOLOGIA DE PESQUISA 7

1.6. CRONOGRAMA 8

1.7. ESTRUTURA DA PROPOSTA 9

2. ESTADO DA ARTE DOS C*LUSTERIZADORES* DE TEXTO 11

2.1. CLUSTERIZADORES 11

2.2. CLUSTERIZADORES DE TEXTOS 11

2.3. CONSIDERAÇÕES FINAIS 16

3. MODELO CASSIOPEIA 17

3.1. APRESENTAÇÃO DO MODELO CASSIOPEIA 17

3.2. SUMARIZAÇÃO 19

3.2.1. OBJETIVOS 19

3.2.2. CONCEITO DE SUMARIZAÇÃO AUTOMÁTICA 20

3.2.3. AQUISIÇÃO DE UM PERFIL PRAGMÁTICO 22

3.2.4. PROPOSTA DE UM MÉTODO DE SUMARIZAÇÃO AUTOMÁTICA 24

3.2.5. ALGORITMO PERFIL 26

3.2.6. CONSIDERAÇÕES FINAIS 26

3.3. CLUSTERIZAÇÃO 27

3.3.1. OBJETIVOS **27**

3.3.2. CONCEITO DE *CLUSTERIZAÇÃO* **28**

3.3.3. ALGORITMOS DE *CLUSTERIZAÇÃO* PROPOSTOS NA LITERATURA **34**

3.3.4. PROPOSTA DE UMA ABORDAGEM PARA *CLUSTERIZAÇÃO* DE TEXTOS USANDO SIMILARIDADE DIFUSA **38**

3.3.5. FASES DO CLUSTERIZADOR **43**

3.3.6. CONSIDERAÇÕES FINAIS **45**

3.4. APRENDIZADO AUTÔNOMO **45**

3.4.1. APRENDIZADO AUTÔNOMO **45**

3.4.2. PROCESSO DECISÓRIOS DE MARKOV – PDM **48**

3.4.2.1. PAPEL DO CRÍTICO NO PDM **49**

3.4.2.2. CRITÉRIOS DE DESEMPENHO PARA UTILIDADE DE ESTADO **50**

3.4.2.3. MAXIMIZAÇÃO DA UTILIDADE DOS ESTADOS **51**

3.4.3. TÉCNICAS DE APRENDIZADO POR REFORÇO **51**

3.4.4. ALGORITMO Q-LEARNING **53**

3.4.5. CONSIDERAÇÕES FINAIS **56**

4. RESULTADOS PRELIMINARES **57**

4.1. MÓDULO SUMARIZAÇÃO **57**

4.2. MÓDULO DE *CLUSTERIZAÇÃO* **64**

4.3. CONSIDERAÇÕES FINAIS **69**

5. CONSIDERAÇÕES FINAIS **70**

6. REFERÊNCIAS BIBLIOGRÁFICAS **71**

LISTA DE ABREVIATURAS

MT	***M***ineração de ***T***exto
KDT	***K****nowledge* ***D****iscovery from* ***T****exts*
EI	***E***xtração de ***I***nformação em Texto
RI	***R***ecuperação de ***I***nformação em Texto
SA	***S***umarizador ***A***utomático
CASSIOPEIA	***C****lustering* ***A****lgorithm by* ***S****election of* ***S****core* ***I****ndexes by* ***O****ccurrence and* ***P****ercentual* ***I****nclusion* ***A****pproach*
HMM	***H****idden* ***M****arkov* ***M****ode*
PDM	***P***rocesso ***D***ecisórios *de* ***M****arkov*
AR	***A***prendizado por ***R***eforço

LISTA DE FIGURAS

FIGURA 1.PROPOSTA DO MODELO *CASSIOPEIA*. 17
FIGURA 2. ESTRUTURA DO SUMARIZADOR AUTOMÁTICO USANDO O ALGORITMO PERFIL. 25
FIGURA 3. DENDOGRAMA DO MÉTODO HIERÁRQUICO AGLOMERATIVO. 32
FIGURA 4. DENDOGRAMA DO MÉTODO HIERÁRQUICO DIVISIVO. 33
FIGURA 5.GRAFO DO ALGORITMO STAR. 36
FIGURA 6.GRAFO DO ALGORITMO *CLIQUES*. 37
FIGURA 7.AS FASES DO *CLUSTERIZADOR* USANDO SIMILARIDADE DIFUSA. 42
FIGURA 8.MODELO DINÂMICO E SEUS ELEMENTOS. 46
FIGURA 9.MODELO DE TRANSIÇÃO PDM. 49
FIGURA 10.UM AGENTE AR INTERAGINDO COM SEU AMBIENTE. 52
FIGURA 11.TABELA DOS PARES (ESTADO - AÇÃO). 54
FIGURA 12.TAXONOMIA DO *CORPUS TEMÁRIO*. 58

LISTA DE TABELAS

TABELA 1.CRONOGRAMA DAS ATIVIDADES A SEREM REALIZADAS. 9

TABELA2.TRABALHOS CORRELATOS NA ÁREA DE DESCOBERTA DE CONHECIMENTO USANDO *CLUSTERIZAÇÃO*. 13

TABELA 3.REPRESENTA A INTERAÇÃO DAS REGRAS DE ESTILO E IMPLICAÇÕES NO CONTEÚDO DO SUMÁRIO SEGUNDO (HOVY, 1988).....23

TABELA 4.MATRIZ DE SIMILARIDADE. 30

TABELA 5.RESULTADO DAS MÉDIAS E DO DESVIO PADRÃO DA APLICAÇÃO DO ALGORITMO *TEXTTILING* E *PERFIL* NO *CORPUS TEMÁRIO* COM TODAS AS MEDIDAS.....60

TABELA 6.COMPARATIVO ENTRE OS ALGORITMOS *TEXTTILING* E *PERFIL* USANDO O TESTE T DE VARIÂNCIA DIFERENTE. 61

TABELA 7.RESULTADO DAS MÉDIAS E DO DESVIO PADRÃO DA APLICAÇÃO DO ALGORITMO *PERFIL*, *GISTSUMM* E *WORD* NO *CORPUS TEMÁRIO* COM TODAS AS MEDIDAS. 62

TABELA 8.COMPARATIVO ENTRE OS *PERFIL* E *GISTSUMM* USANDO O TESTE T DE VARIÂNCIA DIFERENTE. 63

TABELA 9.RESULTADO DAS MÉDIAS E DO DESVIO PADRÃO DA APLICAÇÃO DO *EUREKHA* E *CLUSTERIZADOR* USANDO RESUMOS GERADOS PELOS MÉTODOS *PERFIL*, *GISTSUMM* E *WORD* NO *CORPUS TEMÁRIO*.....67

TABELA 10.COMPARATIVO ENTRE *EUREKHA* E *CLUSTERIZADOR* USANDO RESUMOS GERADOS PELOS MÉTODOS *PERFIL*, *GISTSUMM* E *WORD* NO *CORPUS TEMÁRIO* USANDO O TESTE T DE VARIÂNCIA DIFERENTE 68

1. Introdução

O crescimento das informações textuais, oriundos em grande parte pelas facilidades encontradas hoje em gerar e armazenar informações em meios eletrônicos, e a dificuldade posterior de recuperar estas informações, proporcionou o surgimento do que é denominado de sobrecarga de informações (WIVES, 1999).

O aumento do volume de informações não estruturadas é característico nos dias atuais devido à internet, que por ser um grande repositório de informações e um grande gerador de novos saberes, vem aumentado, diariamente, as informações textuais de forma desordenada. Esta estrutura quase anárquica trouxe consigo um grande problema de organização (de informações), que surge mediante dificuldade do ser humano em armazenar grande quantidade de informações.

Entretanto, há algum tempo, a organização dessas informações textuais não era tida como importante. Porém, estudos provaram o contrário, mostrando que estas informações não estruturadas, a partir de uma organização eficaz, poderiam ser utilizadas de forma inteligente em vários campos, possibilitando tirar alguma vantagem competitiva ou dar suporte a tomada de decisões (EBECKEN, 2003).

A partir desta visão competitiva e organizacional, emergiu a área de Mineração de Textos (*Text Mining*) - MT, que tem como principal finalidade extrair padrões ou inferir algum tipo de conhecimento de um conjunto de textos. A MT é a execução de vários processos em várias etapas, de forma sequencial e interativa, que transforma ou organiza uma quantidade de

documentos em uma estrutura sistematizada. Isso possibilita, posteriormente, a sua utilização de forma eficiente e inteligente (WIVES, 2002).

A MT é um campo novo e multidisciplinar que inclui conhecimentos de áreas como: Informática, Estatística, Linguística e Ciência Cognitiva, inspirada pela Mineração de Dados (*Data Mining*), que procura descobrir padrões emergentes de banco de dados estruturados. A mineração de textos pretende extrair conhecimentos úteis de dados não estruturados ou semi-estruturados.

Existem três grandes áreas da MT segundo Loh (1999), Loh (2001) e Wives (2002): Extração de Informação em Texto (EI), Recuperação de Informação em Texto (RI) e Descoberta de Conhecimento em Texto (*Knowledge Discovery from Texts* - KDT).

A área de extração de informação procura solucionar o problema de encontrar informações específicas dentro dos textos e converter dados não-estruturados em estruturados. Na maioria das vezes, os dados extraídos dos textos são armazenados em bancos de dados e são utilizadas técnicas apropriadas de mineração de dados para obter algum tipo de conhecimento. Desta forma, pode-se dizer que a EI isola as partes relevantes do texto, extrai informações destas partes, transformando-as em informações explícitas.

A área de recuperação de informação é considerada como sinônimo de recuperação de documentos e, hoje em dia, mais precisamente de recuperação de texto. A RI encontra documentos com conteúdo de informação através de uma consulta feita pelo usuário que seja relevante a sua necessidade de informação, porém, nem sempre o usuário sabe que tipo de informação está procurando ou como realizar esta consulta. O resultado é uma quantidade

muito grande de documentos irrelevantes para anseio de conhecimento do usuário, ocasionando, segundo Wives (2002), sobrecarga de informação.

A descoberta de conhecimento em textos (FELDMAN *et al*, 1997), (TAN, 1999), (POTTENGER *et al*, 2001) e (KEOGH *et al*, 2002), está relacionada ao processo de recuperar, filtrar, manipular e resumir o conhecimento retirado de grandes fontes de informações textuais, e apresentá-lo para o usuário final de maneira útil e nova, utilizando-se de diversos recursos, geralmente diferentes dos originais. KDT é o resultado da simbiose entre recuperação de informação, aprendizagem de máquina, estatística e banco de dados (DAGAN *et al*, 1996) e (HEARST, 1997).

A KDT analisa conhecimentos de coleções que compreendem grandes volumes de documentos textuais não estruturados; com a finalidade de identificar as principais categorias. Assim, procura-se transformar conhecimento implícito em conhecimento explícito (FAYYAD e UTHURUSAMY, 1999). A KDT tem uma grande ligação evolutiva com área de RI. Existem várias abordagens comuns nas duas áreas, porém a apresentação da informação é diferente na área de KDT, uma vez que busca o aspecto da não sobrecarga de informação fazendo com que o conhecimento útil e o novo possam contribuir para satisfazer as necessidades de informação do usuário (FELDMAN *et al*, 1997).

Na literatura existem vários tipos de abordagens de KDT (LOH, 1999) e (LOH, 2001): descoberta por extração de passagem, descoberta por análise linguística, descoberta por análise de conteúdo, descoberta por sumarização, descoberta por associação entre passagens, descoberta por listas de conceitos-chave, descoberta de estruturas de textos, descoberta tradicional após

extração, descoberta por c*lusterização*, descoberta por descrição de classes de textos, descoberta por recuperação de informações, descoberta por associação, descoberta por hipertextos, descoberta por manipulação de formalismos, descoberta por combinação de representações e descoberta por comparação de modelos mentais.

Neste trabalho, o foco principal é a área de descoberta de conhecimento por c*lusterização*. Entretanto, para reduzir o tamanho dos textos a serem manipulados no processamento da c*lusterização* é realizado o processo de sumarização de texto.

1.1. Motivação

O acesso aos meios de informação está cada vez mais facilitado. Motivadas pela grande disponibilidade de recursos computacionais e a facilidade de troca e armazenamento de informações. As instituições e pessoas têm produzido e armazenado eletronicamente uma grande quantidade de informações que ficam dispostas de forma desorganizada e não padronizada, dificultando a sua localização e acesso, sendo que muitas dessas informações estão no formato textual.

Essas informações textuais são lançadas diariamente na internet, formando grandes coleções de informações, tais como: relatórios (diversos), especificações de produtos, relatórios de erros e as mensagens de advertências de software, resumos, notas, correspondência eletrônica, documentos diversos (boletins, jornais, revistas, etc.) e publicação eletrônica textual (bibliotecas virtuais, acervos documentais variados, entre outros) (HAN E KAMBER, 2001).

Deve-se usar métodos computacionais que agrupem automaticamente os documentos textuais disponíveis, a fim de recuperar informações com mais agilidade e fidelidade (com relação ao conteúdo dos textos) para que possam ser úteis. Existem vários sistemas, cujo objetivo é tornar racionais e rentáveis, social e economicamente, o armazenamento e o processamento sistêmico da informação. Algumas metodologias contribuíram para o surgimento de sistemas computacionais capazes de adquirir novos conhecimentos, novas habilidades e novas maneiras de organizar o conhecimento existente (MITCHEL, 1997).

A KDT vem tornando possível transformar esse grande volume de informações, geralmente não estruturadas, em conhecimento útil, muitas vezes inovador para quem busca algum tipo de conhecimento na internet ou em outras bases de informações textuais.

O uso da KDT permite obter conhecimento a partir de informações textuais. Tecnologicamente, o apoio do KDT acontece na transformação do conteúdo de repositórios de informações em conhecimento útil e novo a ser analisado e compartilhado pelas instituições, pessoas e, principalmente, pelas empresas para facilitar a tomada de decisões (VELICKOV, 2004).

Por fim, soluções advindas da área de KDT poderão minimizar a sobrecarga de informação, ocasionadas pela disponibilidade de grandes volumes de informações e pela falta de ferramentas adequadas para encontrar informações úteis ou desejadas.

O modelo *Cassiopeia* apresentado neste trabalho para descoberta de conhecimento textual usa as abordagens de sumarização e c*lusterização* e propõe a utilização do Aprendizado Autônomo. Ressalta-se que não foi

encontrado nas referências bibliográficas pesquisadas o uso em conjunto destas duas abordagens da área de descoberta de conhecimento textual e mais ainda, viabilização de todo o processo em KDT de forma não interativa com o uso do Aprendizado Autônomo, o que vem sustentar a necessidade e a motivação desta pesquisa.

1.2. Problema

Os *clusterizadores* de textos não apresentam boa coesão em *corpus* sem restrição de domínio e geram espaços com alta dimensionalidade.

1.3. Hipótese

Um *clusterizador* de texto que inclui um processo de sumarização e um processo de aprendizado autônomo para ajuste dos parâmetros da sumarização e da *clusterização*, viabiliza o problema de alta dimensionalidade e melhora a coesão dos seus agrupamentos em *corpus* sem restrição de domínio.

1.4. Contribuições

Com o uso do modelo *Cassiopeia,* que será apresentado neste trabalho, cita-se algumas contribuições:

- Criar um modelo que possibilite a descoberta de conhecimento em bases textuais, em domínios distintos e/ou antagônicos;
- Viabilizar com o uso de sumarização, o problema de alta dimensionalidade na descoberta de conhecimento em bases textuais;

- Contribuir com um modelo para área de KDT que possibilite o aprendizado autônomo de forma não interativa;
- Propor um modelo poliestruturado de KDT onde estejam em conjunto a abordagem de sumarização e a *clusterização*.

1.5. Metodologia de pesquisa

A metodologia adotada para a pesquisa a ser usada em todo o doutorado consta de:

- Pesquisa bibliográfica: visa constantemente atualizar o referencial bibliográfico da tese. A pesquisa bibliográfica para Tese de Doutorado será constantemente atualizada. Deve ser observado que esta pesquisa está situada na mineração de textos, dentro da área de descoberta de conhecimento em texto (FELDMAN *et al*, 1997), (TAN, 1999), (POTTENGER, 2001) e (KEOGH, 2002); com foco em c*lusterização*, especificamente c*lusterização* de texto (WIVES 2004), (LOPES, 2004), (PASARIN, 2005), (LOH *et al*, 2006) e (STRUBE, 2008).

- Desenvolvimento do modelo: o modelo *Cassiopeia* é composto por três módulos: a Sumarização; a *Clusterização* e o Aprendizado Autônomo. Estes módulos serão integrados no modelo *Cassiopeia*. A explicação e seus objetivos encontram-se formalizados no Capítulo 3, no qual é apresentado formalmente o modelo *Cassiopeia*.

- Avaliação: para mensurar o modelo *Cassiopeia* usar-se-á a implementação do teste de viabilidade, que será realizado em *corpus* sem restrição de domínio e de língua. Para isto, será utilizada a métrica de coesão de agrupamentos.

- Resultados Parciais de Publicações da Pesquisa: esta metodologia de pesquisa já obteve quatro publicações que estão descritas em detalhes no Anexo 01.

1.6. Cronograma

A seguir são descritas as atividades (Tabela 1) a serem realizadas para atingir os objetivos do doutorado em andamento.

ATIVIDADE 1: Entrega da proposta de qualificação de tese e preparação para qualificação.

ATIVIDADE 2: Iniciar as revisões propostas pela banca de qualificação. Avaliar e programar PDM, validando ou refutando sua eficiência em predições estatísticas para a descoberta do conhecimento em documentos textuais, como também no Aprendizado Autônomo, usando o algoritmo *Q-Learning*.

ATIVIDADE 3: Implementar o modelo de Aprendizado Autônomo no modelo *Cassiopeia*. Elaborar as estratégias de simulação. Validar estatisticamente os resultados obtidos. Buscar a viabilidade da realização do doutorado no exterior (doutorado sanduíche).

