Universidad de Salamanca

Departamento de Informática y Automática

Técnicas de Mineração Incrementais em Recuperação de Informação

Tese de Doutoramento

D. José Filipe Ribeiro de Figueiredo Lopes

Director:

Dr. D. José Luis Alonso Berrocal

Fevereiro 2010

VNiVERSiDAD
D SALAMANCA

UNIVERSIDAD DE SALAMANCA
Departamento de Informática y Automática

Técnicas de Mineração Incrementais em Recuperação de Informação

Tese de doutoramento apresentada por:
D. José Filipe Ribeiro de Figueiredo Lopes

Dirigida por:
Dr. D. Jose Luis Alonso Berrocal

O doutorando

Salamanca, Fevereiro de 2010

Jose Luis Alonso Berrocal, *Profesor Titular de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca*

HACE CONSTAR: *Que* D. José Filipe Ribeiro de Figueiredo Lopes, *Licenciado in 'Informática de Gestão' por la Universidad Católica Portuguesa (Portugal) ha realizado bajo mi dirección la Memoria que lleva por título* 'Técnicas de Mineração Incrementais em Recuperação de Informação', *con el fin de obtener el grado de Doctor por la Universidad de Salamanca.*

Y para que surta los efectos oportunos firmo en Salamanca, a veinticuatro de Febrero de dos mil diez.

Agradecimentos

Eu poderia escrever uma nova dissertação apenas com agradecimentos a todos que me ajudaram. Acho que todos nós aprendemos a viver e a ser quem somos através de conhecimentos e emoções provenientes de cada um que conhecemos na nossa vida. Assim, só tenho a agradecer a todos que conviveram e convivem comigo por serem óptimas pessoas, tendo contribuído para eu chegar onde cheguei e ser quem sou.

Para atingir este grande objectivo que eu vim buscar a Espanha, começo por agradecer ao meu orientador e incentivador, Dr. D. José Luís Alonso Berrocal. Foram dois anos de muita aprendizagem, trabalho e amizade. Uma pessoa que me ensinou muito não apenas na área académica e técnica, mas também na área humana, com verdadeiras lições de vida. Muito obrigado por tudo e espero que esta amizade continue.

Obrigado também ao meu colega Armando Carlos Costa Carvalho pela companhia nas viagens para Salamanca, pela troca de ideias e pela sua amizade.

Por último, agradeço à minha família. Ao meu irmão Carmindo António Ribeiro de Figueiredo Lopes, sempre companheiro e incentivador, e aos meus pais, Maria Clara Alexandre Ribeiro Lopes e Carmindo Figueiredo Lopes. Sem o apoio, compreensão e amor que vocês têm por mim, eu nunca chegaria ao final deste objectivo. Muito obrigado.

Resumo

Uma das propriedades desejadas nos algoritmos de aprendizagem automática é a capacidade de aprender incrementalmente. Este tipo de algoritmos tem recebido atenção por parte da comunidade cientifíca nos últimos anos. As redes Bayesianas, impõem uma dificuldade acrescida a esta tarefa, pois um exemplo pode alterar toda a sua estrutura. Neste tese foca-se a incrementalidade para o algoritmo Tree Augmented Naive Bayes (TAN). A versão incremental deste algoritmo poupa recursos, tempo e é mais adequada para áreas como o data mining ou ambientes dinâmicos. Mas como é usual nas redes Bayesianas, este algoritmo está limitado a lidar com atributos discretos. Deste modo também faz parte desta tese o estudo de um método incremental. Uma vez que estamos a lidar com uma rede Bayesiana incremental o mais natural é a discretização também o ser. Deste modo é possível avaliar o algoritmo incremental com conjuntos de dados que contêm atributos contínuos. A discretização é um pré-processamento fundamental para alguns algoritmos bem conhecidos, mas a discretização incremental tem recebido muito pouca atenção por parte da comunidade cientifíca.

Esta tese apresenta duas grandes contribuições para a área de aprendizagem incremental, a primeira é um TAN incremental, a segunda um método de discretização incremental. Apresenta-se e testa-se a versão incremental do TAN, esta reconstrói a estrutura da rede baseada na soma pesada das informações mútuas. Apresenta-se ainda o método de discretização incremental. O método é composto por duas camadas, cada uma das quais contém uma discretização. Esta arquitectura é bastante flexível. O método pode ser aplicado de modo supervisionado ou não supervisionado, dependendo da opção tomada para a segunda camada. Na segunda camada é possível aplicar a maior parte dos métodos de discretização. O aspecto mais relevante deste método é que os limites dos

intervalos da segunda camada podem mudar quando o algoritmo tem acesso a mais conjuntos de exemplos. Testou-se experimentalmente a discretização incremental com métodos supervisionados e não supervisionados. O método foi ainda testado com algoritmos incrementais e não incrementais.