ATIVIDADE 4: Escrita final da tese de doutorado e preparação para a defesa da tese. Durante a realização das atividades, serão realizadas outras tarefas tais como: escrita de artigos científicos a serem submetidos a periódicos, congressos nacionais e internacionais da área.

O tempo de duração é calculado em semestre.

Tabela 1.Cronograma das atividades a serem realizadas.

Período	***Atividades***			
	1	2	3	4
1/2009	✓	✓		
2/2009		✓	✓	
1/2010			✓	✓
2/2010				✓

1.7. Estrutura da Proposta

Esta proposta está estruturada como segue:

Capítulo 2 – Estado da Arte dos *clusterizadores* de texto

Nesse capítulo são apresentados conceitos sobre *clusterização* e *clusterização* de texto, além de trabalhos correlatos.

Capítulo 3- Modelo *Cassiopeia*

É apresentada a arquitetura do modelo *Cassiopeia* e seus módulos de Sumarização, *Clusterização* e Aprendizado Autônomo.

Capítulo 4: Resultados Preliminares

São apresentados resultados preliminares com experimentos usados na sumarização e comparados com ferramentas da literatura. Para os experimentos da *clusterização* foram usados um *clusterizador* da literatura e o *clusterizador* proposto no trabalho, utilizando a sumarização como fase de pré-processamento.

Capítulo 5: Considerações Finais

Neste capítulo, é apresentada a necessidade da implementação do módulo de Aprendizado Autônomo no modelo *Cassiopeia*; uso da categorização

no pós-processamento para melhorar a descoberta de conhecimento em textos e a questão do *corpus* em outras línguas para novas simulações.

2. Estado da Arte dos C*lusterizadores* de Texto

2.1. Clusterizadores

Clusterização corresponde ao processo de agrupar os elementos (objetos) de uma base de dados (conjunto) de tal forma que os grupos formados, ou *clusters*, representem uma configuração em que cada elemento possua uma maior similaridade com qualquer elemento do mesmo *cluster* do que com elementos de outros *clusters* (BERKHIN, 2002).

Existem basicamente duas classes de problemas de *clusterização*; o caso mais estudado é o número de *clusters* previamente definido conhecido como o problema de *K* - *clusterização* ou simplesmente problema de *clusterização.* No segundo caso onde este número *K* não é conhecido previamente, o problema é denotado por Problema de *Clusterização* Automática (BERKHIN, 2002).

2.2. Clusterizadores de Textos

A denominação do processo de *clusterização* na literatura é ampla: processo de aglomeração, processo de agrupamento, ou simplesmente aglomeração ou agrupamento (WIVES, 2004). Para Wives (1999), agrupamentos identificam associações ou correlações entre objetos que são colocados em grupos relativamente homogêneos, denominados conglomerados.

O agrupamento em documentos é usado nas três grandes áreas de mineração de texto: EI, RI e KDT. Na área de KDT, é conhecida como a técnica de descoberta por *clusterização,* que é uma das abordagens escolhidas para este trabalho. Na literatura, a análise de *clusters* pode ser denominada, segundo Wives (2004), como análise de classificação, taxonomia numérica,

quantização de vetores, aprendizado por observação e análise de dados esparsos; isto se deve a sua aplicação em diversas áreas, tais como Biologia, Psiquiatria, Psicologia, Arquitetura, Geologia, Geografia, Marketing, Inteligência Competitiva, entre outras.

Para o uso de agrupamentos textuais, a literatura dispõe de dois grandes grupos de algoritmos que se destacam na área de agrupamento textual. Um focando o agrupamento de palavras similares (relacionamento entre palavras) e outro focando o agrupamento de documentos (relacionamento entre documentos). Neste trabalho, o objetivo é usar as duas abordagens algorítmicas, que serão discutidas na seção 3.3.4 (algoritmos de agrupamento de documentos) e na seção 3.3.5 (relacionamento entre palavras).

Existem alguns trabalhos na literatura que utilizam o processo de *clusterização* em bases textuais, boa parte deles está na área de RI e outros na área de mineração de dados: (SUGENO *et al*, 1993) apresenta um algoritmo *fuzzy* para *clusterização,* denominado de *C-means*, baseado em análises estatísticas; (CRESTANI *et al*, 1997) sugere, por intuição, a extração das características que ocorrem em mais de um elemento da classe usando *clusterização*; (RIZZI *et al*, 2000) utiliza técnicas de seleção de palavras representativas em conglomerados (técnicas de identificação de centróides), a fim de identificar descritores de classes para o seu método de classificação de textos através de Redes Neurais. (GREHS, 2002) avalia a influência da variação morfológica das palavras no processo de agrupamento, aplicando métodos de *stemming* em documentos; (LEUNG *et al*, 2003) propõe um algoritmo baseado em mineração de dados para calcular a similaridade estrutural entre documentos XML por agrupamento.(LOPES, 2004) propõe

uso de mineração de dados textuais utilizando técnicas de *clustering* para a língua portuguesa.

Os trabalhos correlatos na área de KDT utilizam a *clusterização* e outros métodos para descoberta de conhecimento de texto. Na Tabela 2, são relacionados alguns trabalhos que usam *clusterização* e podem ser colocados como trabalhos correlatos. Tais trabalhos são descritos a seguir:

Tabela2.Trabalhos Correlatos na área de Descoberta de Conhecimento usando *Clusterização*.

Autor(es)-Ano	Técnicas	Domínio
WIVES-2004	*Clusterização* + Conceitos	Taxa de Mortalidade
LOPES-2004	Biblioteca de *Clusters* + Visualização *(*dendrograma)	Vário *Corpus* de domínios variados
PASARIN-2005	Termos de Seleção + *Clusterização*	Textos de RSS
Loh *et al* -2006	Análise Seqüências Temporais + Conceitos	Prontuários Médicos de Psiquiatria
Strube-2008	*Clusterização*(CLUTO) + Conceitos	Textos de Esporte

- Wives (2004) usa a *clusterização* agrupa os textos usando a lógica *Fuzzy.* Como forma de representação de conhecimento textual, usa-se a representação de conceitos. Segundo Wives (2004), conceito é representar o conteúdo dos documentos, como descritores no processo de identificação de similaridade entre os textos e, consequentemente, para que estes sejam agrupados em conglomerados de assuntos similares, Wives (2004) consegue obter estes descritores através de vocabulário controlado ou padronizado. Isto faz com que seu trabalho tenha uma forte manipulação na fase de pré-processamento e uma dependência muito forte do domínio ao qual está trabalhando.
- Lopes (2004) usa uma biblioteca de algoritmo de *clusters* denominada *C-Clustering Library,* que tem os algoritmos *Clustering* Hierárquico, *Clustering K-means* e *Self-Organizing Maps*, com uso da matriz de similaridade onde são realizados os cálculos com TF (*Term Frequency*), IDF (*Inverse Document Frequency*) e TF-IDF (*Term Frequency * Inverse Document Frequency*) e um processo de pós-processamento que usa visualização de dendrograma baseado em matriz de cores. Existe forte dependência da fase de pré- processamento onde usa *stopwords*, substituição de sinônimos, algoritmos *RSLP* e *Porter.* Nesta fase, Lopes (2004) garante o uso em outras línguas, mas devem ser mudados todos os arquivos. No processamento é realizado cálculo da matriz de similaridade, são feitos vários cálculos e no pós-processamento existe a visualização bastante complexa para identificar padrões e, consequentemente, obter algum conhecimento. O autor

relata ainda, dependência para uma quantidade de termos a ser descrita para cada *corpus* para a melhora da taxa de acerto.

- Passarin (2005) propõe o uso de *clusterização* com termos de seleção em texto coletados do RSS. Os resultados são dependentes dos termos utilizados e não tem uma abrangência. Ainda o autor, relata uma coesão ruim de alguns agrupamentos formados. Outro problema relatado por Passarin (2005) é a falta no processamento da *reclusterização* que provavelmente seja outro fator que influencia a falta de coesão.
- Loh *et al* (2006) aborda a questão da análise de sequência de uma série temporal ou uma ordem cronológica com o uso de conceitos em textos. Este trabalho foi usado dentro de um domínio específico de textos extraídos de prontuários médicos de uma clínica de psiquiatria. A ideia avalia a probabilidade condicional entre características dos textos ao longo do tempo, considerando temporalidade que existem em várias sequências e reconhecimento de padrões existentes, bem como seus conceitos. O grande problema está na limitação do domínio, haja vista que este é um assunto bastante específico e o autor deseja aplicar em outro domínio como, por exemplo, ocorrências policiais. Ele possui limitações citadas pelo próprio autor em relação aos operadores temporais.
- Strube (2008) apresenta em seu trabalho uma ferramenta denominada CLUTO (*Clustering Toolkit*), que organiza os agrupamentos e define os conceitos. A autora escolhe um *corpus* e determina um domínio específico (esporte), pois afirma ser mais constante e reduzido. No pré-processamento usa a ferramenta FORMA (etiquetagem morfológica e

lematização) e duas medidas TFIDF e *C-Value* para atribuir valor aos termos e selecioná-los em limiar de similaridade. Alguns problemas descritos por Strube (2008): ferramenta CLUTO tem que definir o número de *clusters* para análise dos conceitos, isto ocorre manualmente e um domínio específico, o de esporte. Houve vários problemas em relação a *clusters* gerados devido a medidas usadas para similaridade. A autora afirma problemas de semântica em alguns *cluters* gerados.

2.3. Considerações Finais

Nesta seção foi apresentado o conceito de *clusterização*, *clusterização de texto* e os trabalhos correlatos desta área. Na seção 3 será apresentado modelo *Cassiopeia*.

3. Modelo Cassiopeia

3.1. Apresentação do Modelo Cassiopeia

O modelo *Cassiopeia,* ilustrado na Figura 1, possui três módulos: Sumarização, *Clusterização* e Aprendizado Autônomo. A descrição detalhada de cada módulo será feita nas seções 3.2, 3.3 e 3.4, respectivamente.

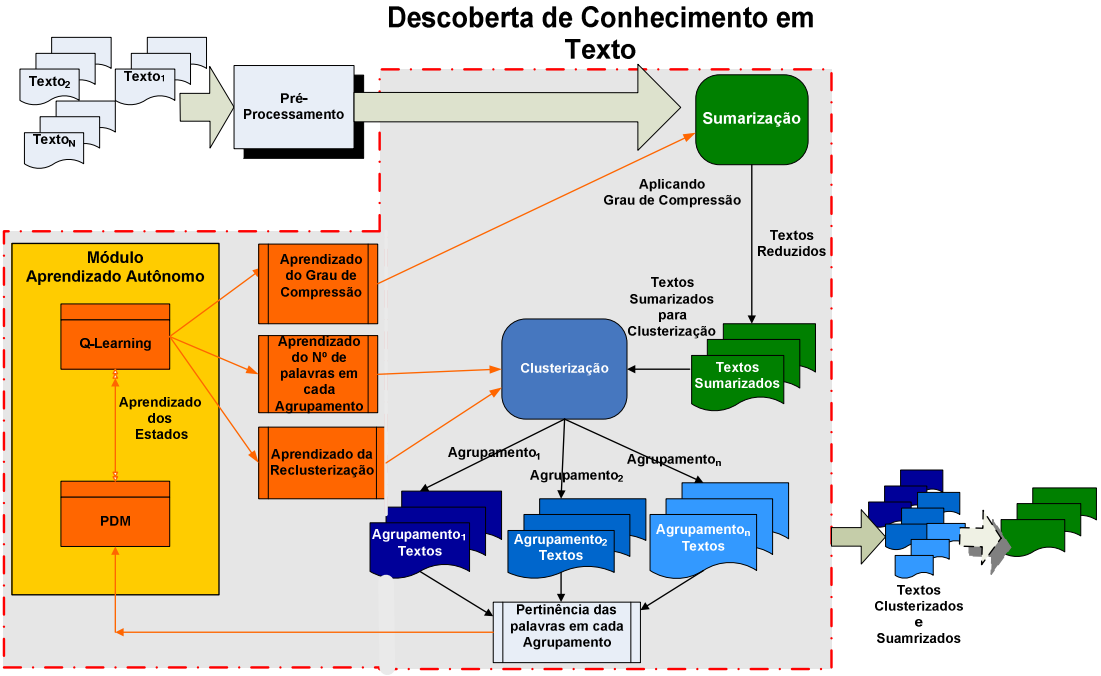

Figura 1.Proposta do Modelo *Cassiopeia.*

O modelo *Cassiopeia* começa com entradas de textos para descoberta de conhecimento. Estes textos passam pela fase de pré-processamento, onde são preparados para serem processados computacionalmente, este processo ocorre com o uso da técnica *case folding,* que coloca todas as letras em minúsculas e retira todas as figuras, tabelas e marcações existentes. Os textos passam a ficar em formato texto prontos para serem processados. Ainda na fase de pré-

processamento, é utilizada a sumarização, que tem como finalidade no modelo *Cassiopeia,* diminuir o número de palavras para a *clusterização,* que ocorre na fase de processamento. Desta forma, viabiliza a questão da alta dimensionalidade e possibilita a permanência das *stopwords* e a não utilização da matriz de similaridade*s* na fase de processamento, estes dois itens serão detalhados na seção 3.2.1 e 3.3.2. Para ocorrer a sumarização no modelo *Cassiopeia,* é definido o grau de compressão a ser usado em cada texto de forma não interativa. Para isto, é usado o algoritmo *Perfil,* que também será abordado na seção 3.2. A compressão é um parâmetro a ser aprendido pelo módulo de Aprendizado Autônomo, conforme a Figura 1, que ao longo do tempo, obterá valores cada vez melhor para compressão dos textos. A sumarização também é iterativa, pois este parâmetro é alterado mediante os resultados da *clusterização*.

O modelo *Cassiopeia* não prevê a fase de análise de dados isoladamente. Ela ocorre no módulo de sumarização, mas é não interativa, pois usa o algoritmo *Perfil* que viabiliza esta possibilidade.

Depois dos textos pré-processados, começa a fase de processamento, onde eles são *clusterizados*, ou seja, colocados em agrupamentos por critério de similaridade, que será explicado em detalhes na seção 3.3. Cada um dos agrupamentos criados possui um vetor de palavras, ordenado de forma decrescente, denominado centróides dos agrupamentos, ao qual possui palavras pertinentes aos textos do agrupamento. Comparando a entrada de novos textos, podem surgir agrupamentos, subagrupamentos ou até mesmo a fusão destes (LOH, 2001).

Os vetores de palavras, por questão de dimensionalidade, adotam segundo Wives (1999), um limiar de similaridade (*threshold*). O porquê deste limiar será explicado em detalhes na seção 3.3.5, mas, mediante o processamento, poderá sofrer variações até chegar ao seu valor de estabilização, ou seja, o grau de pertinência de cada palavra em cada agrupamento, conforme a Figura1. Este é outro parâmetro também a ser aprendido no módulo de aprendizado autônomo que, ao longo do tempo, obterá índices de pertinências dos vetores de palavras cada vez melhores.

Estes agrupamentos estão organizados de uma forma hierárquica (*top-down*). Todo este processo é iterativo e não interativo. A *reclusterização* ocorre até o momento que os vetores de palavras de cada agrupamento estejam estáveis. Este parâmetro, é outro valor a ser aprendido pelo módulo de aprendizado autônomo que, ao longo do tempo, definirá o melhor momento da parada da *reclusterização*.

Terminado a fase de processamento, começa a fase de pós-processamento onde cada um dos agrupamentos ou subagrupamentos de textos conterá por similaridade um conjunto de textos sumarizados.

3.2. Sumarização

3.2.1. Objetivos

Na Sumarização, o processo de diminuição do volume de palavras não relevante é bastante considerável, justamente pela definição do grau de compressão do sumário (percentuais de sentenças a serem extraídas do texto original). Este grau de compressão normalmente nos sumarizadores da

literatura é definido pelo usuário dependendo do nível de conhecimento que este tenha sobre o assunto abordado e/ou seu grau de interesse no texto.

A definição do grau de compressão pelo usuário seria um fator negativo, que inviabilizaria a utilização do módulo de aprendizado autônomo no modelo *Cassiopeia*, no qual a proposta é a não interatividade em qualquer uma das suas fases. Sendo assim, este trabalho desenvolveu uma proposta de sumarização, em que o grau de compressão é definido automaticamente. Este parâmetro de compressão será essencial para a entrada no módulo de aprendizado autônomo, cujo objetivo é, ao longo do processo, aprender o grau de compressão a ser utilizado nos textos.

A redução de palavras não relevantes para o processo de sumarização na literatura é descrito por (HEARST, 1993); (HEARST, 1997), (HEARST, 1998); (MITAL *et al*, 1999); (LARROCA *et al*, 2000) e (PARDO *et al*, 2001), como eliminação das *stopwords,* classes fechadas de palavras que não carregam significados, tais como artigos, pronomes, interjeições e preposições.

Outro objetivo para a utilização de sumarização é o nível de informação que esta compressão adiciona para o modelo *Cassiopeia*. Esta ideia foi inspirada nos resultados obtidos no trabalho de (PARDO, 2002), no qual foi proposto o *GistSumm*, que simula o comportamento humano na sumarização e o resultado. Seu sumário é a relevância do documento para o usuário.

3.2.2. Conceito de Sumarização Automática

A sumarização de textos é um processo que busca criar uma versão mais curta do texto original (MANI E MAYBURY, 1999). Esta necessidade de

simplificar e resumir ocorre devido ao aumento do volume de informações disponíveis nos meios de comunicação e ao pouco tempo para a leitura de textos de diversas naturezas. Como consequência deste processo, ocorre a incapacidade dos leitores em absorver conteúdos dos textos originais. Sendo assim, o sumário é um resumo que tem o objetivo de captar a ideia principal do autor e passá-la em poucas linhas para o leitor.