A avaliação experimental da versão incremental do TAN indica que o seu desempenho é similar ao do TAN não incremental. O mesmo resultado se aplica ao teste da discretização incremental. Esta, apesar da pouca atenção por parte da comunidade cientifica, é fundamental para a avaliação de algoritmos incrementais. A avaliação de algoritmos incrementais que usem conjuntos de dados com atributos contínuos torna-se mais precisa e realista. Mas o método tem outras vantagens, pois pode ser aplicado a grandes conjuntos de dados, onde um método de discretização não incremental tem dificuldades em ser aplicado. Pode ainda ser usado em ambientes dinâmicos, área em que a discretização não incremental é inadequada.

Abstract

A desirable property of learning algorithms is the ability of incorporating new data in an incremental way. Incremental algorithms have received attention on the last few years. Particulary Bayesian networks, this is due to the hardness of the task. In Bayesian networks one example can change the whole structure of the Bayesian network. In this theses we focus on incremental induction of Tree Augmented Naive Bayes (TAN) algorithm. A incremental version of TAN saves computing time, is more suite to data mining and concept drift. But, as usual in Bayesian learning TAN is restricted to discrete attributes. Complementary to the incremental TAN, we propose an incremental discretization algorithm, necessary to evaluate TAN in domains with continuous attribute. Discretization is a fundamental pre-processing step for some well- known algorithms, the topic of incremental discretization has received few attention from the community.

This theses has two major contributions, the benefict of both proposals is incremental learning, one for TAN and the other for discretization. We present and test a algorithm that rebuilds the network structure of tree augmented naive Bayes (TAN) based on the weighted sum of vectors containing the mutual information. We also present a new discretization method, this works in two layers. This two-stage architecture is very flexible. It can be used as supervised or unsupervised. For the second layer any base discretization method can be used: equal width, equal frequency, recursive entropy discretization, chi-merge, etc. The most relevant aspect is that the boundaries of the intervals of the second layer can change when new data is available. We tested experimentally the incremental approach to discretization with batch and incremental learners.

The experimental evaluation of incremental TAN shows a perfor-

mance similar to the batch version. Similar remarks apply to incremental discretization. This is a relevant aspect, because few works in machine learning address the fundamental aspect of incremental discretization. We believe that with Incremental discretization, the evaluation of the incremental algorithms can become more realistic and accurate. We evaluated two versions of incremental discretization: supervised and unsupervised. We have seen that this feature can improve accuracy for the incremental learners and that the preview of future algorithm performance can be more precise. This method of discretization has another advantages, like, can be used with large data set's or can be used in dynamic environments with concept drift, areas where a batch discretization can be difficult or is not adequate.

Indice

Indice de Figuras

Capítulo 1

Introdução

Actualmente vivemos num mundo competitivo em que os crescentes avanços da tecnologia e da comunicação vêm exercendo uma forte influência sobre a sociedade. De entre os meios de comunicação desenvolvidos, destaca-se, como o de maior amplitude e importância global, a Internet. Com a Internet rompem-se paradigmas e intensificam-se relações sociais, fazendo surgir um novo cenário onde a informação é o elemento chave. Desde o seu surgimento, a sua função é permitir o acesso e a partilha de informações. No entanto, no decorrer do tempo, por apenas se preocupar em absorver e suportar todo o volume de informação criado tornou-se um grande espaço emaranhado de nós interconectados desprovidos de qualquer organização. Somos testemunhas do enorme aumento de informações e recursos na Web...

A literatura da Ciência da Informação é concebida com divergências no que diz respeito ao surgimento dessa ciência. Existem duas linhas que sustentam diferentes pontos de vista históricos para o surgimento da Ciência da Informação, a primeira atribui aos estudos de Paul Otlet e Henri La Fontaine, referentes à bibliografia e à documentação, os elementos predecessores da Ciência da Informação; e a segunda atribui o aparecimento dessa ciência ao surgimento de tecnologias para o tratamento da informação científica e tecnológica no período Pós Segunda Guerra Mundial. Esta segunda traz as tecnologias de informação como preocupação capital para o período Pós Guerra.

Traçando uma evolução histórica do desenvolvimento da Ciência da Informação nos Estados Unidos, faz sentido a ideia apresentada nesta

segunda linha de pensamento afirmando que o primeiro período marcante da evolução da Ciência da Informação, compreendido no período de 1948 à 1964, foi marcado pela necessidade de investir em tecnologias que visassem a automatização da informação e unidades de informação.

O Institute of Information Scientists (IIS) formou-se no Reino Unido com o intuito de diferenciar o cientista da informação do cientista de laboratório. Os membros do IIS eram cientistas de diversas áreas e estavam preocupados com o estudo da informação e dos processos envolvidos na comunicação científica. Lidavam, portanto, com o problema de organização, incremento e disseminação do conhecimento registado, gerado antes da Segunda Guerra Mundial. A história da Ciência da Informação pode ser resumida pelos principais conceitos utilizados para lidar com o tratamento da informação.

A primeira ideia original que emerge nos anos 1950 é a recuperação da informação para o processamento da informação baseada na lógica formal; a segunda, que surge pouco tempo depois, é a relevância, orientando e associando o processo com as necessidades de informação dos utilizadores e avaliação, e a terceira, que chegou aproximadamente duas décadas depois, é a interacção, que permite intercâmbio directo e retroalimentação entre sistemas e os utilizadores ligados ao processo de recuperação da informação.

A recuperação da informação é assunto essencial na sociedade moderna, submersa numa infinidade de informações. Com a rápida evolução das tecnologias de informação e comunicação, a produção e a divulgação do conhecimento em forma de informação tornaram-se mais numerosas e velozes, dificultando a recuperação eficiente das informações.

A actividade científica tem como principais objectivos a comunicação da informação e a disseminação dos conhecimentos produzidos, visando o próprio desenvolvimento e progresso da ciência. Pode-se, inclusive, afirmar que a ciência se desenvolveu largamente em torno dos documentos científicos. Ao longo da história e no mundo actual, pode-se perceber que praticamente nada mudou nesse sentido. O valor das publicações é mantido para a ciência, como o permite perceber a declaração clássica de Ziman (1979), quando afirma que 'ciência é conhecimento público'. O que realmente mudou e vem mudando são os suportes das publicações científicas. Se antes eram as cartas, actas escritas à mão, hoje são fotocópias, faxes, e-mails e tantos outros formatos que ainda expressam os

mesmos objectivos daquelas primeiras comunidades, ou seja, unir grupos de pessoas com os mesmos interesses e disseminar a informação entre os seus pares.

A utilização da Internet, mais especificamente da Web, no mundo cresceu muito na última década. O seu potencial é imenso, só o facto de disponibilizar acesso à informação em diversos lugares e promover a comunicação entre as pessoas é algo de grande utilidade e, só por estas características, o facto já justificaria a sua utilização. Com a popularização da Internet e, consequentemente, o aumento de seu número de utilizadores, houve também um grande aumento de recursos disponíveis através da rede, sejam eles serviços ou informações. Desta maneira, tal avanço fez com que a Internet passasse e ser encarada como um grande sistema aberto distribuído e ao alcance e disposição de toda a população mundial.

Mais de um bilião de páginas são indexadas pelos motores de busca [Pal, 2000] e achar a informação desejada pode algumas vezes tornar-se uma tarefa difícil. Essa abundância de informações e recursos instigou a necessidade do desenvolvimento de ferramentas automáticas de mineração e descoberta de informações na Web.

Os Sistemas Abertos caracterizam-se principalmente pela capacidade de conectar redes a outras redes, tornando documentos, dados e software acessíveis remotamente por pessoas e outros sistemas, agentes e ferramentas. O surgimento desta tecnologia acarretou uma forte mudança de paradigma na relação homem-máquina, trazida pela popularização da Internet. Os computadores pessoais, que, há aproximadamente 25 anos eram destinados apenas a tarefas dentro de um ambiente de informações restrito, controlado e estático, transformaram-se em janelas para um mundo continuamente renovável de informações, pessoas e software. Com isso, a antiga metáfora da manipulação directa da informação - que se sustentava devido à pequena quantidade de informação manipulada - começou a entrar em declínio.

Os Sistemas de Recuperação de Informação objectivam a realização das tarefas de indexação, busca e classificação de documentos (expressos na forma textual), a fim de satisfazer a necessidade de informação do indivíduo, geralmente expressa através de consultas. A necessidade de informação pode ser entendida como a busca de respostas para determinadas questões a serem resolvidas, a recuperação de documentos que

tratam de determinado assunto ou ainda o relacionamento entre assuntos.