A Sumarização Automática - SA vem sendo formalizada desde 1950, sendo que o marco inicial dessas pesquisas foi o método de (LUHN. 1958), o da palavra-chave. A partir daí surgem outros trabalhos na área, tais como:

A importância da primeira e da última frase do texto original (BAXENDALE, 1958); Escolha computacional das frases com o maior potencial de transmitir significância do texto original (EDMUNDSON, 1969); relevância da restrição do domínio (POLLOCK e ZAMORA, 1975); classificação dos sumários em indicativos, informativos e de críticas (HUTCHINS, 1987); hierarquização dos papéis semânticos em cada frase (PAICE e JONES, 1993); o uso da abordagem híbrida (MAYBURY, 1993); as relações retóricas de cada frase do texto (MARCU, 1997); uso do conhecimento simbólico e de técnicas estatísticas para sumarização (HOVY e LIN, 1997). Por fim, a taxonomia criada por (HUTCHINS, 1987) é um fator determinante para estabelecer sua aplicabilidade e uma avaliação consistente deste processo (SPARCK, 1999). Como constatado nos trabalhos que delimitam a SA, verifica-se uma taxonomia metodológica em que existe um processo de abordagem superficial, também descrito como empírico ou estatístico, e outra abordagem denominada de profunda ou fundamental.

Os SAs atuais têm dificuldade em gerar sumários que mantenham a fidelidade do pensamento dos autores dos textos, bem como adaptar o sumário para o interesse dos leitores, independente de usar a abordagem profunda ou superficial.

Segundo Pardo *et al* (2001), os trabalhos recentes estão adotando metodologias híbridas, ou seja, o uso da abordagem superficial e profunda, variando os métodos de cada uma. Os trabalhos de sumarização automática enfatizam nos seus algoritmos a exclusão das *stopwords* para que possam realizar a fase da redução, como por exemplo, o algoritmo *TextTiling,* usado por (HEARST, 1993), (HEARST, 1997), (MITAL *et al*, 1999) e (LARROCA *et al*, 2000).

3.2.3. Aquisição de um Perfil Pragmático

Hovy propõe na abordagem profunda o uso de perfis pragmáticos, estabelecendo algumas métricas denominadas pelo autor como características de estilo. Ele estabelece uma relação temporal na preparação do texto original e classifica como escasso, pouco, suficiente ou ilimitado, de acordo com a Tabela 3. O autor também aborda o tipo de escrita textual, baseando-se no uso de algumas regras para classificar textos quanto a sua formalidade, tais como: coloquial, normal ou formal (HOVY, 1988).

Os textos com características formais tendem a ter frases mais longas, ou seja, possuem um número maior de palavras. Por sua vez, textos mais coloquiais tendem a ter um número menor de palavras. Há garantia segundo Hovy (1988), que o tamanho do sumário seria coerente com o nível de formalidade do texto, tendo como consequência a determinação do tamanho de compressão do sumário. Outra característica extremamente relevante é a

temporalidade na confecção de textos. Esta métrica é importante, pois é outro fator determinante para compor o sumário. Seja devido à limitação de espaço ou devido ao nível de conhecimento.

Tabela 3.Representa a interação das regras de estilo e implicações no conteúdo do sumário segundo (HOVY, 1988).

Tempo / Formali-dade	Escasso	Pouco	Suficiente	Ilimitado
Coloquial	Sumarização alta apenas tópico principal	Sumarização média	Sumarização média tópico principal, detalhes desconhecidos	Sumarização baixa tópico principal, detalhes relevantes
Normal	Sumarização média tópico principal, poucos detalhes	Sumarização média tópico principal	Sumarização média tópico principal, detalhes importantes	Sumarização baixa tópico principal, detalhes relevantes
Formal	Sumarização média tópico principal.	Sumarização média tópico principal	Sumarização baixa tópico principal, detalhes importantes	Sumarização baixa tópico principal e correlatos, detalhes relevantes

O uso da compressão em resumos para sentenças que compartilham de certo grau de sobreposição de informações, também foi usada por Hovy (HOVY *et al*, 2005). Nesta proposta, a ideia da compressão é automática. Ela é usada para garantir que o tamanho do sumário seja coerente com o grau de formalidade do texto, tendo, como consequência direta, a determinação automática do tamanho de compressão do sumário.

3.2.4. Proposta de um método de sumarização automática

O sumarizador proposto neste trabalho usa em sua forma de composição a extração e a transposição das sentenças, respeitando a sua posição no texto original, (abordagem superficial), e também adota as regras de Hovy para classificar o texto original (abordagem profunda), baseando-se no perfil pragmático do usuário, como ilustra a Figura 2. O sumarizador usa o texto-fonte obtido de *Corpus* de Textos. Os textos são classificados pelo algoritmo *Perfil* proposto por Guelpeli *et al* (2007), quanto à sua formalidade e temporalidade (fase de análise), baseada nas regras de estilo de Hovy. Com isso, o algoritmo *Perfil* determina o grau de compressão automático do texto que será usado para obter o sumário, refletindo assim, o perfil pragmático do usuário (fase de redução) de forma não interativa. O sumário realiza a extração e a transposição das sentenças, compondo o sumário com as frases de maior frequência de palavras determinada pelo algoritmo *Perfil* (fase síntese).

Outro fato a destacar do trabalho é a manutenção das *stopwords* usadas na fase de análise. A remoção das palavras do texto serve para diminuir o volume de processamento. Segundo estudos de Riloff *et al*(2005), estas

palavras são relevantes, porém, não existem estudos comparativos que comprovem ou neguem tal eficiência.

O estudo realizado neste trabalho usa a metodologia que remove as *stopwords* (algoritmo *TextTiling*) e o algoritmo *Perfil* (GUELPELI *et al*, 2007), que as mantém.

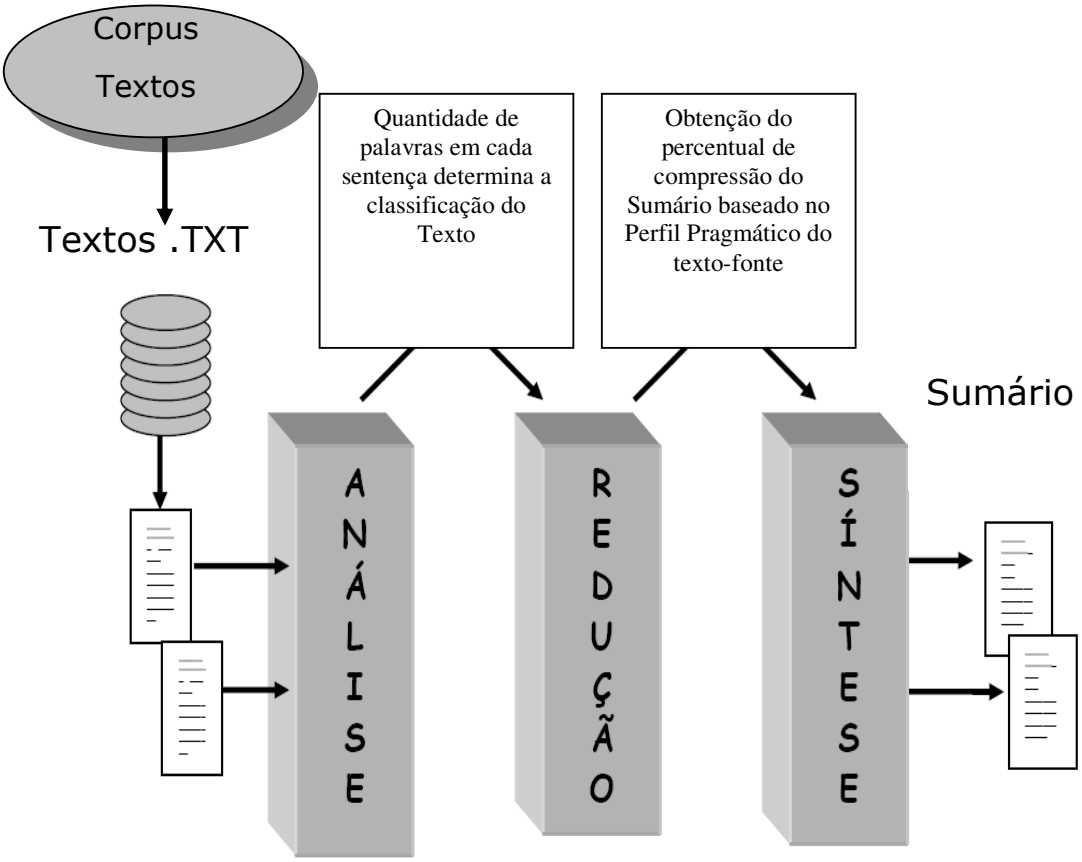

Figura 2. Estrutura do Sumarizador Automático usando o algoritmo Perfil.

3.2.5. Algoritmo Perfil

A construção do SA neste trabalho foi baseada no algoritmo *Perfil* proposto por Guelpeli *et al* (2007), que possui 11 passos. Seu objetivo é definir automaticamente o índice de compressão de forma não interativa, através da maior frequência de palavras encontradas no texto.

Três fatos colaboram para diferenciar o algoritmo *Perfil* dos sumarizadores propostos na literatura:

Primeiro a definição do índice de compressão automaticamente pelo *Perfil* em comparação com o *GistSumm* (PARDO, 2002) e a ferramenta *Autoresumo do MS-Word* (Microsoft®Word, 2002), que são definidos pelo usuário.

Segundo é a questão manutenção das *stopwords*. Os resultados apresentados no capítulo IV na seção 4.1 mostram uma significativa vantagem do algoritmo *Perfil* para o *GistSumm* (Pardo,2002) e a ferramenta A*utoresumo do MS-Word* (Microsoft®Word, 2002), que elimina este grupos de palavras. Estes resultados obtidos são indícios significativos da questão apresentada por Riloff (2005), na qual as *stopwords* podem ser relevantes.

Por último, os sumarizadores de diversas línguas são dependentes da linguagem. Usam técnicas específicas que reduzem as palavras aos seus radicais e *stopwords,* que são altamente dependentes. O algoritmo *Perfil* não usa estas técnicas, sendo assim, não depende da linguagem.

3.2.6. Considerações Finais

Nesta seção, foi apresentado o objetivo da sumarização para o modelo *Cassiopeia*, conceitos de sumarização e os fundamentos teóricos para

aquisição de um perfil pragmático. Foi proposta uma métrica que usa todas as palavras dentro da sentença para classificar os textos dentro do perfil pragmático, baseado nas regras de (HOVY, 1988) e (HOVY, 2005) e também o algoritmo *Perfil* (GUELPELI *et al*, 2007). A metodologia proposta neste trabalho mantém todas as palavras dentro da sentença, preservando as *stopwords*, ao contrário dos trabalhos desta área, que removem as *stopwords* para diminuir o volume de processamento. Outro contraponto desta metodologia em relação aos trabalhos da literatura é justamente no momento de definir o percentual de compressão. Nos sumarizadores da literatura é necessária uma interação para definir o grau de compressão, já no *Perfil,* este percentual é obtido automaticamente e, por fim, a questão da não dependência da linguagem.

3.3. Clusterização

3.3.1. Objetivos

A *clusterização* de textos é utilizada quando não se conhece os elementos do domínio disponível e, assim, procura separar, automaticamente, elementos em agrupamentos, por algum critério de afinidade ou similaridade (LOH, 2001). A *clusterização* auxilia o processo de descoberta de conhecimento em texto, facilitando a identificação de padrões nos agrupamentos (RIZZI *et al*, 2000). Como os agrupamentos não são previamente definidos, o processo é não interativo.

Para o modelo *Cassiopeia,* este tipo de descoberta é extremamente relevante, que ocorre a identificação dos agrupamentos “interessantes”,

potenciais para descoberta de algum conhecimento útil e de forma não interativa.

A identificação dos agrupamentos através de suas características denominadas análise de agrupamento (*cluster analysis*) é importante para o modelo *Cassiopeia*, já que os textos são agrupados por uma avaliação de similaridade entre eles, discutidas na Seção 3.3.5. A *reclusterização* acontece entre agrupamentos para encontrar conhecimentos mais específicos. A quantidade de *reclusterização* é um parâmetro a ser aprendido e é um argumento de entrada para o módulo de aprendizado autônomo. Determina o término do algoritmo e a parada do processo de forma não interativa.

Por fim a adoção da *clusterização* no modelo *Cassiopeia* é possibilitar a descoberta de conhecimento em texto de *corpus* distintos e/ou antagônicos, já que a estrutura proposta é sensível a esta mudança de contexto.

3.3.2. Conceito de *Clusterização*

A palavra *clusterização* é um neologismo do termo *clustering,* que difere da classificação, pois a primeira, visa criar os agrupamentos através da organização dos elementos, enquanto a segunda, procura alocar elementos em classes já pré-definidas. A *clusterização* não necessariamente precisa definir os agrupamentos, basta separar os elementos nos grupos (LOH, 1999); (LOH, 2001).

O processo de *clusterização* pode ser definido como regiões contínuas de um espaço que tem grande número de pontos e por regiões com baixa quantidade de pontos. Esses agrupamentos têm as seguintes propriedades: densidade, variância, dimensão, forma e separação. É a partir destas

propriedades que surgem diferentes tipos de agrupamentos, que podem ser hiperesféricos, alongados, curvilíneos ou podem possuir estruturas mais diferenciadas (ALDENDERFER E BLASHFIELD, 1984); (FASULO, 1999); (BERKHIN, 2002).

Os agrupamentos têm uma taxonomia quanto à sua configuração: hierárquicos aglomerativos (*hierarchical aglomerative*), hierárquicos divisivos (*hierarchical divisive*), de particionamento iterativo (*iterative partitioning*), de busca em profundidade (*densitysearch*), fator-analíticos (*factor analytic*), de amontoamento (*clumping*) e grafoteoréticos (*graph-theoretic*). Esses algoritmos, quando aplicados a um conjunto de dados, geram resultados diferentes (ALDENDERFER E BLASHFIELD, 1984), (EVERITT, 2001) e (BERKHIN, 2002).

Para este trabalho foi escolhida a configuração hierárquica. Nos métodos hierárquicos, os dados são particionados sucessivamente, produzindo uma representação hierárquica dos agrupamentos. Este tipo de representação facilita a visualização sobre os agrupamentos a cada estágio, bem como a percepção do grau de similaridade entre eles. É um método que não requer definições de número de agrupamentos. A principal vantagem deste método está na facilidade em lidar com qualquer medida de similaridade utilizada, e a sua consequente aplicabilidade a qualquer tipo de atributo (numérico ou categórico) (BERKHIN, 2002). As principais desvantagens são: o critério de parada e o não refinamento de seus resultados conforme a hierarquia vai sendo construída. Quanto ao critério de parada, esse pode ser definido quando chegar a um determinado número de agrupamentos ou quando algum tipo de condição de parada ocorrer. Tal critério exige uma matriz de distância entre

os agrupamentos, denominada matriz de similaridade (JAIN E DUBES, 1988). A Tabela 4 mostra a matriz de similaridade que caracterizaria outro problema nos métodos hierárquicos porque ela cresce exponencialmente mediante sua base de dados (VIANNA, 2004).

Tabela 4.Matriz de Similaridade.

	Texto1	Texto2	Texto3	Texto4	Texto5
Texto1	1.0	0.3	0.2	0.7	0.6
Texto2	0.3	1.0	0.5	0.3	0.9
Texto3	0.7	0.5	1.0	0.3	0.4
Texto4	0.6	0.3	0.3	1.0	0.8
Texto5	0.2	0.9	0.4	0.8	1.0

Para o cálculo da distância na matriz de similaridade ilustrada na Tabela 4, podem ser usados vários métodos (ARORA E BANGALORE, 2005). Os mais importantes são: Ligação Simples (distância entre dois agrupamentos mais similares); Ligação Completa (distância entre dois agrupamentos menos similares); Centróides (distância entre dois agrupamentos é obtida pelos seus centróides); Média das Ligações (média da distância entre os elementos de cada agrupamento); Média dos grupos das ligações (a distância de dois agrupamentos é obtida pela média da união dos dois agrupamentos relacionados) e *Ward* (encontra partições que minimizem a perda associada a cada agrupamento). Para este trabalho não foi adotada a matriz de similaridade e, com isto, não houve a necessidade do cálculo de distância. Este procedimento será mais bem detalhado nas seções 3.3.4 e 3.3.5.

Nos métodos hierárquicos existem duas abordagens: aglomerativo (*Botton-up*) e divisivo (*Top-down*) (ALDENDERFER E BLASHFIELD, 1984), (FASULO, 1999) e (BERKHIN, 2002). Na abordagem hierárquica

aglomerativa ilustrada na Figura 3, os dados são inicialmente distribuídos de maneira que cada exemplo represente um agrupamento, e então esses são recursivamente agrupados, considerando alguma medida de similaridade. Sendo assim, no início, os agrupamentos são em números reduzidos com alto grau de similaridade entre seus elementos, mas com o decorrer do processo estes grupos vão aumentando e seus elementos tornando-se dissimilares (SILVA *et al*, 2005).

Neste trabalho, a abordagem hierárquica adotada foi a aglomerativa, pois computacionalmente é mais viável, devido ao custo computacional e sua eficiência, pois não necessita da definição de um número inicial de agrupamentos.

***1* - ALGORITMO AGLOMERATIVO:**

1. Procure pelo par de agrupamentos com a maior semelhança.

2. Crie um novo agrupamento que reúna o par selecionado no passo 1.

3. Decremente em 1 o número de agrupamento restante.

4. Volte ao passo 1 até que reste apenas um cluster.

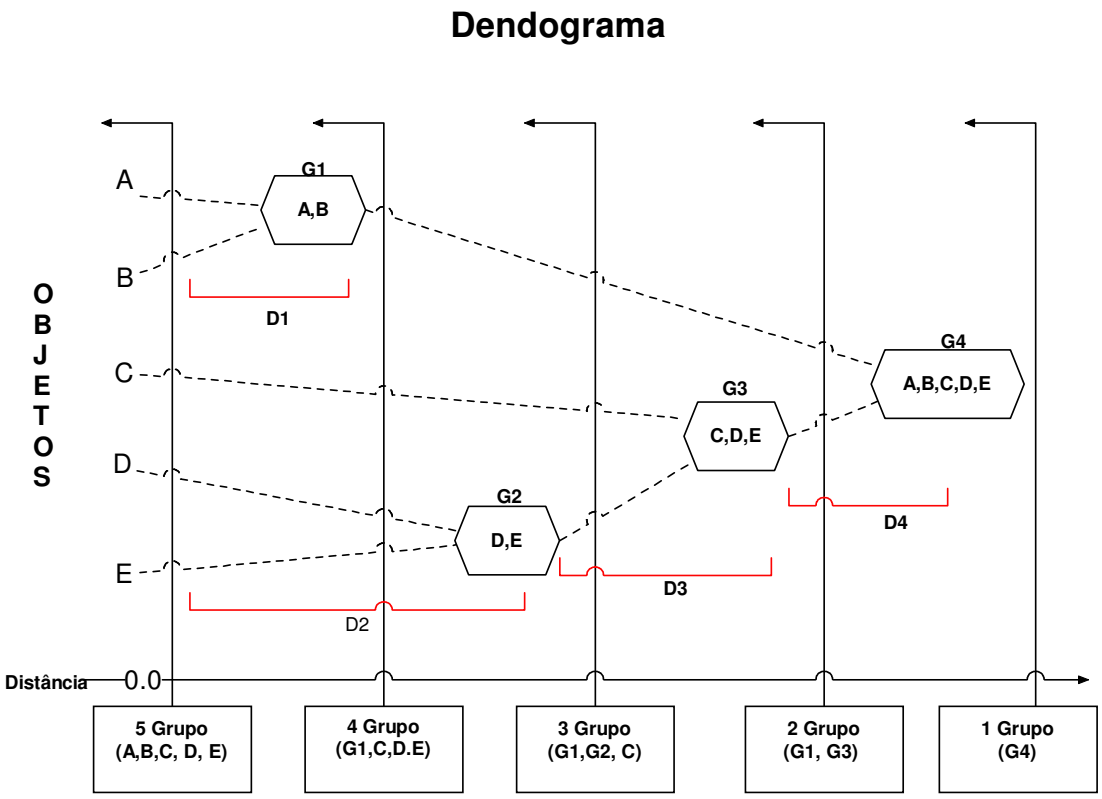

Figura 3. Dendograma do Método Hierárquico Aglomerativo.

No Algoritmo 1, são descritos os passos realizados nessa abordagem. Desta forma pode-se interpretar a Figura 3 como inicialmente contendo cinco agrupamentos [A, B, C, D, E]. Decorrido os passos, forma-se um agrupamento denominado G1, onde se encontram os agrupamentos [A, B], e a similaridade do agrupamento G1 é medida pela distância D1. O agrupamento G2 é formado pelos agrupamentos [D, E], sendo que a medida de similaridade de G2 é igual D2. No passo seguinte, é criado o agrupamento G3 constituído pelo agrupamento [C] e pelo agrupamento G2 e a distância de similaridade de

G2 para G3 é a distancia D3. O próximo passo é a composição do agrupamento G4, que é formado pelos agrupamentos G1 e G3 e a distância de similaridade é a D4.

Já o método divisivo é o menos comum entre os métodos hierárquicos, pois é ineficiente e tem um alto custo computacional (ALDENDERFER E BLASHFIELD, 1984), (EVERITT, 2001) e (BERKHIN, 2002). Na abordagem hierárquica divisiva (Figura 4), o processo é iniciado com apenas um agrupamento contendo todos os dados, e segue dividindo recursivamente segundo alguma métrica até que alcance algum critério de parada, frequentemente o número de *clusters* desejados (KARYPIS, 1999).

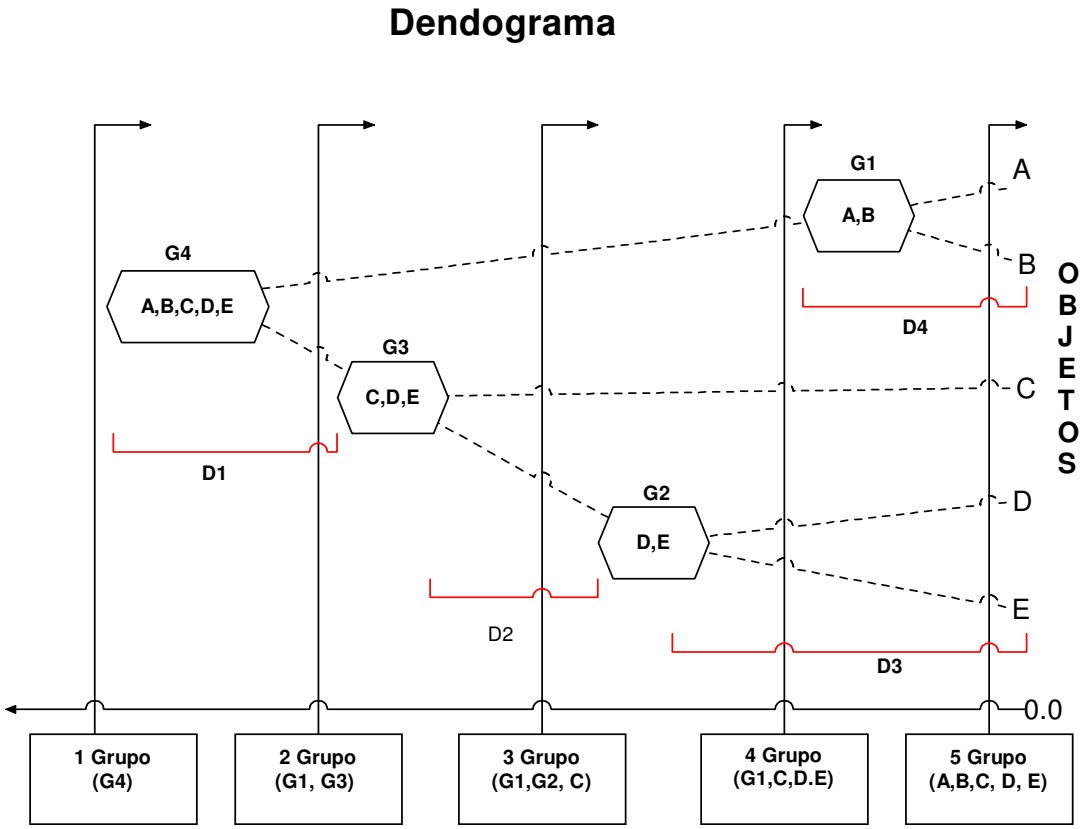

Figura 4. Dendograma do Método Hierárquico Divisivo.

Pode-se interpretar a Figura 4 como único agrupamento [G4], que é dividido em dois agrupamentos [G1 e G3] na qual a medida de similaridade é representada por D1. O próximo passo é dividir o agrupamento G3 em [C e G2] e a medida de similaridade entre esses agrupamentos é D2. Agora existem três agrupamentos [G1, G2 e C]. O agrupamento [G2] é dividido em [D e E] e a similaridade entre esses agrupamentos é a distância D3, sendo assim, restaram quatro agrupamentos [G1, C,D,E]. O próximo passo é dividir o agrupamento G1, em [A e B]; a similaridade entre eles é a distância D4. Agora ficaram os agrupamentos [A, B, C, D e E].

No Algoritmo 2, são descritos os passos realizados dessa abordagem.Um algoritmo hierárquico aglomerativo pode ser descrito basicamente da seguinte maneira:

2 - ALGORITMO DIVISIVO:

1. Um único agrupamento contendo todos os elementos.

2. Calcula-se a matriz de similaridade entre todos os pares do agrupamento.

3. Cria-se um novo agrupamento com divisão dos pares com menor grau de similaridade.

4. Volte ao passo 1 até que reste apenas um cluster.

3.3.3. Algoritmos de *clusterização* propostos na literatura

Os algoritmos mais importantes pertencentes ao método hierárquico aglomerativo, segundo Kowalski (1997), são os seguintes: *Cliques, Stars, Connected Components e Strings*.

Na visão algorítmica, os textos são analisados em agrupamentos para fins de descoberta de conhecimento e são necessários que os grupos

constituídos pelos textos tenham certa coesão entre si, este é o grande problema de processamento de linguagem natural, sua complexidade. Envolvem análises de uma série de questões como a coerência e a coesão dos textos, podendo assim, estar relacionado com questões de âmbito cultural, social, situacional, político, e/ou estarem diretamente relacionadas com o autor e o momento em que o texto foi escrito (FÁVERO, 2000).

Os agrupamentos com textos muito diferentes não seriam admissíveis mediante a falta de coesão entre seus textos.

O problema é que alguns dos algoritmos, como *Connected Components* e *Strings*, não são tão restritivos quanto se espera (WIVES, 1999), pois permitem que textos com pouco grau de similaridade se agrupem por possuírem uma forte relação com apenas um único texto do grupo e não com todos os textos que compõem o agrupamento.

Desta forma, para este trabalho, foram adotados os algoritmos *Cliques* e *Star*, que são da categoria de algoritmos grafoteoréticos, inovam por se basear em teoremas e axiomas conhecidos da teoria dos grafos e possuem alta capacidade dedutiva, tendo, portanto, maior fundamentação teórica (ALDENDERFER E BLASHFIELD, 1984). Devido à sua capacidade de construir agrupamentos mais coesos, ou seja, textos mais coerentes entre si (KOWALSKI, 1997), estes foram os escolhidos para a proposta deste trabalho.

- Algoritmo Stars

O algoritmo estrelas (*stars*) tem este nome justamente pelo fato de formarem conglomerados que possuem forma similar a uma estrela, ou seja, um

elemento central, diversos outros ligados a ele, formando as pontas de uma estrela (Figura 5). Nesse caso, o elemento central é quem possui relação com todos os elementos da estrela, interligados. Os elementos das pontas não necessitam ter relação direta entre si, sendo este, um dos maiores problemas desse algoritmo, pois eles podem não ser similares.

Para minimizar este problema da falta de similaridade entre os elementos que ficam nas pontas da estrela, um limiar de similaridade deve ser estabelecido. Logo, a solução para que elementos em pontos opostos da estrela não sejam muito dissimilares ou distantes, consiste na técnica de selecionar um grau de similaridade maior, já que quanto mais próximos do centro, mais similares entre si os elementos serão, dando maior coerência ao grupo. O Algoritmo 3 descreve os passos do Algoritmo *Stars*.

3 - ALGORITMO STAR:

1. *Selecionar 1 elemento e colocar todos os similares no mesmo cluster.*
2. *Elementos ainda não alocados/classificados são colocados como semente de cluster (repetir passo 1 para 1 elemento ainda não alocado).*

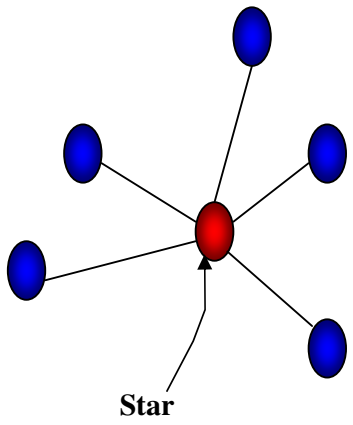

Figura 5.Grafo do Algoritmo Star.

- Algoritmo Cliques

O algoritmo cliques, cujo grafo formado é ilustrado na Figura 6, é similar ao algoritmo star. Entretanto, os elementos só são adicionados a um conglomerado, caso seu grau de similaridade seja maior do que o limiar para todos os elementos já presentes no conglomerado e não somente em relação ao elemento central. Neste caso, os conglomerados tendem a ser mais coesos e de maior qualidade, pois os elementos são mais semelhantes ou próximos.

O Algoritmo 4 descreve os passos do Algoritmo *Cliques*.

4 - Algoritmo Cliques:

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.*

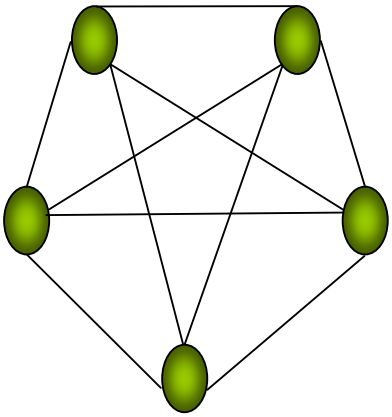

Figura 6.Grafo do Algoritmo *Cliques*.

3.3.4. Proposta de uma abordagem para *clusterização* de textos usando similaridade difusa

A ambiguidade é o maior desafio enfrentado pelos sistemas que tratam da linguagem natural. Identificar o verdadeiro significado de uma determinada palavra pode ser tão complicado, que às vezes só é possível com uma consulta ao usuário. No processo de escolha de uma alternativa de tratamento matemático mais adequado às questões formuladas em linguagem natural, existe uma grande vantagem em optar pelo uso da lógica difusa (*fuzzy*), pois a lógica convencional apresenta algumas dificuldades em representar conceitos abstratos. Na lógica convencional, ou booleana, usualmente empregada em computação, são definidos apenas dois valores possíveis - verdadeiros (1) ou falsos (0). Essa lógica não é ideal para uso em sistemas que tratam de linguagem natural, pois não conseguem abordar toda a representação do contexto linguístico com fidelidade. Baseiam-se apenas no certo ou errado, no sim ou não, ou seja, em apenas dois valores para representação de um mundo extremamente complexo.

A lógica nebulosa ou lógica difusa baseia-se na teoria de conjuntos nebulosos (*Fuzzy Sets*), e teve seus conceitos e princípios introduzidos por Lotfi Zadeh (ZADEH, 1965); (ZADEH, 1973). A lógica nebulosa é multivalorada, ou seja, há um conjunto de valores possíveis. Assim, a Lógica *Fuzzy* pode ser definida como a lógica que suporta os modos de raciocínio aproximados, em vez de exatos. O tratamento matemático da lógica nebulosa é mais adequado para informações imprecisas usualmente empregadas na comunicação humana, permitindo inferir uma resposta aproximada para uma questão baseada em um conhecimento que é inexato, incompleto ou não

totalmente confiável. O uso de conjuntos nebulosos, naturalmente inclinados a lidar com o conhecimento linguístico do domínio, pode produzir soluções mais interpretáveis (MITRA *et al*, 2003), o que permite criar sistemas utilizando variáveis linguísticas. É nessa natureza da informação que se encontra a nebulosidade (KLIR *et al*, 1988).

Uma função em Lógica *Fuzzy* deve ser generalizada de forma que ela possa assumir valores em um determinado intervalo, e o valor assumido indica o grau de pertinência do elemento no conjunto em questão. Dessa maneira, a função de pertinência μA de um conjunto nebuloso A é da forma: $\mu A: X \rightarrow [0,1]$. Em outras palavras, um conjunto nebuloso A é caracterizado por uma função de pertinência $\mu A(x)$, que associa a cada elemento do conjunto um número real no intervalo [0,1]. Desta forma, o valor de $\mu A(x)$ representa o grau de pertinência do elemento $x \in X$ no conjunto A (HELLMANN, 2001).

Existem trabalhos importantes que usam, por exemplo, lógica difusa na mineração de dados. Em (KWOK *et al*, 1998), essa técnica é utilizada em sistemas de tomada de decisão e marketing;(TSAUR *et al*, 2002) utiliza lógica *fuzzy* para analisar comportamentos do consumidor; (JIANAN e RANGASWAMY, 1999) mostra que a lógica *fuzzy* é o modelo matemático mais adequado para o tratamento dos dados em uma pesquisa na qual busca reproduzir o comportamento do consumidor na escolha de marcas em um supermercado virtual, em detrimento de métodos convencionais devidamente comparados, como modelos de lógica booleana determinísticos e probabilísticos.

O problema da ambiguidade em processamento de textos pode ser atacado com o uso da lógica difusa, pois a mesma se propõe tratar situações imprecisas, oferecendo melhores resultados através do cálculo da pertinência de um elemento a um conjunto. Utilizando essa técnica, é possível definir o quanto um termo é importante e relevante ou não para um agrupamento. Existem várias funções difusas que podem ser utilizadas para esta finalidade.

A função difusa usada neste trabalho é a de inclusão simples (*set theoretic inclusion*) relatada por Cross (1994). Esta função *fuzzy* avalia a presença de palavras em dois documentos sendo comparados. Se o termo aparece nos dois, soma-se o valor 1 ao contador; caso contrário, soma-se 0. Ao final, o grau de similaridade é um valor difuso entre 0 e 1, calculado pela média, isto é, o valor total do contador de palavras comuns dividido pelo número total de palavras nos dois documentos. Esse valor *fuzzy* representa o grau no qual um elemento está incluso no outro ou o grau de igualdade entre eles.

Existe um problema nessa 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 aparecer com frequências diferentes, não é levado em consideração.

Esse problema pode ser resolvido, em parte, por outra função (OLIVEIRA, 1996), que realiza a média por operadores *fuzzy*, semelhantes à anterior, porém, utilizando pesos para as palavras. Assim, o fato das palavras aparecerem com importâncias diferentes nos documentos é levado em consideração.

O valor de similaridade é calculado pela média entre os pesos médios das palavras 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 palavras nos dois documentos. Essas duas funções encontradas na literatura (CROSS 1994); (OLIVEIRA, 1996) foram usadas de forma isoladas, mas neste trabalho serão utilizadas em conjunto, para o cálculo de similaridade entre textos (*set theoretic inclusion*) e entre agrupamentos (média por operadores *fuzzy*)

Cada texto é representado por um conjunto de características que melhor o define, e a similaridade difusa é então utilizada para definir quão semelhantes são dois vetores representativos. A partir de um conjunto de características de um texto, composto neste trabalho pela atribuição da relevância das palavras em relação a esse texto, a abordagem *fuzzy* se baseia na ideia da similaridade do texto em relação ao agrupamento.