Hoje em dia, a localização de documentos através de engenhos de busca, é feita, geralmente, com a utilização de buscas por palavras chave ou expressões contidas nos documentos. O sucesso em encontrar documentos relevantes depende do casamento dos termos fornecidos pelo utilizador numa consulta, com os utilizados como índices na indexação da base de dados de documentos.

Com o crescimento das colecções de documentos digitais, os sistemas de recuperação de informação que localizam documentos utilizando buscas por palavras chave e expressões simples têm-se tornado cada vez menos eficazes. Este insucesso está relacionado com os seguintes motivos: a dificuldade do utilizador em expressar o que ele realmente procura através de uma consulta; a forma desorganizada como os documentos resultantes da busca são mostrados; o número excessivo de documentos devolvidos.

Com a vasta quantidade e variedade de documentos disponíveis, formular uma consulta eficiente para uma busca é uma tarefa difícil, e examinar uma lista resultante de uma pesquisa, onde os itens são muitos e estão ordenados de forma claramente não significativa, pode ser fastidiosa. Assim, tornam-se necessários métodos que sejam capazes de realizar uma organização automática dos documentos em conjuntos, evidenciando o relacionamento entre os conteúdos desses documentos, e as relações de proximidade entre os conjuntos de documentos de forma visual. Esta organização facilitará a navegação e a pesquisa sobre a colecção de documentos.

A partir desse momento, as atenções voltaram-se para o desenvolvimento de tecnologias visando a busca e recuperação eficiente das informações, visto que as dificuldades de encontrar informações relevantes num meio e num espaço não estruturado são grandes.

1.1 Factores que favoreceram o crescimento da Internet

A Internet pode ser definida como uma vasta e omnipresente rede global. Não é à toa que é chamada de 'a rede das redes', interligando

vários computadores em todo o mundo, alcançando níveis de abrangência e utilização jamais imaginados pelos seus criadores.

Desde a sua criação (meados da década de 1970), tinha como propósito permitir a comunicação, a troca de informações e a partilha de computadores e outros recursos. O seu público-alvo consistia em cientistas, investigadores e militares, que a utilizavam para fins educativos e militares.

O processo de transição da Internet destacou-se a partir do final da década de 1980 quando os EUA libertaram a rede para uso comercial. No entanto, somente no início da década de 1990, com o surgimento das primeiras empresas fornecedoras de acesso comercial, e, principalmente, com o surgimento da Web, houve, de facto a explosão popular da Internet. A partir dessa época até os dias actuais, a Internet passou a integrar-se, progressivamente, nos diversos segmentos da sociedade, contribuindo para o desenvolvimento de um novo tecido social, caracterizado por mudanças de hábitos e comportamentos.

A expansão da Internet, desencadeada pelo aumento do número de utilizadores conectados, tornou-se um fenómeno amparado pelo desenvolvimento tecnológico levando em conta dois pressupostos:

1. Novas tecnologias que possibilitaram a interconexão de hosts a partir de equipamentos de telecomunicações e processamento cada vez mais rápidos;

2. Softwares de comunicação fáceis de serem utilizados que permitiram o acesso e partilha de informações na rede, bem como facilitaram a interacção entre os 'internautas' (denominação dada àquele que utiliza a Internet).

Este contexto favorável contribuiu para que o uso da Internet evoluísse, continuamente, a ponto de tornar-se um hábito nas suas vidas. Consequentemente, novos documentos foram produzidos e distribuídos no meio social através da Web, criando assim uma cultura humana de produção.

1.2 A Importância da Exploração de Dados

A Web é hoje a maior fonte de informação electrónica que dispomos. Entretanto, por causa da sua natureza dinâmica, a tarefa de encontrar informações relevantes torna-se muitas vezes uma experiência frustrante. Muitos esforços de pesquisa têm sido feitos no sentido de remediar tal problema. Um deles é a utilização de técnicas de exploração de dados para a descoberta de informações na Web.

De forma geral, a exploração de dados em rede (Web) pode ser descrita como a descoberta e análise inteligente de informações úteis [Cooley, 1997]. Pode-se estar interessado, por exemplo, na informação contida dentro dos documentos da Web - mineração de conteúdo - na informação contida entre os documentos da Web - mineração de estrutura - ou na informação contida na utilização ou interacção com a Web - mineração de uso.

Estas são as três categorias em que se divide a exploração de dados na Web, de acordo com a parte da Web a ser explorada.

Para cada classificação são desenvolvidas técnicas e metodologias distintas, muitas delas herdadas de outras áreas disciplinares como Aprendizagem de Máquina, Bases de Dados, Estatística, Recuperação de informação, Inteligência Artificial e Redes Sociais.

O recurso de exploração de dados na Web não é uma ferramenta recente, pois vem sendo citada e estudada desde meados do ano de 1996, mas tem realmente crescido em importância nestes últimos anos. Aponta-se dois factores principais que contribuíram para tal facto:

- Considerável aumento da quantidade de transacções comerciais na Web, que motivaram o desenvolvimento de técnicas para a mineração de uso, pois através delas os sites de venda puderam aprender acerca dos perfis dos compradores para montarem melhores estratégias de venda e marketing;

- Crescente desenvolvimento da Web semântica [Decker, 2000] e, consequentemente, da tecnologia dos agentes de informação [Sycara, 1996], onde as técnicas de mineração na Web são utilizadas. A Web semântica poderá, entre outras coisas, ampliar a inteligência dos agentes e não apenas o seu conhecimento.

Dessa forma, os serviços da Web poderão, eles próprios, tornarem-se entidades dotadas de comportamento autónomo, que poderão, entre outras coisas, comunicar entre si através de uma linguagem comum. Os recursos de exploração na Web serão ferramentas cruciais a serem utilizadas pelos agentes e serviços nessa visão da Web, tendo em vista que os ajudarão em tarefas diversificadas, de entre as quais estão busca por informações, personalização e talvez até como mecanismo de aprendizagem.

Entretanto, existem muitos desafios e problemas que devem ser contornados antes que a Web possa realmente transformar-se num meio mais rico, amigável e inteligente na qual todos possamos explorar e partilhar. Percebe-se que a inexistência de regras e padrões rígidos, bem como a dinâmica, a informação e a horizontalidade da rede a favoreceram face à participação social no acesso e produção de informações.

Através do apoio e do desenvolvimento tecnológico, as características da Internet foram fortalecidas, garantindo a sua rápida expansão e tornando-a um espaço extremamente complexo, dotado de infinitas ramificações, contendo informação em diferentes formatos e lugares.

1.3 Aprendizagem Automática

Actualmente existe uma proposta que diz que a inteligência é a utilização habilidosa do conhecimento. Se se seguir esta linha de pensamento um programa é inteligente se conseguir reter conhecimento e o usar correctamente.

Desde que os primeiros computadores começaram a ser projectados e construídos, na década de 1950, têm surgido projectos ambiciosos na tentativa de transformar um computador numa máquina que consiga processar cada vez mais informações e mais rapidamente. Muitos destes projectos foram-se transformando com o tempo até que surgiu uma nova ambição, a denominada 'Inteligência Artificial'.

A Inteligência Artificial pode ser definida como um conjunto de técnicas e metodologias de programação utilizadas na tentativa de resolução dos problemas, de forma mais eficiente do que a utilização de soluções algorítmicas [Coe, 1995]. Assim, o primordial objectivo da Inteligência

Artificial é conseguir que um computador (genericamente pode ser substituído por uma máquina) consiga processar informações da mesma forma que um ser humano. A tentativa de compreensão e imitação do processo de aprendizagem dos seres humanos funcionou desta forma como motivação para este campo da informática. Actualmente, existem já algumas aplicações da Inteligência Artificial que permitem a resolução de problemas que eram considerados impossíveis de serem resolvidos por uma máquina.

Um dos objectivos da aprendizagem automática é a obtenção de algoritmos que melhorem o seu desempenho através da experiência. Mais recentemente surgiram grandes bases de dados que se tornaram habituais em muitas áreas, tais como na ciência e negócios. Desta forma, surgiram novas aplicações e motivações neste campo.

No momento, a aprendizagem automática parece dividida em duas áreas: Reconhecimento de padrões e 'Data Mining'. O principal objectivo da primeira é a classificação de exemplos com base em padrões encontrados nos dados, enquanto que o objectivo da segunda é encontrar modelos matemáticos nos dados.

Alguns autores argumentam que os métodos heurísticos e simbólicos se fundem quando associados à inteligência artificial, enquanto que outros estão preocupados em associar os métodos numéricos com a aprendizagem estatística. Além destas duas visões existe ainda uma terceira que afirma que todos os métodos referidos são apenas pontos de vista diferentes de uma mesma ciência e que a aprendizagem automática tem como objectivo alcançar algoritmos que melhoram com a experiência.