O uso da lógica difusa, neste trabalho, está centrado em agrupar os textos, não apenas em palavras de pertinência ou não, mas em palavras de graus variados de similaridade.

A abordagem *fuzzy* neste trabalho é utilizada, agrupando textos de acordo com uma medida de similaridade entre eles. Sendo assim, quanto mais próximo do centro estiver o texto, mais similar ele será, e quanto mais longe do centro, menos similaridade existe.

Os resultados encontrados são agrupamentos parciais, sendo atribuído um grau de relevância cada um, com relação ao texto analisado. Para verificar a semelhança entre um texto e um agrupamento, são comparadas todas as

palavras que compõem o conjunto de características do texto com palavras que compõem o conjunto de características do agrupamento.

Um termo é considerado semelhante quando ele é encontrado tanto no índice do agrupamento como no índice do texto. Os graus de igualdade das palavras são, então, usados para determinar o grau de similaridade entre o índice do texto e do agrupamento. Desta forma, o texto é alocado no agrupamento em que obtiver o maior grau de similaridade. A proposta deste trabalho para agrupar documentos textuais é denominada de *clusterizador* e é dividida em 3 fases, como é mostrado na Figura 7.

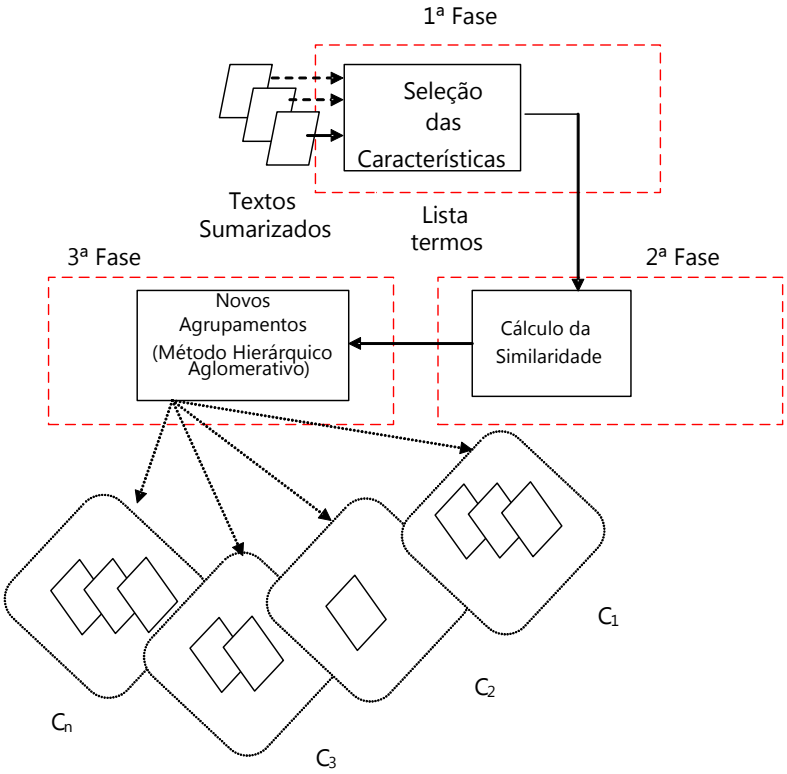

Figura 7.As fases do *Clusterizador* usando similaridade difusa.

3.3.5. Fases do Clusterizador

Primeira fase - (Seleção de características): são selecionadas características das palavras no texto utilizando a frequência relativa. Ela define a importância de um termo dado de acordo com a frequência em que o termo é encontrado no texto. Quanto mais vezes um termo aparecer em um texto, mais importante este termo é para sua definição. A frequência relativa é calculada pela equação (*1*) (SALTON, 1983). Esta fórmula normaliza o resultado da frequência absoluta das palavras, evitando que documentos pequenos sejam representados por vetores pequenos e documentos grandes, por vetores grandes.

Com a normalização, todos os documentos serão representados por vetores de mesmo tamanho.

Equação 1: $$F_{rel}X = \frac{F_{abs}X}{N}$$

Onde:

- $F_{real}X$ = frequência relativa de *X;*
- $F_{abs}X$ = frequência absoluta de *X,* ou seja, quantidade de vezes que *X* aparece no documento;
- *N* = número total de palavras do texto.

Um dos maiores problemas encontrados em KDT é lidar com espaços de dimensões muito altas pois, se for considerado um espaço-vetorial, cada palavra representa uma dimensão (existem tantas dimensões quanto palavras diferentes). Dessa forma, um dos problemas importantes tratados na primeira fase da abordagem é reduzir a dimensionalidade. Para isso, foi adotado também um valor mínimo de importância, um limiar (*threshold*) ou limiar de similaridade (WIVES,1999), no qual as palavras (características) com importância (frequência) inferior a este valor são, simplesmente, ignoradas. O

uso dessa técnica é importante devido à alta dimensionalidade do espaço de características, ou seja, à grande quantidade de palavras que compõem um documento, e que devem ser tratadas. É necessário então reduzir o espaço para se alcançar um melhor agrupamento.

Segunda fase - (Cálculo de Similaridade): esta fase identifica a similaridade entre os textos (características selecionadas na primeira fase). Para tal, foi utilizada medida de similaridade difusa (*fuzzy*) - a medida de inclusão simples (*set theoretic inclusion*), (CROSS, 1994), na qual avalia a presença de palavras nos dois textos comparados. Esse valor difuso representa o grau em que um elemento está incluso no outro ou o grau de igualdade entre eles.

Se a palavra aparece nos dois textos, 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 comuns divididos pelo número total de palavras nos dois textos (sem contar repetições).

O fato de uma palavra ser mais importante em um ou outro texto, por aparecer com frequências diferentes, não é levado em consideração. Esse problema pode ser resolvido, em parte, por outra função (OLIVEIRA, 1996), que realiza a média por operadores *fuzzy*, semelhantes à anterior, porém, utilizando pesos para as palavras. Assim, o fato das palavras aparecerem com importâncias diferentes nos textos é levado em consideração. Nesse caso, os pesos das palavras têm como base a frequência relativa. O valor de similaridade é calculado pela média entre os pesos médios das palavras comuns. Isto é, quando a palavra aparece nos dois documentos, soma-se a

média dos seus pesos em vez de somar o valor um (1). Ao final, a média é calculada sobre o total de palavras nos dois documentos.

Terceira fase - (Método Hierárquico Aglomerativo): utiliza o método Hierárquico Aglomerativo e através da análise dos dendogramas construídos define o número prévio de agrupamentos. Na segunda fase, foi calculada a similaridade entre os textos, sendo o algoritmo *Cliques*, capaz de construir agrupamentos mais coesos (GUELPELI *et al* 2008).

3.3.6. Considerações Finais

Nesta seção, foi discutido o conceito de agrupamentos de textos, métodos hierárquicos (aglomerativo e divisível) e algoritmos grafoteoréticos (*Cliques*). Foi proposta uma metodologia usando duas funções *fuzzy* (*set theoretic inclusion*) e média (por operadores difusos) para diminuir a questão de ambiguidade semântica.

3.4. Aprendizado Autônomo

3.4.1. Aprendizado Autônomo

O aprendizado autônomo é empregado em sistemas em que um processo dinâmico a ser controlado sofre modificações estruturais ao longo do tempo, ou quando a modelagem do processo é difícil, ou muito custosa. A aprendizagem autônoma tem a capacidade de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência, realizando assim, uma aprendizagem a partir das próprias experiências. Um modelo de interação com o processo dinâmico externo (ambiente) para o problema de aprendizado autônomo está ilustrado na Figura 8.

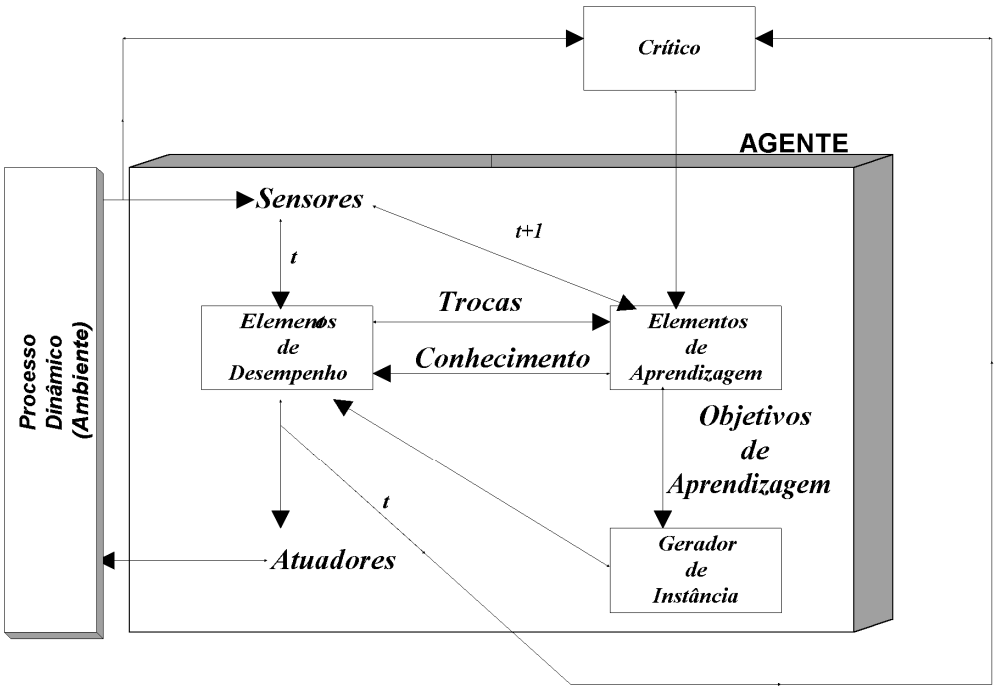

Figura 8.Modelo Dinâmico e seus elementos.

O modelo dinâmico contém os seguintes elementos:

O agente relaciona-se com o ambiente via percepção (Sensores) e ação. Em cada passo da interação o agente recebe indicações do estado atual do ambiente através do crítico e escolhe uma ação para gerar como saída (Atuadores).

O crítico comunica ao elemento de aprendizado quão bem ou mal o agente está operando de acordo com algum critério. Isto é necessário porque observações sensoriais em si, não são indicativas de desempenho.

Elemento de aprendizado é o processo pelo qual parâmetros do agente (armazenados no elemento de desempenho) são modificados. O elemento de aprendizado armazena instruções sobre como a modificação dos parâmetros

deve ser feita. Essas modificações de parâmetros têm como objetivo melhorar o desempenho do sistema, tal como medido pelo crítico.

Elemento de desempenho executa as ações do agente, passa parâmetros para o elemento de aprendizagem, recebe estes parâmetros dos sensores do gerador de instância e passa as ações para o agente através dos atuadores.

O Gerador de instâncias sugere ações alternativas (passadas pelo elemento de aprendizado) que podem ser tomadas pelo agente, e passadas para o elemento de desempenho.

Sensores possibilitam que o agente observe o comportamento do processo. Estas observações são transformadas em representações inteligíveis para o agente e podem corresponder a estados completos (seria ideal) ou parciais.

Atuadores são mecanismos pelos quais o agente atuará no mundo exterior. Os parâmetros recebidos pelos atuadores são passados pelo elemento de desempenho que modifica o estado do processo.

Uma função que especifica as ações combinadas do elemento de desempenho e do gerador de instância, em termos de observações ou tempo, é uma política de ações.

No processo dinâmico, o agente tem como objetivo mapear uma política de ações que lhe permita obter a maior quantidade possível de recompensa, segundo algum critério de comportamento ótimo. Desta forma, o projetista do sistema deve obter uma maximização ou minimização dos reforços obtidos pelo agente em longo prazo (GUELPELI *et al,* 2004). Dependendo da qualidade de informação (Reforço) fornecida pelo crítico,

podem-se obter alguns tipos de aprendizado, como por exemplo: Aprendizado Supervisionado, Aprendizado Não-Supervisionado e Aprendizado por Reforço. De acordo com as definições:

- Aprendizagem supervisionada, a informação de retorno (*feedback*) especifica a ação desejada (SUTTON E BARTO, 1998) e o objetivo da aprendizagem é aproximar-se desta ação desejada o máximo possível;
- Aprendizagem não-supervisionada, nenhum *feedback* explícito é fornecido e o objetivo é encontrar ações úteis e desejadas através de um processo de autorganização;
- Aprendizagem por reforço, o *feedback* especifica apenas o reforço associado à ação realizada, e o objetivo é maximizar uma função dos reforços obtidos ao longo do tempo.

No aprendizado autônomo associa-se um reforço às condições críticas do sistema. O *Q-Learning* é o algoritmo de aprendizado autônomo que será usado neste trabalho, é descrito na seção 4.5.1 capaz de propagar reforços ao longo do tempo, e executar este algoritmo várias vezes sobre o sistema. O modelo *Cassiopeia* tem características estocásticas, que forçará a adoção de um enfoque probabilístico para o estudo do algoritmo de aprendizado.

3.4.2. Processo Decisórios de Markov – PDM

Na propriedade de *Markov* o estado atual representa uma sumarização dos estados passados. Segundo Sutton e Barto (1998), pode-se predizer qual será o próximo estado e a próxima recompensa esperada, dado o estado e a ação

atual. A observabilidade perfeita destes estados, garante que o processo sempre seja de *Markov*.

Um Processo Decisório de *Markov* - PDM é definido por uma quádrupla {S, A, P, R} Sutton e Barto (1998). Sendo assim, um PDM consegue modelar o ambiente e mostrar a formalização do problema de aprendizado autônomo em que contém os seguintes elementos:

- Um conjunto finito de estados **S**
- Um conjunto finito de ações **A**
- Uma função de recompensa **R** esperada ao mover-se para **s'** é $\mathbf{R}^{a}_{ss'}$ **,** onde **Novo Estado – E{r_{t+1}| s_t = s, a_t=a, s_{t+1}=s'}**
- Uma função de probabilidade do novo estado **s'** é $\mathbf{P}^{a}_{ss'}$, onde **Probabilidade Pr{s_{t+1} =s'| s_t = s, a_t=a}**

A Figura 9 mostra a transição no processo decisório de *Markov* com seus elementos.

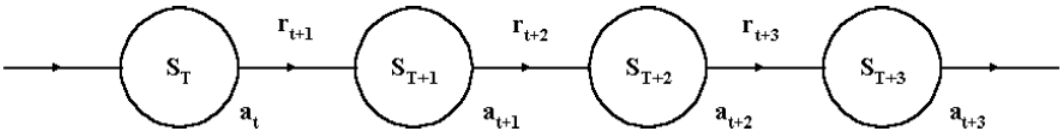

Figura 9.Modelo de Transição PDM.

3.4.2.1. Papel do Crítico no PDM

A função do crítico é produzir, para cada par (estado, ação) do PDM, um valor real (reforço $r(s_t, a_t)$), que é um indicativo do resultado instantâneo da

evolução do PDM e pode ser não-determinístico. O reforço produzido é instantâneo (pouco significativo, pois precisa de um horizonte mais a longo prazo para avaliar) e pouco informativo de caráter estocástico, ou seja, enfoques probabilísticos, estando normalmente associados a uma condição dramática do processo. Conforme mostra a Figura 8, o crítico não faz parte do PDM, ele é um crítico externo ao processo, pois pode ser projetado de forma independente, pelo projetista, desde que este não escolha valores com reforços infinitos. Sendo assim, o critério de observação será do agente para as políticas estacionárias μ.

3.4.2.2. Critérios de Desempenho para Utilidade de Estado

A Equação (2) (BELLMAN, 1957) fornece um método para calcular as utilidades para cada um dos estados, sem precisar calcular as utilidades como reforços associados ao sair do estado inicial até chegar ao estado objetivo. Com isso, têm-se a soma descontada. A utilidade que está sendo calculada, embora a equação relacione utilidade com estado e com a utilidade do estado sucessor. O V^{μ} (s) (para qualquer estado) é o reforço obtido numa política μ mais o reforço obtido do sucessor de (s) seguindo a mesma política, obtendo assim, a equação (3).

Equação 2: $V^{*}(s) = E\{ r(s,\mu^{*}(s)) + \gamma V^{*}(s')\}$

Equação 3: $V^{\mu}(s) = E_{\mu}\{R_t | s_t = s\}$

Para cada possível política μ que o agente pode adotar, define-se uma função de avaliação sobre os estados, onde r_t, r_{t+1}, são gerados seguindo-se a política μ, começando pelo estado s.

3.4.2.3. Maximização da Utilidade dos Estados

O agente precisa estabelecer uma política de ações que maximize o critério de desempenho da utilidade dos estados (equação 4), ou seja, uma política ótima. A política é ótima quando consegue produzir o maior valor de utilidade possível para todos os possíveis estados de (s).

Formalmente, esta política ótima μ*deve ser:

Equação 4: $V\ \mu^*(S)=V^*= \max \mu\ V\ \mu(S)$

A política ótima μ é a melhor que o agente pode seguir para a utilidade de um estado **s** qualquer, para todas as possíveis políticas μ, V(s) é a maior utilidade entre todos os **s**.

3.4.3. Técnicas de Aprendizado por Reforço

O Aprendizado por Reforço - AR permite ao agente adquirir uma capacidade de conhecimento do ambiente que não estava disponível em tempo de projeto (SUTTON e BARTO 1998). AR é baseado na existência de um crítico externo ao ambiente, que avalia a ação tomada, mas sem indicar explicitamente a ação correta.

Formalmente, AR utiliza uma estrutura composta de estados, ações e reforços conforme mostra a Figura 10.