Nos últimos anos, muitas aplicações desenvolvidas na área da aprendizagem automática tiveram sucesso. Estas quando alinhadas na área do data mining, permitem detectar operações fraudulentas dos cartões de crédito e conduzir um automóvel de forma autónoma numa auto-estrada pública. Simultaneamente ocorreram importantes avanços nos algoritmos que são a base deste campo [Mit, 1997]. Assim, as potencialidades deste campo são enormes. Imagine-se, por exemplo, as consequências que traria o facto de um programa poder aprender com a experiência: um computador poderia aprender a partir de registos médicos quais os tratamentos mais efectivos para novas doenças, as casas inteligentes aprenderiam a optimizar o custo de energia baseado no comportamento das pessoas que nela habitassem ou o software poderia aprender os gostos

e comportamento do utilizador para, automaticamente, evidenciar uma notícia que possa potencialmente ser do interesse deste.

Uma correcta compreensão da forma de tornar possível que os computadores aprendam, pode tornar possível o aparecimento de novas utilidades e novos níveis de competências para estes. Contudo ainda não se conhece a forma de tornar possível que um computador aprenda como um humano. No entanto, existem já algoritmos que são eficientes para aprender alguns tipos de tarefas específicos. A compreensão teórica da aprendizagem é assim um campo emergente.

1.4 Classificação

A classificação supervisionada é parte integrante da inteligência artificial denominada por reconhecimento de padrões, a qual se aplica a diversos campos científicos e tecnológicos [Múg, 2002].

O problema central da classificação centra-se na obtenção automática de um modelo que consiga atribuir uma classe a um caso que é composto por um vector de atributos. Estes modelos são normalmente obtidos através de um conjunto de teste constituído por exemplos classificados, ou seja, para os quais é conhecido a classe a que pertencem. Ao longo das últimas décadas, na tentativa de resolução deste problema, a estatística e a inteligência artificial começaram a propor diferentes paradigmas classificatórios nomeadamente árvores de classificação [Qui, 1993], redes neuronais, análise do discriminante linear e máquinas de suporte vectorial. Assim, na literatura existem diversas tentativas de comparação dos diferentes paradigmas de classificação, como por exemplo Michie [Mst, 1994], Heckerman [Hec, 1997b] e Lim [Lls, 2000].

A decisão de escolher o paradigma adequado ou ideal (caso se possa utilizar este termo) para fazer face ao problema é difícil e está normalmente condicionada aos conhecimentos dos investigadores ou técnicos. No entanto, existem algumas formas e critérios para avaliar e comparar o desempenho dos diferentes paradigmas dos classificadores, dos quais se deve salientar a taxa de erro, a complexidade algorítmica, a simplicidade de interpretação dos resultados e a simplicidade do modelo obtido.

1.5 Motivação

Apesar de existirem diversos algoritmos de classificação, o naive Bayes [Dhs, 2001] é considerado pela comunidade científica, como competitivo com algoritmos bem mais sofisticados [Múg, 2002] [Friedman & Goldszmidt, 1996] [Hec, 1997a] [Rou, 2002], e também computacionalmente mais complexos.

O naive Bayes é amplamente utilizado em tarefas de classificação, não só devido à sua simplicidade mas também devido ao facto de ser um algoritmo incremental. No entanto, o TAN supera o naive Bayes sem acrescentar uma complexidade proibitiva, mas perdendo a característica de algoritmo incremental. Desta forma acredita-se que o TAN poderá ser uma alternativa mais viável se for dotado desta característica. De notar que em alguns casos da literatura, por exemplo [Rou, 2004], são utilizadas conjuntos de dados com atributos contínuos. De forma a ser possível testar o algoritmo neste tipo de conjuntos de dados os autores efectuavam uma discretização prévia. De salientar que com o teste dos algoritmos se pretende analisar o seu desempenho, para que seja possível deduzi-lo no futuro e numa situação real.

No entanto, numa situação real o algoritmo não terá acesso a todos os dados previamente, o que indica que os resultados obtidos podem ser enviesados. Este facto ocorre porque se a discretização for efectuada à medida que o algoritmo tem acesso aos dados a discretização resultante é diferente, logo, os seus resultados também. Contudo, os respectivos autores apenas pretendem testar o algoritmo, ignorando este aspecto. Assim, uma das motivações é a correcta avaliação do algoritmo. Deste modo torna-se relevante o desenvolvimento de um método de discretização incremental que possibilita não só uma aproximação do desempenho do algoritmo em 'condições reais', mas também o melhoramento da discretização encontrada inicialmente. De salientar, que os trabalhos sobre discretização incremental são escassos, o que se acredita ser uma grande lacuna na área, apesar de este ser um tópico em foco nos últimos anos.