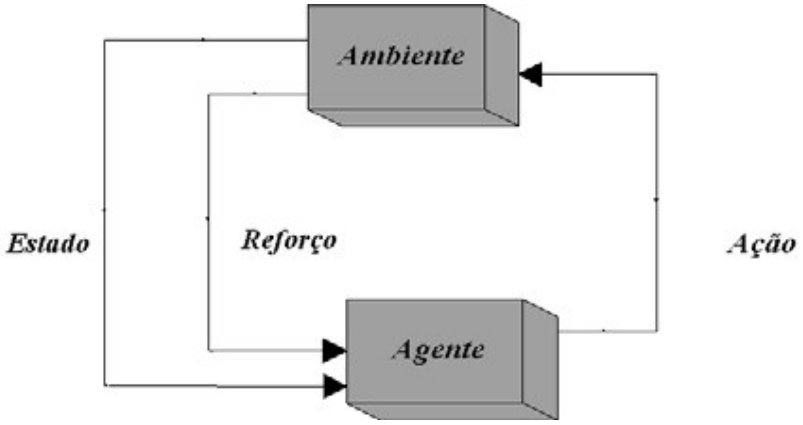

Figura 10.Um agente AR interagindo com seu ambiente.

O agente existe em um ambiente descrito por um conjunto de possíveis estados e pode executar, para cada estado, uma ação dentro de um conjunto de ações possíveis, recebendo um valor de reforços a cada vez que executa uma ação. Este reforço indica o valor imediato desta transição estado-ação. Isto produz uma sequência de pares estado-ação, e seus respectivos valores de reforços. O objetivo do agente é aprender uma política que maximize a soma esperada destes reforços a longo prazo (GUELPELI *et al,*2003).

No contexto deste trabalho, a técnica de AR terá como função modificar parâmetros e armazená-los em "esquemas de planos", que definam a forma de representar de cada estado. O grande desafio, neste caso, é escolher a melhor ação que – baseada no estado de cada texto e/ou categoria (Estado) - possa mudar a estratégia de *clusterização* e/ou grau de compressão (Ação), para obter resultados significativos, fornecedores de parâmetros indicam quão boa ou ruim está sendo uma determinada estratégia. O algoritmo utilizado (*Q-learning*) é baseado em estimativas de utilidades de pares estado-ação, e com estas estimativas podem-se gerar alternativas de

novas estratégias para o modelo *Cassiopeia*. Neste caso, a aprendizagem das estratégias mais adequadas, que indicam a ação a escolher para cada estado do texto e/ou da categoria dar-se-á por reforço, através de uma estrutura de parâmetros adaptativos sobre os quais o algoritmo opera.

AR procura aproximar uma função que define a utilidade relativa dos pares estado-ação. Esta função de utilidade fornece indicações estimativas, mapeando os pares estado-ação em uma medida baseada na soma dos reforços esperados a longo prazo (GUELPELI *et al,*2004).

Para cada par estado-ação (s, a) é definido o reforço r (s, a), indicando uma consequência imediata da execução da ação a no estado s. O problema em AR é achar uma política ótima de ações (μ*), ou seja, um conjunto de ações que maximizem, para cada estado s, os valores de utilidade Q (s, a). Com base nesses valores, o algoritmo de AR estima a ação de maior valor de utilidade – a ação que deverá ser executada pelo agente.

3.4.4. Algoritmo Q-Learning

O algoritmo *Q-Learning* foi proposto em 1989 (WATKINS, 1989), para definir a escolha da melhor ação, no aprendizado por reforço. No *Q-Learning* a escolha de uma ação é baseada em uma função de utilidade que mapeiam estados e ações a um valor numérico.

Neste trabalho, o valor de utilidade Q (s, a) de um par (estado(s), ação (a)), é calculado a partir de reforços medidos pela qualidade do estado do texto e/ou das categorias. Portanto, o principal objetivo do algoritmo *Q-Learning* é estimar autonomamente, em cada estado que o texto e/ou as categorias se encontram, a ação com maior valor de utilidade. Como consequência, conseguirá estimar o estado do texto e/ou das categorias.

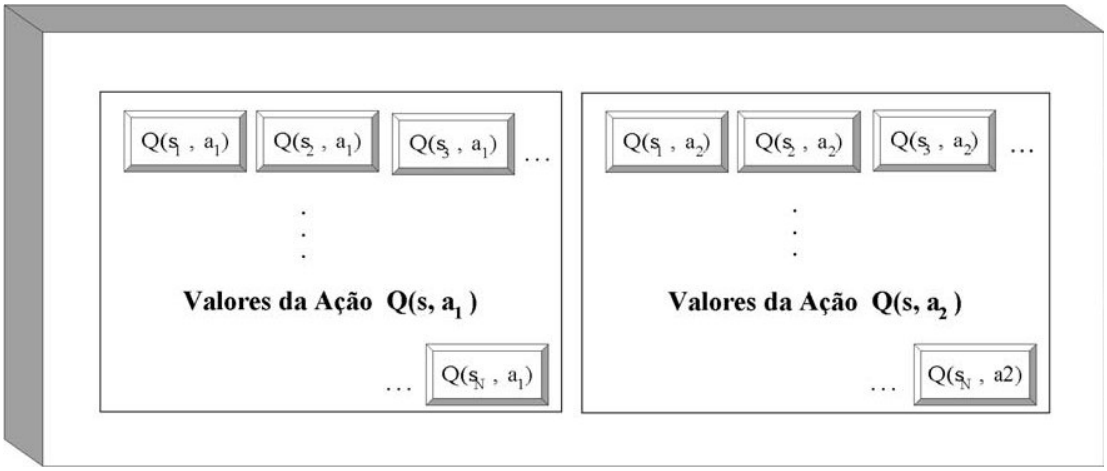

Figura 11.Tabela dos pares (estado - ação).

O *Q-Learning* produz uma atualização dos valores Q na direção $\max_a Q^{\mu}_t(s_{t+1}, a)$, originando o seguinte algoritmo:

5-Algoritmo Q-Learning:

Inicialize **Q(s, a)**.

Para cada instante **t** repita:

1- Observe estado s_t e escolha ação a_t de acordo com a política de ações (μ);

2- Observe o estado s_{t+1} e atualize $Q^{\mu}_t(s_t, a_t)$ de acordo com

$$Q^{\mu}_{t+1}(s_t,a_t) = Q^{\mu}_t(s_t,a_t) + \alpha_t[r(s_t) + \gamma \max_a Q^{\mu}_t(s_{t+1},a) - Q^{\mu}_t(s_t,a_t)];$$

até **t** igual a limite de passos.

Pode-se definir:

- $Q^{\mu}_{t+1}(s_t,a_t)$ **-** É o valor (qualidade) da ação a_t no estado s_t, seguindo a política de ações (μ)**.**

- $r(s_t)$ **-** É o reforço imediato recebido no estado s_t .

- α **-** É a taxa de aprendizagem.

- γ **-** É a taxa de desconto.

- **t -** É uma sequência discreta de passos no tempo, ou seja, t=0, 1, 2, 3,...

- $\mathbf{max_a Q^{\mu}_t(s_{t+1}, a)}$ **-** política de maximização, que escolhe na tabela (Figura 11) a ação com maior valor de utilidade no estado futuro.

- O fator γ**(entre 0 e 1)** - quanto mais perto de 1**,** mais importância é dada aos reforços mais distantes no tempo.

No início, o algoritmo estará muito longe da utilidade ótima associada ao estado s, mas com o passar do tempo, as estimativas de Q melhoram. De fato, os valores Q atualizados pelo algoritmo *Q-Learning* convergem para os corretos, desde que os pares (s, a) tenham sido visitados um número infinito de vezes (WATKINS e DAYAN 1992). Na prática, a convergência para políticas de ações adequadas é obtida apenas com a exploração do espaço de estados, durante um número razoável de iterações.

Além do Q-learning, existem vários outros métodos (DIETTERICH, 1996) (KAELBLING, 1995) como: *Average Reward Reinforcement Learning, Actual Return Reiforcement Learning*, *Feudal Q-Learning, Compositional Q-Learning, Hierarchical Q-Learning e Recurrent Q-Learning*.

A razão pela qual o *Q-Learning* é um dos mais usados e pela qual justifica sua utilização neste trabalho é porque, para aprender a utilidade dos estados, é necessária uma descrição bastante detalhada do ambiente, o que nem sempre é possível (neste trabalho seria a quantidade de palavras dos textos a serem sumarizados e os agrupamentos onde estes textos serão *clusterizados*). Para este tipo de caso, considera-se o *Q-Learning* um método mais "econômico", porque se baseia apenas em valores de qualidade, para cada ação em cada estado, muitas vezes mais fáceis de determinar.

3.4.5. Considerações Finais

Nesta seção foi apresentado o conceito de aprendizado autônomo, foi definido formalmente o processo decisório de *Markov* para a representação da modelagem no modelo Cassiopeia e o uso da técnica de AR para calcular o valor de utilidade Q (s, a) de um par (estado(s), ação (a)), a partir de reforços medidos pela qualidade do estado do texto e/ou dos agrupamentos. Foi definido também o algoritmo *Q-Learning* que estima autonomamente, em cada estado, que o texto e/ou os agrupamentos se encontram. Na seção 4 serão apresentados os resultados preliminares.

4. Resultados Preliminares

4.1. Módulo Sumarização

Na primeira simulação foram realizadas comparações com algoritmo *Perfil* e algoritmo conhecido na literatura (*TextTiling*).

Na segunda simulação, algoritmo *Perfil,* o sumarizador *GistSumm* (PARDO, 2002) e a ferramenta A*utoresumo do MS-Word* (Microsoft®Word, 2002).

Seguindo a estratégia do uso de *extratos ideais*, que são resumos do texto original para ser comparado com resultados obtidos de sumarizadores automáticos, o trabalho adotou o *corpus Temário*, (PARDO E RINO 2006) para realizar os testes dos sumários gerados. Este *corpus* é composto por 100 textos, que são classificados por sumários e textos fontes. Os textos fontes são divididos por texto fonte com origem e título, texto fonte com título e texto fonte sem título. Já os sumários são classificados por: extratos ideais automáticos, sumários manuais e sumários manuais marcados. No diretório texto fonte com origem e título encontra-se uma subdivisão com textos de dois jornais de grande circulação no Brasil (Folha de São Paulo e Jornal do Brasil).

No diretório da Folha de São Paulo, nos textos fontes, existe a seção Especial Mundo e Opinião.

Já na seção do Jornal do Brasil encontram-se os textos fontes classificados por Internacional e Política. Para realizar os experimentos, foram usadas as seções grifadas conforme ilustrado na Figura 12.

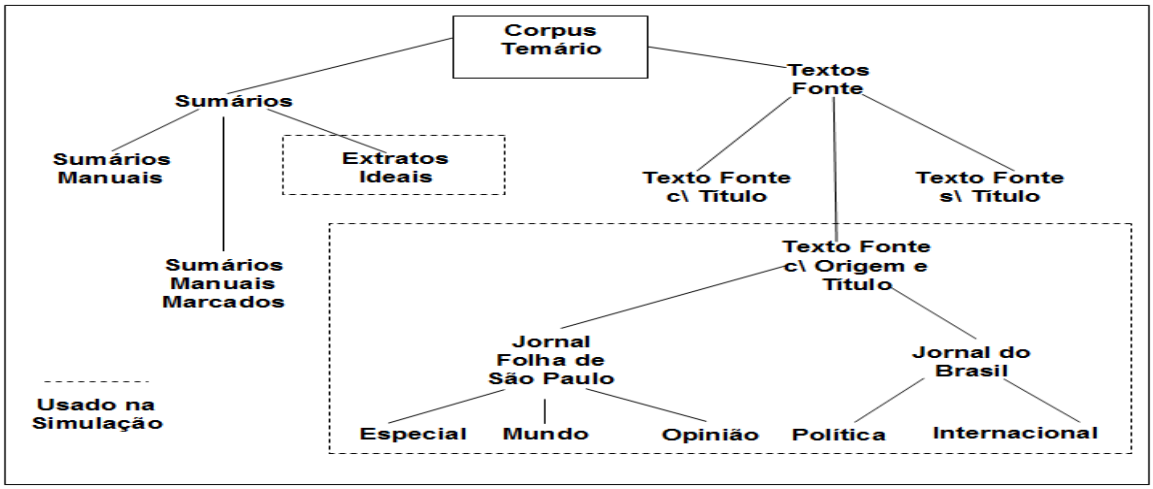

Figura 12.Taxonomia do *Corpus Temário*.

Este trabalho analisa o comportamento Sumarizador Automático - SA usando o algoritmo *TextTiling* formalizada por (SALTON e BUCKLEY, 1988) e o comportamento do SA utilizando o algoritmo *Perfil*, que foi formalizado por (GUELPELI *at al*, 2007) e faz parte desta proposta de trabalho. Estes valores foram comparados com o *extrato ideal*, que é o melhor tipo de sumário de referência para a avaliação de sistemas de SA, por conter somente sentenças do texto-fonte, podendo ser comparado mais facilmente com um sumário automático, já que este, também se origina do mesmo texto-fonte. No *corpus Temário* o extrato ideal foi composto pelas sentenças do texto-fonte julgadas por humanos como essenciais para compor um sumário do texto. Utilizaram-se as seguintes medidas de avaliação, nas quais foram considerados os aspectos qualitativos e o conteúdo de um sumário:

- *Recall* (R): número de sentenças do sumário automático presente no sumário de referência / número de sentenças do sumário automático;

- *Precision* (P): número de sentenças do sumário automático presente no sumário de referência / número de sentenças do sumário de referência;
- *Medida-F* (*F-Measure*): $\frac{PxR}{P+R}x2$, a medida F combina as métricas de *recall* e de *precision*. O resultado da medida F é um indicativo de que, quanto mais próximo de 1, melhor é o sumário, e resultados mais próximos de 0, demonstra que os sumários são ruins, ou seja, longe do proposto pelo extrato ideal.

Outra medida usada para avaliação foi a compressão (C_r – compression rate) e o número de sentenças do texto original mantidas no sumário (NS_{org}), C_r é dado pelo número de sentenças eliminadas para compor o sumário e NS_{org} é dada pelo número de sentenças do texto original mantidas no sumário. C_r varia de 0 e 1, quanto menor o valor de C_r maior é o sumário; o desejado é que seja mais próximo de 1.

Para validar as médias obtidas na Tabela 4 usou-se a análise de variância. Este método estatístico é usado para testar diferenças entre médias de amostras e suas combinações lineares. Classicamente, o teste-t é a ferramenta utilizada para investigar diferenças entre duas médias, sendo que a análise de variância também é utilizada para esta finalidade. Os parâmetros da Tabela 5 foram analisados pelo teste "t" *Student*, considerando variâncias diferentes. As duas amostras (independentes) foram coletadas na simulação.

Em Guelpeli *et al* (2007) e Guelpeli *et al* (2008) são mostrados outros resultados como teste de hipóteses, que são usados para as duas amostras coletadas na simulação dos algoritmos *TextTiling* e *Perfil* no *corpus Temário.*

Na Tabela 5 são apresentados os resultados das médias e do desvio padrão obtido nas aplicações das medidas em todas as seções do *corpus*. Observa-se que os resultados obtidos no algoritmo *Perfil* são, significativamente na média, melhores em todas as seções e usando qualquer métrica.

Tabela 5.Resultado das Médias e do Desvio Padrão da aplicação do algoritmo *TextTiling* e *Perfil* no *corpus Temário* com todas as medidas.

Jornais	Seções		Recall		Precision		F-Mesure		C_R		NS_{org}	
			Text Tiling	Perfil	Text Tiling	Perfil	Text Tiling	Perfil	Text Tiling	Perfil	Text Tiling	Perfil
Folha SP	Mundo	Med	25.72	37.51	33.33	*46.24*	28.17	39.94	61.65	64.00	38.35	36.00
		DP	13.69	13.38	15.65	15.31	13.91	13.18	5.62	5.06	5.62	5.06
	Opnião	Med	23.62	35.80	28.78	37.87	25.48	35.94	60.40	66.40	39.6	34
		DP	8.70	14.68	10.77	16.17	8.88	14.44	6.74	5.71	6.74	6.48
	Especial	Med	25.72	37.72	33.45	44.11	25.84	40.43	59.90	63.80	39.85	36.20
		DP	13.58	13.96	17.03	14.89	13.43	14.00	7.56	4.34	7.56	4.34
JB	Inter nacional	Med	36.16	52.90	38.91	46.92	33.95	45.85	59.05	67.20	40.28	32.80
		DP	21.57	22.52	17.07	17.27	12.89	13.81	12.14	9.64	12.14	9.64
	Política	Med	25.30	40.07	33.61	47.18	28.05	42.27	59.35	64.90	40.13	34.65
		DP	10.80	16.37	10.06	13.88	9.38	13.52	10.10	3.65	10.10	4.82

Os resultados obtidos na Tabela 5 mostra que as amostras são iguais na seção Internacional, usando a medida *precision,* e nas seções Mundo e Especial, usando as medidas C_R e $NS_{orig.}$. Sendo assim, não houve diferença

significativa entre as variâncias, determinando a igualdade estatística entre suas médias. Este resultado foi obtido no grau de significância de 5%.

Nos outros casos, o que equivale 80% dos resultados representados na Tabela 6, mostra que houve uma diferença significativa entre as médias obtidas pelos algoritmos *TextTiling* e *Perfil*, onde o grau de significância também foi 5%. Pode-se observar ainda que esta diferença esta também presente no grau de significância de 1% nas seguintes seções, com suas respectivas medidas: Opinião (*Recall*, *F-Mesure* e C_R), Política (*Recall, Precision* e *F-Mesure*), Especial (*Recall* e *F-Mesure*) e Internacional (*Recall*).

Tabela 6.Comparativo entre os algoritmos *TextTiling* e *Perfil* usando o Teste t de variância Diferente.

Jornais	Seções		Recall		Precision		F-Mesure		C_R		NS_{orig}	
			t_{cal}	t_{tab}	t_{cal}	t_{tab}	t_{cal}	t_{tab}	t_{cal}	t_{tab}	t_{cal}	t_{tab}
Folha SP	Mundo	5%	2.588	2.024	2.481	2.024	2.655	2.024	**1.389**	2.024	**1.389**	2.024
	Opnião	5%	3.193	2.024	2.092	2.024	2.758	2.024	3.038	2.024	2.678	2.024
	Especial	5%	2.757	2.024	2.106	2.024	3.361	2.024	2.001	2.024	**2.001**	2.024
JB	Inter	5%	2.400	2.024	**1.474**	2.024	2.817	2.024	2.351	2.024	2.351	2.024
	Política	5%	3.368	2.024	3.541	2.024	3.865	2.024	2.310	2.024	2.397	2.024

Na segunda simulação são apresentados os resultados com um sumarizador já implementados como: *GistSumm* (Pardo,2002) e a ferramenta A*utoresumo do MS-Word* (Microsoft®Word, 2002).

Estas simulações também usaram o *corpus TeMário* (PARDO E RINO 2003) ilustrado na Figura 12 que segue a estratégia do uso de *extratos ideais*, para comparação com testes dos sumários gerados.

As simulações diferem-se dos resultados obtidos nas Tabelas 5 e 6 devido aos sumarizadores e ao grau de compressão, que agora foi ajustado igualmente para todos, não seguindo o *default*. Como algoritmo P*erfil* não permite ajustes manuais, tanto *GistSumm*, como A*utoresumo do Word* foram ajustados ao mesmo percentual de compressão do *Perfil*, sendo assim, não foi mais viável a apresentação dos resultados das métricas C_R e NS_{orig} que neste caso apresentaram-se iguais para *Perfil, GistSumm e Autoresumo do Word* para cada texto sumarizado.

Tabela 7.Resultado das Médias e do Desvio Padrão da aplicação do algoritmo *Perfil*, *GistSumm* e *Word* no *corpus Temário* com todas as medidas.

Jornais	Seções		Recall			Precision			F-Measure		
			Gist Summ	Word	Perfil	Gist Summ	Word	Perfil	Gist Summ	Word	Perfil
Folha SP	Mundo	Med	***36.27***	27.17	36.05	***49.16***	35.88	48.90	***40.87***	30.11	40.65
		DP	11.96	13.95	11.17	17.28	14.98	12.59	12.35	13.00	10.41
	Opnião	Med	***38.98***	28.03	33.94	***38.31***	35.44	37.99	***37.61***	30.59	36.64
		DP	13.77	13.07	13.45	15.25	19.34	19.15	13.03	14.87	16.23
	Especial	Med	33.56	30.83	***37.45***	39.41	39.31	***46.77***	35.86	34.24	***41.83***
		DP	13.18	10.98	13.32	13.59	13.09	15.26	12.70	11.19	12.89
JB	Inter nacional	Med	60.53	32.34	***83.37***	***46.86***	34.72	32.65	***48.88***	29.76	40.80
		DP	23.44	15.04	23.73	22.49	29.85	20.56	16.28	16.58	16.54
	Política	Med	33.69	29.15	***39.52***	42.73	37.99	***51.36***	36.71	32.08	***43.38***
		DP	14.47	10.95	15.64	10.77	10.89	13.84	11.67	9.26	12.34

Para validar as médias obtidas na Tabela 7 usou-se novamente a análise de variância (que é o método estatístico onde testa as diferenças entre médias de amostras e entre combinações lineares das médias). O teste-t é a forma utilizada para investigar diferenças entre duas médias.

Tabela 8.Comparativo entre os *Perfil* e *GistSumm* usando o Teste t de variância Diferente.

Jornais	Seções		Recall		Precision		F-Mesure	
			t_{cal}	t_{tab}	t_{cal}	t_{tab}	t_{cal}	t_{tab}
Folha SP	Mundo	5%	0.0594	2.0244	0.0556	2.0244	0.0616	2.0244
	Opnião	5%	1.1713	2.0244	0.0596	2.0244	0.2081	2.0244
	Especial	5%	0.9282	2.0244	1.6091	2.0244	1.4754	2.0244
JB	Inter	5%	**3.0623**	2.0244	**2.0862**	2.0244	1.5558	2.0244
	Política	5%	1.2227	2.0244	**2.2013**	2.0244	1.7571	2.0244

As duas amostras (independentes) foram coletadas na simulação do resultado das duas maiores médias, ou seja, *Perfil* e *GistSumm*.

Na Tabela 8 o teste-t mostra que as médias e as variâncias em 86,66% dos casos são iguais, não havendo diferença significativa entre as médias do *Perfil* e *GistSumm*. Apenas 13,34% das médias são significativamente diferentes. Isto ocorre na seção Internacional com as métricas *Recall* (*Perfil*), *Precision* (*GistSumm*) e na seção Política na métrica *Precision* (*Perfil*). Estes resultados, apesar de parelhos são significativamente favoráveis ao *Perfil*, já que este preserva as *stopwords*. Para o modelo *Cassiopeia* estes resultados são

animadores uma vez que o nível de compressão será aprendido e definido automaticamente.

4.2. Módulo de *Clusterização*

Para o experimento com o *clusterizador* proposto por Guelpeli *et al* (2007) e Guelpeli *et al* (2008) e usado nesta simulação foi escolhido o *clusterizador Eurekha* de Wives (1999); Wives (2000). Esta escolha justifica-se pelo fato do *Eurekha* usar o mesmo algoritmo grafoteoréticos (*star e cliques*) e trabalhar com lógica *Fuzzy*, porém com funções difusas diferentes e uso da abordagem por conceito. O *Eurekha* usa a função *fuzzy* para calcular a similaridade, que leva em conta as diferenças e as semelhanças entre os documentos. Já o *clusterizado*r baseia na idéia da similaridade do texto em relação ao agrupamento.

Em Guelpeli *et al* (2007) e Guelpeli *et al* (2008) foram feitas análises qualitativas dos agrupamentos construídas pelo *clusterizador* e pelo *Eurekha*. Foram usados os *Corpus*: *Temário* (Pardo e Rino, 2003), Reuters-21578[2], Distribution 1.0 e Really Simple Syndication (RSS)[1] [3], obtendo outros resultados e estudos como: análise de variância, para testar diferenças entre médias de amostras e entre combinações lineares das médias. Teste de

---

[1] *Corpus* extraído do Terra Networks Brasil S/A.

[2] A coleção completa possui 1578 textos, porém estes arquivos não estavam disponíveis para uso em sua totalidade. Por esse motivo, foram utilizados somente os 100 textos disponíveis on-line.

[3] Esses arquivos, provenientes dos seus mais diversos canais de RSS do Terra Networks Brasil S/A, foram coletados diariamente durante o período de fevereiro à março de 2008.

hipóteses: usados para as amostras coletadas na simulação do *clusterizador* e do *Eurekha* no *corpus Temário, corpus Reuters* e *RSS_Terra.*

Os resultados apresentados neste trabalho foram baseados apenas no *corpus TeMário* ilustrados na Figura 2. Isto se deve ao fato do *corpus* já ter uma classificação feita e assim permite uma mensuração dos agrupamentos feitos pelo *clusterizador* e pelo *Eurekha*. Para os resultados apresentados nas Tabelas 9 e 10, na fase de pré-processamento, foi usado a sumarização para redução do número de palavras.

Para tratar da mensuração na *clusterização*, a medida *precision* é analisada no âmbito de cada agrupamento, sendo então o número de textos corretamente associados ao agrupamento considerado, dividido pelo número de textos alocados no agrupamento pela técnica ou sistema que está sendo avaliado. Enquanto que *recall* divide o número de textos corretamente associados ao agrupamento pelo total de textos que deveriam estar presentes no agrupamento em questão (LOH, 1999). Estas são as medidas adotadas para avaliação do *corpus TeMário*. É importante lembrar que, segundo Sparck *et al* (1997), nos sistemas de MT, principalmente em RI, por razão das técnicas empregadas nestes, quando se consegue um aumento numa das medidas, há uma diminuição na outra, levando-se em conta alguns problemas de avaliação que cercam esta área de MT.

Segundo Loh (1999), os critérios utilizados para avaliação dos documentos relevantes e não relevantes (tanto os recuperados ou classificados, quanto na coleção toda) é o julgamento por especialistas humanos e os problemas físicos (milhões de documentos), além da questão da

subjetividade no processo de determinação da relevância ou não dos documentos.

Afirma Loh (1999), que a relevância é uma graduação e não uma simples decisão binária, envolvendo somente dois tipos de resposta (sim ou não). Ela pode mudar à medida que o conhecimento disponível muda (por exemplo, documentos podem se tornar mais relevantes com o passar do tempo).

Em geral, as medidas de avaliação não levam em conta a necessidade do usuário (HERSH, 1995). Por exemplo, às vezes basta obter um documento para que se tenha a informação. Em outras situações é preciso ter noção da coleção inteira, principalmente quando a informação pode mudar com o tempo.

Algumas propostas na literatura para solucionar a questão de avaliação: (KORFHAGE, 1997) medidas orientadas ao usuário; (MORITA *et* al, 1994) medidas por monitoração de tempo (considerando que o usuário gasta mais tempo para a leitura de documentos mais interessantes); (SARACEVIC, 1997) esquema para a definição de relevância, segundo vários pontos de vista.

Como este capítulo trata de resultados preliminares para o modelo *Cassiopeia* e o grande foco é validar a escolha deste tipo de abordagem, pode-se mostrar que a proposta do *clusterizador* é no mínimo igual aos outros softwares existentes. Desde que estes tenham como objetivo agrupamento textual, usando a lógica difusa e abordagem hierárquica aglomerativa juntamente com o algoritmo *Cliques* que, segundo Guelpeli *et AL* (2007) e

Guelpeli *et al* (2008), obtém resultados mais coesos . Para avaliação usar-se-á:

- *Recall:* número de textos corretamente associados ao agrupamento/ total de textos que deveriam estar presentes no agrupamento
- *Precision:* número de textos corretamente associados ao agrupamento/ número de textos alocados neste agrupamento

A Tabela 9 ilustra os resultados mostrando as médias e seus desvios padrões para *clusterizador* e o *Eurekha,* usando os resumos gerados pelo *Perfil*, *GistSumm* e A*utoresumo* do *Word* no *corpus TeMário*. Verifica-se que a amostra do *Eurekha* obteve os melhores resultados para a medida *Recall* usando os resumos do *Perfil*, *GistSumm* e *Autoresumo* do *Word* e também para a medida *Precision* usando os resumos *Perfil*, *GistSumm* A média do *clusterizador* foi melhor usando a medida *Precision* e *Autoresumo* do *Word.*

Tabela 9.Resultado das Médias e do Desvio Padrão da aplicação do *Eurekha* e *Clusterizador* usando resumos gerados pelos métodos *Perfil*, *GistSumm* e *Word* no *corpus Temário*.

	Eurekha						***Categorizador***					
	Recall			**Precision**			**Recall**			**Precision**		
	Gist Summ	**Word**	**Perfil**	**Gist Summ**	**Word**	**Perfil**	**Gist Summ**	**Word**	**Perfil**	**Gist Summ**	**Word**	**Perfil**
NºAgrupamentos	4	4	4	4	4	4	**33**	**32**	**24**	**33**	**32**	**24**
Média	**63.60**	**63.60**	**73.97**	**80.00**	63.60	**85.00**	8.94	9.06	12.29	**64.79**	66.66	71.29
DP	11.68	13.48	17.46	28.28	13.48	30.00	5.27	4.99	8.47	24.60	25.21	25.99

Cabe aqui ressaltar que o *Eurekha* gerou apenas 4 agrupamentos para os 100 textos, enquanto o *clusterizador obteve* 33 agrupamentos para o resumo *GistSumm*, 32 para autoresumo do *Word* e 24 para *Perfil* para os mesmos 100 textos.

Para validar as médias obtidas na Tabela 8 usou-se a análise de variância e o teste-t para investigar diferenças entre duas médias.

Tabela 10.Comparativo entre *Eurekha* e *Clusterizador* usando resumos gerados pelos métodos *Perfil*, *GistSumm* e *Word* no *corpus Temário* usando o Teste t de variância Diferente

Corpus TeMário Sumarizados		Recall		Precision	
		t_{cal}	t_{tab}	t_{cal}	t_{tab}
Gist Summ	5%	**16.1331**	2.0301	1.1521	2.0301
Word	5%	**16.5210**	2.0322	0.2362	2.0322
Perfil	5%	**11.5005**	2.0555	0.9583	2.0555

Na Tabela 10 o teste-t mostra que as médias e as variâncias em 50% dos casos são iguais, não havendo diferenças significativas entre as médias do *Eurekha* e do *clusterizador,* usando os resumos gerados pelo *Perfil*, *GistSumm* e autoresumo do *Word.*

Estes resultados, apesar de ruins são significativamente favoráveis ao *clusterizador*, já que este, tem a finalidade segundo Guelpeli *et al* (2007) e

Guelpeli *et al* (2008) de ser mais específico, isto é, comprovado pelo número de agrupamentos criados.

Para o modelo *Cassiopeia* com o uso do aprendizado autônomo estes resultados podem ser melhorados.

4.3. Considerações Finais

Nesta seção foram apresentados os resultados preliminares da simulação usando a sumarização e a *clusterização*.

Na sumarização foram realizadas duas simulações. A primeira usou a sumarização com algoritmo *Perfil* e com algoritmo conhecido na literatura (*TextTiling*). Na segunda simulação usou o algoritmo *Perfil* e o sumarizador *GistSumm* (PARDO, 2002) e a ferramenta A*utoresumo do MS-Word* (Microsoft®Word, 2002).

A *clusterização* usou- o experimento com o *clusterizador* proposto por Guelpeli *et al* (2007) e Guelpeli *et al* (2008) e *clusterizador Eurekha* de Wives (2000); Wives (2004).

Os resultados obtidos preliminarmente ainda podem ser melhorados com uso do aprendizado autônomo, que será implementado no modelo *Cassiopeia*.

5. Considerações Finais

No modelo *Cassiopeia* ainda falta implementar o módulo de aprendizado autônomo. Acredita-se que inclusão deste módulo, venha obter resultados significativos para métrica de coesão (escolhida para realização dos testes no modelo, quando este estiver com todos os seus módulos implementados).

Outro fator que merece uma atenção futura é a questão do pós-processamento no modelo *Cassiopeia*. Como os índices dos agrupamentos estão altamente estimados e as palavras indexadas têm uma forte correlação com os textos ali agrupados, seria interessante usar alguma técnica para transformar estas palavras em categorias e, com isto, melhorar ainda mais a descoberta de conhecimento em textos.

A questão do *corpus* é outro detalhe a ser visto para os próximos testes no qual existem outras possibilidades de simulação: uso em outras línguas, com alta e baixa correlação e com número maior de textos. Único problema esta relacionado à necessidade deste corpus estarem classificados previamente por especialistas para facilitar as comparações e com isto validar a métrica de coesão proposta neste trabalho.

Por fim, o modelo Cassiopeia poderia ser usado como ferramenta onde usaria a palavra para formulação da consulta e com isto conseguir recuperar textos com maior similaridade em relação à consulta formulada.

6. Referências Bibliográficas

Aldenderfer, M. S.; Blashfield, R. K.(1984) ***Cluster Analysis***. Beverly Hills, CA:Sage, 1984. p88.

Arora, R. e Bangalore, P.(2005). ***Text mining: classification & clustering of articles related to sports***.Proceedings of the 43rd annual Southeast regional conference - Volume 1 ACM-SE 43 Publisher: ACM Press, March 2005.

Baxendale, P. B. (1958). ***Machine-made index for technical literature – an experiment***. IBM Journal of Research and Development, 2, pp. 354-361.

Bellman, R. (1957). ***Dynamic programming***. Princeton: Princeton University Press, 1957.

Berkhin, P. (2002). ***Survey of Clustering Data Mining Techniques.*** Accrue Software,2002.

Crestani, F; Rijsbergen, C. J.(1997). ***A model for adaptative information retrieval.*** Journal of Intelligent Information Systems, v.8, 1997. p.29-56.

Cross, V.(1994)***Fuzzy information retrieval.*** Journal of Intelligent Information Systems, Boston, v.3, n.1, p.29-56, 1994.

Dagan, I., Feldman, R, e Hirsh, H. (1996). ***Keyword-based browsing and analysis of large document sets.*** In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR), Las Vegas, NV.

Dietterich, T G.(1996). ***Machine Learning***, Netherlands, V.22, Jan/Feb/Mar, 1996

Ebecken, N. F. F. e Lopes, M C S. Costa, M. C. A. ***Mineração de Textos. In:. Sistemas Inteligentes: Fundamentos e Aplicações.*** Manole: São Paulo, 2003.

Edmundson, H. P. (1969). ***New Methods in automatic extracting.*** Journal of the ACM,16, pp. 264-285.

Everitt, B.S.; Dunn, G.(2001). ***Applied multivariate analysis.*** 2nd. ed. London: Arnold, 2001.

Fasulo, D.(1999). ***An Analysis of Recent Work on Clustering Algorithms.*** Technical Report, Dept. of Computer Science and Engineering, Univ. of Washington, 1999.

Fávero, L. L.(2000). ***Coesão e Coerência Textuais***. 9. ed. São Paulo: Ática, 2000.

Fayyad, U e Uthurusamy, R. (1999). ***Data mining and knowledge discovery in databases: Introduction to the special issue.*** Communications of the ACM, 39(11), November.Usama Fayyad. 1997. Editorial. Data Mining and Knowledge Discovery.

Feldman, R. , Hirsh, H.(1997). ***Exploiting background information in knowledge discovery from text.*** Journal of Intelligent Information Systems, v.9, n.1, Julho/Agosto de 1997.

Jain A. K. e Dubes R. C.(1988). ***Algorithms for Clustering Data***, Prentice Hall,1988. Disponível em http://www.cse.msu.edu/~jain/Clustering_Jain_Dubes.pdf .Acessado em 25 de junho de 2007.

Jianan, W; Rangaswamy, Arvind.(1999). ***A fuzzy set model of consideration set formation calibrated on data from an online supermarket.*** EBusiness research Center Working Paper, n. 5, 1999.