Por conseguinte os objectivos principais, desta tese, são três:

- desenvolver a característica de incrementalidade para o TAN;
- desenvolver um método de discretização incremental;
- melhorar o método de avaliação de algoritmos incrementais.

1.6 Contribuições

Assim, como contribuições desta tese, temos:

- Estudo dos vários tipos de TAN desenvolvidos pela comunidade cientifica;
- Desenvolvimento e estudo de um classificador TAN incremental;
- Estudo da área da discretização, nomeadamente a sua aplicação a métodos incrementais;
- Desenvolvimento de um método de discretização incremental;
- Desenvolvimento de um método de discretização que acelera a discretização em grandes conjuntos de dados;
- Desenvolvimento e teste de um classificador Bayesiano como algoritmo incremental com discretização incremental.

1.7 Organização da tese

A tese encontra-se organizada do seguinte modo:

No capítulo 2 é feita uma abordagem à mineração na Web. Neste capítulo é também focada a Recuperação de Informação e alguns dos seus pressupostos.

No capítulo 3 é analisam-se alguns resultados de pesquisa, exploração e extracção de dados na Web. São referidos conceitos como Inteligência Artificial e Inteligência Artificial Distribuída.

No capítulo 4 são referidas algumas técnicas de mineração de dados que tornam possível criar perfis de clientes de determinadas organizações. São abordados classificadores e algoritmos de classificação e de retropropagação.

No capítulo 5 é abordado o modelo de integração de motores de indexação que permite o desenvolvimento de soluções que possibilitem a realização de recuperação de informações num ambiente de fontes de dados heterogéneas.

No capítulo 6 descrevem-se as motivações para o desenvolvimento de algoritmos incrementais. De seguida, efectua-se a revisão bibliográfica sobre o TAN e uma revisão dos métodos de discretização, sendo introduzida a definição e os conceitos. É descrita a proposta de um novo algoritmo, o TANi (uma versão do TAN incremental) e são apresentados vários resultados de testes efectuados ao novo algoritmo, nomeadamente, a comparação de árvores de dados e a análise da taxa de acerto. É descrita a motivação para o desenvolvimento de um método de discretização incremental, seguida de uma descrição de uma nova metodologia. Além disso, são apresentados os resultados obtidos com o novo método proposto.

No capítulo 7 apresentam-se as conclusões da tese, resumem-se os pontos chave da tese e discutem-se alguns pontos que se pensa serem importantes para futuro desenvolvimento.

Capítulo 2

Mineração na Web

2.1 A Web

A Web é uma vasta colecção de documentos heterogéneos. Possui natureza dinâmica e milhões de páginas surgem e desaparecem todos os dias. Por isso sente-se um anseio para que a Web realmente alcance todo o seu potencial e se torne uma ferramenta mais utilizável, eficaz e compreensível. Nesse contexto a mineração de dados aparece como uma possibilidade óbvia a ser explorada. Em parte pelo seu grande sucesso quando aplicada a bases de dados tradicionais, e em parte porque a Web parece ser uma área fértil em potencial para a aplicação das suas técnicas.

A mineração de dados refere-se ao processo não trivial de identificação de padrões válidos, previamente desconhecidos e potencialmente úteis dos dados [Frawley, 1992]. Entretanto, utilizar e compreender os dados disponíveis na Web não é uma tarefa simples, pois esses dados são muito mais sofisticados e dinâmicos do que os sistemas de armazenamento de bases de dados tradicionais. Enquanto estes últimos utilizam estruturas de armazenamento bem definidas e estruturadas, a Web não possui qualquer controlo sobre a estrutura ou o tipo dos documentos que virtualmente armazena. Outro aspecto que diferencia a mineração de dados tradicional da mineração na Web é a existência de vínculos de hipertexto entre os seus documentos. Os vínculos de hipertexto são uma rica fonte de informações a ser explorada, pois entre outras coisas, ajudam no processo de ranking de páginas pelos motores de busca e na identificação de micro-comunidades na Web.