Kaelbling, L. P. (1995).***The importance of being adaptable. In: The Artificial Life Route to Artificial Intelligence.*** Building Embodied, Situated Agents. Steels, Luc; Brooks, Rodney (eds.). Lawrence Erlbaum Assoc., USA, 1995.

Karypis, G., Han, E.-H. S.; and NEWS, V. K. (1999). ***Chameleon: Hierarchical clustering using dynamic modeling.*** Computer, 32(8):68–75.

Klir, G. J.; Folger, T. A (1988). ***Fuzzy sets, uncertainty, and information***. EnglewoodCliffs, NewJersey: Prentice Hall, 1988.

Kwok R.C. W.; Ma J. e Zhou D. (1998). ***Improving Group Decision Making: A Fuzzy GSS Approach***, Disponível em citeseer.ist.psu.edu/668925.html. Acessado em 22 de junho de 2007.

Guelpeli, M.V.C.(2003). ***Utilização de Aprendizado por Reforço para Modelagem Autônoma do Aprendiz em Sistemas Tutores Inteligentes***, Instituto Tecnológico da Aeronáutica – ITA-SP Dissertação de Mestrado-2003.

Guelpeli, M.V.C., Ribeiro, C. H. R., Nizam, O. (2003). ***Aprendizado por Reforço para um Sistema Tutor Inteligente sem Modelo Explícito do Aprendiz.*** Anais do XIV-Simpósio Brasileiro de Informática na Educação-(SBIE2003)-Realizado na UFRJ - mês de novembro de 2003 – pág 493-502 - ISBN 85-88442-70-1.

Guelpeli, M.V.C., Ribeiro, C. H. R., Nizam, O. (2004). ***Aprendizado por Reforço para um Sistema Tutor Inteligente sem Modelo Explícito do Aprendiz***. Revista Brasileira de Informática na Educação- RBIE – SBC Volume 12 – Número 2 pág. 69-77 - mês de Julho a Dezembro de 2004 ISSN 1414-5685.

Guelpeli, M.V.C.; Garcia A.C.B.(2007). ***Automatic Summarizer Based on Pragmatic Profiles,*** International Conference WWW/Internet 2007- IADIS- Volume II pág. 149-153- ISBN: 978-972-8924-44-7 - mês de Outubro de 2007- Vila Real-Portugal .

Guelpeli, M.V.C.; Garcia A.C.B.(2007). ***An Analysis of Constructed Categories for Textual Classification Using Fuzzy Similarity and Agglomerative Hierarchical Methods***, proceedings of The 2007 IEEE International Conference on SIGNAL-IMAGE TECHNOLOGY and INTERNET- BASED SYSTEMS. Shanghai, China 16 a 19 de December, 2007.

Guelpeli, M.V.C.; Garcia A.C.B. Bernadini, F. C.(2008). ***An Analysis of Constructed Categories for Textual Classification using Fuzzy Similarity and Agglomerative Hierarchical Methods.*** an Proceding. An Analysis of Constructed Categories for Textual Classification using Fuzzy Similarity and Agglomerative Hierarchical Methods, LNCS-Studies in Computational Intelligence. ISBN: 1860-949X, Berlim: Springer Verlag, 2008

Guelpeli, M. V. C.; Bernardini, F. C. ; Garcia, A. C. B.(2008). ***Todas as Palavras da Sentença como Métrica para um Sumarizador Automático.*** In: Tecnologia da Informação e da Linguagem Humana-TIL, 2008, Vila Velha. Todas as Palavras da Sentença como Métrica para um Sumarizador Automático. Vila Velha : WebMedia, 2008. p. 287-291.

Grehs, G. H. (2002).***Um Estudo sobre a influência da variação morfológica das palavras no processo de agrupamento automático de textos (clustering***). 2002. 74 f. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Informática,UFRGS, Porto Alegre.

Han, J. , Kamber, M.(2001). ***Data Mining: Concepts and Techiniques***.1.ed. New York: Morgan Kaufmann, 2001.

Hearst, M. A. (1993).***TextTiling: A quantitative approach to discourse segmentation***. Technical Report Sequoia 93/24, Computer Science Division, University of California, Berkeley.

Hearst, M. A.(1997).***TextTiling: Segmenting text into multi–paragraph subtopic passages***. Computational Linguistics, vol. 23, no. 1 pp. 33–64, 1997. Disponível em: http://ucrel.lancs.ac.uk/acl/J/J97/J97-1003.pdf , acessado em 06 de Maio de 2007.

Hearst, M.A. (1998). ***Automated discovery of wordnet relations.*** In Christiane Fellbaum, editor, WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

Hellmann, M. (2001) .***Fuzzy Logic Introduction.***, Université de Rennes, 2001.

Hersh, W. R.(1995) ***Towards new measures of information retrieval evaluation.*** In: International ACM-SIGIR Conference on Research and Development in Information Retrieval. Proceedings 1995.

Hovy, E. (1988). ***Generating Natural Language under Pragmatic Constraints.*** Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey.

Hovy, E.; Lin, C. (1997). **Automated Text Summarization in SUMMARIST**. In: I. Mani and M. Maybury (1997.) ***Intelligent Scalable Text Summarization*** ACL 1997 Workshop, pp. 39-46. Madrid, Spain.

Hovy, E.; C.Y. Lin; L. Zhou. (2005). ***A BE-based multi-document summarizer with sentence compression.*** Proceedings of Multilingual Summarization Evaluation (ACL 2005), Ann Arbor, MI.

Hutchins, J. (1987). ***Summarization: Some problems and Methods***. In: Jones. Meaning: The frontier of informatics. Cambridge. London, pp. 151-173.

Keogh, E.; Kasetty, S.(2002). ***On the need for time series data mining benchmarks: a survey and empirical demonstration***, In: ACM SIGKDD, Edmonton, Canada, 2002, p.102-111.

Korfhage, R. R.(1997). **Information storage and retrieval**. EUA: John Wiley & Sons, 1997.

Kowalski, G. (1997).***Information Retrieval Systems: Theory and Implementation*** Boston: Kluwer Academic, 1997. 282 p.

Larocca, J. N.1, Santos,A. D. S, Kaestner ,C. A.A. e Freitas A. A. (2000). ***Generating Text Summaries through the Relative Importance of Topics***. Lecture Notes in Computer Science Springer Berlin / Heidelberg Volume 1952/2000, ISSN0302-9743 (Print) 1611-3349 (Online) pp 300, 2000, Brazil.

Leung, H., Chung, K. F.-L., and fai Chan, S. C. (2003). ***A new sequential mining***
approach to XML document similarity computation. pages 356–362.

Loh, S.(1999). ***Descoberta de Conhecimento em Textos***. Universidade Federal do Rio Grande do Sul-Instituto de Informática-Curso de Pós-graduação em Ciência da Computação.Exame de Qualificação- EQ-29 CPGCC-UFRGS, 1999.

Loh, S(2001). ***Abordagem Baseada em Conceitos para Descoberta de Conhecimento em Textos*** Universidade Federal do Rio Grande do Sul-

Instituto de Informática-Curso de Pós-graduação em Ciência da Computação.Tese de Doutorado- UFRGS, 2001.

Loh, S.; Wives, L. K.; Amaral, L.A.;Oliveira J.P.M.(2006)***Descoberta de Conhecimento em Textos através da Análise de Seqüências Temporais.*** Workshop em Algoritmos e Aplicações de Mineração de Dados, WAAMD, II; SBBD, 2006, Florianópolis, ISBN 85-7669-088-8. Florianópolis: Sociedade Brasileira de Computação, 2006. p. 49-56.

Lopes, M.C.S. (2004). ***Mineração de dados textuais utilizando técnicas de clustering para o idioma português*** Tese de Doutorado COPPE/UFRJ em outubro de 2004-Rio de Janeiro.

Luhn, H. P. (1958). ***The automatic creation of literature abstracts.***.IBM Journal of Research and Development, 2, pp. 159-165.

Marcu, D. (1997a). ***From Discourse Structures to Text Summaries.*** In I. Mani and M. Maybury (eds.), Proc. of the Intelligent Scalable Text Summarization Workshop, pp. 82-88. ACL/EACL'97 Joint Conference. Madrid, Spain.

Mani, I.; Maybury, M.T. (1999). ***Advances in automatic text summarization.*** *MIT Press*, Cambridge, MA.

Maybury, M. (1993). Automated Event Summarization Techniques. In: ***Seminar Report of Summarizing Text for Intelligent Communication Seminar.*** Dagstuhl, Germany.

Mitchell, T. M. (1997). ***Machine Learning***. McGraw-Hill Series in Computer Science.

Mitra, S. ; Acharya, T.(2003). ***Data Mining: multimedia, soft computing, and bioinformatics***. Wiley-Interscience, 2003.

Mittal,V. O.,Kantrowitz,M.,Goldstein,J.,Carbonell,J. G.(1999) ***Selecting Text Spans For Document Summaries: Heuristics And Metrics*** ,In *Aaai/Iaai* (1999), Pp. 467-473.

Morita, M.; Shinoda, Y.(1994). ***Information filtering based on user behavior analysis and best match text retrieval.*** In: CROFT, W. Bruce; RIJSBERGEN, C. J. van (eds). VII International ACM-SIGIR Conference on Research and Development in Information Retrieval. Proceedings... London: Springer-Verlag. 1994.

Oliveira, H. M.(1996). ***Seleção de entes complexos usando lógica difusa.*** Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, PUC-RS, Porto Alegre

Paice, C.D.; Jones, P.A. (1993). ***The identification of important concepts in highly structure technical papers.*** In R. Korfaghe, E. Rasmussen, and P. Willett (eds.), Proc. of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 69- 78. ACM Press, June.

Pardo, T.A.S.; Espina, A.P.; Rino, L.H.M.; Martins, C.B.;(2001). ***Introdução à Sumarização Automática.*** Tech. Report RT-DC 002/2001, Departamento de Computação, Universidade Federal de São Carlos. Abril. 38p.

Pardo, T.A.S. (2002). *DMSumm: Um Gerador Automático de Sumários*. Dissertação de Mestrado. Departamento de Computação. Universidade Federal de São Carlos. São Carlos- SP.

Pardo, T.A.S. (2002). *GistSumm: um sumarizador automático baseado na idéia principal de textos*. São Carlos- SP: USP/ICMC, 2002. Série de Relatórios do NILC-TR-02-13.25p.

Pardo, T.A.S.; Rino, L.H.M.; Martins (2003). ***TeMário: Um Corpus para Sumarização Automática de Textos.*** Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional, NILC - ICMC-USP, Outubro de 2003.

Pardo, T.A.S.; Rino, L.H.M.; Martins (2006) ***A Coleção TeMário e a Avaliação de Sumarização Automática. S***érie de Relatórios do Núcleo Interinstitucional de Lingüística Computacional, NILC - ICMC-USP, Janeiro de 2006.

Passarin D.(2005) ***Text Mining no Aperfeiçoamento de Consultas e Definição de Contextos de uma Central de Notícias Baseada em RSS*** Trabalho de conclusão de curso. Universidade Luterana de Plamas-Tocantis.

Pollock, J.J.; Zamora, A. (1975). ***Automatic Abstracting Research at Chemical Abstracts Service.*** Journal of Chemical Information and Compute Sciences 15(4): 226-232.

Pottenger, W. M.; Yang, T.(2001). ***Detecting emerging concepts in textual data mining***, in: Michael Berry (ed.), Computational Information Retrieval, SIAM, Philadelphia, August 2001.

Riloff, E.; Wiebe, J.; Phillips, W. (2005). ***Exploiting Subjectivity Classification to Improve Information Extraction***, Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05) .

Rizzi, C.; Wives, L. K.; Engel, P. M.; Oliveira, J. P. M.(2000). ***Fazendo Uso da Categorização de Textos em Atividades Empresariais***. In: International Symposium on Knowledge Management/Document Management (ISKM/DM 2000), III, Nov, 2000.

Salton, G.(1983). ***Introduction to Modern Information Retrieval***. New York: McGraw-Hill, 1983.

Salton, G.; Buckley, C. (1988). ***Term-weighting Approaches in Automatic Text Retrieval***. Information Processing and Management 24, pp. 513-523.

Silva, C.M.; Vidigal, M.C.; Vidigal Filho, P.S.; Scapim, C.A.; Daros,E.; Silvério, L.(2005) ***Genetic diversity among sugarcane clones (Saccharum spp.).*** Acta Scientiarum. Agronomy, v.27, p.315-319, 2005.

Saracevic, T.(1997) ***Relevance: a review of and a framework for the thinking on the notion in information science.*** In: Sparck J. K; Willet, P,1997.

Sparck J. K; Willet, P.(1997). ***Readings in Information Retrieval.*** San Francisco: Morgan Kaufmann, 1997.

Sparck J. K. (1999). ***Automatic Summarizing: factors and directions***. In I. Mani and M. Maybury (eds.), Advances in automatic text summarization, The MIT Press, pp. 1-12.

Strube L.L. (2008). ***Abordagem não supervisionada para Extração de Conceitos a partir de Textos*** In: Tecnologia da Informação e da Linguagem Humana-TIL, 2008, Vila Velha. Todas as Palavras da Sentença como Métrica para um Sumarizador Automático. Vila Velha : WebMedia, 2008. p. 359-363.

Sugeno, M.; Yasukawa, T.(1993) ***A fuzzy-logic-based approach to qualitative modeling.*** IEEE Transactions on Fuzzy Systems, v.1, n.1, Fevereiro de 1993.

Sutton, R. S.; Barto A. G.(1998). ***Reforcement learning: an Introduction.*** Cambridge: MIT Press, 1998.

Tan, A. H. (1999). ***Text mining: the state of the art and the challenges.*** In: WORKSHOP ON KNOWLEDGE DISCOVERY FROM ADVANCED DATABASES, 1999. Proceedings... Heidelberg, 1999. p.65-70. (Lecture Notes in Computer Science, 1574).

Tsaur, S.; Chang, T. e Yen, C. (2002). ***The evaluation of airline service quality bu fuzzy MCDM***. Tourism Management, n. 23, p. 107-115, 2002. Disponível em: <http://mslab.hau.ac.kr/mgyoon/master_02/ahp8.pdf>.
Acessado em: 23 junho de 2007.

Velickov, S.(2004). ***TextMiner theoretical background.***
Disponível em: <http://www.delft-cluster.nl/textminer/theory/>. Acesso em: 10 Set. 2007.

Vianna, D. S. (2004). ***Heurísticas híbridas para o problema da logenia.*** Tese de doutorado, Pontifícia Universidade Católica - PUC, Rio de Janeiro, Brasil.

Watkins, C. J. H.(1989). ***Learning from delayed rewards.*** 1989. Thesis. (PhD in thesis) - University of Cambridge.

Watkins, C. J.C.H.; Dayan, P. (1992).***Q-learning. Machine Learning***, v. 8, n. p. 279-292, 1992.

Wives, L.K.(1999).***Um estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de "Clustering".*** Porto Alegre: UFRGS, 1999. Dissertação (Mestrado em Ciência da Computação), Instituto de Informática, Universidade Federal do Rio Grande do Sul, 1999.

Wives, L. K.; Rodrigues, N. A.(2000). ***Eurekha***. Revista Eletrônica da Escola de Administração da UFRGS (READ), Porto Alegre, v.6, n.5, 2000.

Wives, L.K.(2002). ***Tecnologias de Descoberta de Conhecimento em Textos Aplicadas à Inteligência Competitiva*** Universidade Federal do Rio Grande do Sul-Instituto de Informática-Curso de Pós-graduação em Ciência da Computação.Exame de Qualificação- EQ-069 PPGC-UFRGS, 2002.

Wives, L.K.(2004).***Utilizando conceitos como descritores de textos para o processo de identificação de conglomerados (clustering) de documentos*** – Tese (doutorado) – Universidade Federal do Rio Grande do Sul.Programa de Pós-graduação em Computação, Porto Alegre, BR – RS,2004 p.136

Zadeh, L. A. (1965).***Fuzzy Sets***. Information and Control, [S.l.], v.8, p. 338-353, 1965.

Zadeh, L. A. (1973).***Outline of a new approach to the analysis of complex systems anddecision processes.*** IEEE Transactions on Systems, Man and Cybernetics, LosAlamitos, v.SMC-3, n.1, p. 28-44, 1973.

ANEXO

ANEXO 01

Resultados Publicados da Pesquisa

- Guelpeli, M.V.C.; Garcia A.C.B.(2007).***Automatic Summarizer Based on Pragmatic Profiles,*** International Conference WWW/Internet 2007- IADIS- Volume II pág. 149-153- ISBN: 978-972-8924-44-7 - mês de Outubro de 2007- Vila Real-Portugal .
- Guelpeli, M.V.C.; Garcia A.C.B.(2007). ***An Analysis of Constructed Categories for Textual Classification Using Fuzzy Similarity and Agglomerative Hierarchical Methods***, proceedings of The 2007 IEEE International Conference on SIGNAL-IMAGE TECHNOLOGY and INTERNET- BASED SYSTEMS. Shanghai, China 16 a 19 de December, 2007.
- Guelpeli, M.V.C.; Garcia A.C.B. Bernadini, F. C.(2008). ***An Analysis of Constructed Categories for Textual Classification using Fuzzy Similarity and Agglomerative Hierarchical Methods.*** Extend an invitation to you to publish a chapter in the upcoming book on "Emergent Web Intelligence" published by Springer Verlag in the series Studies in Computational Intelligence.
- Guelpeli, M. V. C.; Bernardini, F. C. ; Garcia, A. C. B. . ***Todas as Palavras da Sentença como Métrica para um Sumarizador Automático.*** In: Tecnologia da Informação e da Linguagem Humana- TIL, 2008, Vila Velha. Todas as Palavras da Sentença como Métrica para um Sumarizador Automático. Vila Velha : WebMedia, 2008. p. 287-291.