Apesar das diferenças e particularidades entre as duas abordagens - mineração em dados tradicionais e mineração de dados da Web -, a metodologia utilizada para a mineração na Web segue os mesmos passos utilizados no processo geral de descoberta de conhecimento em bases de dados (KDD - Knowledge Database Discovery). O processo de mineração na Web é dividido em quatro sub-tarefas [Etizione, 1996], que na verdade são idênticas às fases do processo KDD. Com base nas quatro fases descritas a seguir e na representação da Figura 2.1, a mineração na Web pode ser vista como a utilização de técnicas de mineração de dados para a recuperação automática, extracção e avaliação de informação para a descoberta de conhecimento em documentos e serviços da Web. Aqui, avaliação inclui tanto 'generalização' quanto 'análise' [Pal, 2000].

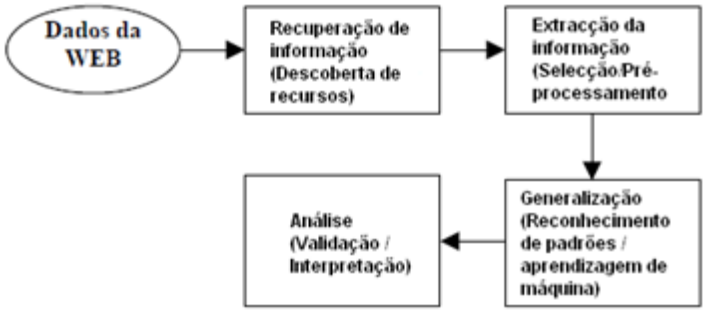

Figura: 2.1: Sub-tarefas da mineração na Web

2.2 Recuperação de Informação (descoberta de recursos)

A recuperação de informação ou descoberta de recursos trata da automatização do processo de recuperação de documentos relevantes, que inclui, principalmente, representação, indexação e busca de documentos.

Um índice é basicamente uma colecção de termos retirados dos documentos com ponteiros para os lugares onde as informações sobre os documentos podem ser encontradas [Pal, 2000]. A indexação de páginas Web, para facilitar o processo de recuperação, é bem mais complexa que o processo de indexação utilizado em bases de dados tradicionais. A enorme quantidade de páginas na Web, o seu dinamismo e as suas

actualizações frequentes fazem da indexação uma tarefa aparentemente impossível. E, na verdade, esse é um dos grandes desafios dos serviços de busca actuais: indexar toda a Web. Os serviços de busca - programas destinados a consultar e recuperar informações armazenadas tanto em bases de dados, páginas HTML ou texto - estão ainda bem longe disso e isso tem influência na recuperação das informações desejadas, pois, algumas vezes, os utilizadores estão atrás de uma informação que está justamente no segmento da Web que ainda não foi indexado.

A indexação de documentos na Web pode ser humana, manual ou automática [Pal, 2000], e está baseada nos modelos tradicionais de recuperação de informação: espaço vectorial, estatístico e linguístico [Baeza-Yates & Ribeiro-Neto, 1999] [Girardi, 1998] [Salton, 1983].

A indexação é essencialmente um processo de classificação onde é realizada uma análise conceptual do documento ou elemento de informação. Por exemplo, nas técnicas baseadas no modelo do espaço vectorial, a indexação envolve a atribuição de elementos de informação a certas classes, onde uma classe é o conjunto de todos os elementos de informação para o qual um termo de indexação (ou palavra-chave), em particular, tem sido atribuído.

Os elementos de informação podem fazer parte de várias classes. Algumas técnicas atribuem pesos aos termos de indexação de um elemento de informação de forma a reflectir a sua relativa relevância [Girardi, 1998]. Nas técnicas baseadas no modelo estatístico os termos de indexação são extraídos a partir de uma análise de frequência das palavras ou frases em cada documento e em toda a fonte de informação. Nas técnicas linguísticas, os termos de indexação são extraídos utilizando técnicas de processamento da linguagem natural, por exemplo, análise morfológica, lexical, sintáctica e semântica [Girardi, 1995].

2.3 Selecção da Informação / Extracção e Pré-Processamento

Uma vez tendo sido os documentos recuperados, o próximo passo é transformar ou pré-processar estes documentos de forma que os algoritmos de mineração de dados e aprendizagem de máquina possam ser aplicados de forma efectiva.

O campo disciplinar conhecido como extracção de informação presta um grande serviço à mineração da Web, no que diz respeito à fase de extracção e pré-processamento da informação. Denomina-se extracção de informação à tarefa de identificar fragmentos específicos que constituem o núcleo semântico de um documento em particular e construir modelos de representação da informação (conhecimento) a partir dele [Palm, 2002]. Os métodos geralmente envolvem a escrita de código específico, popularmente chamados de wrappers responsáveis pelo mapeamento do documento para algum modelo de representação do conhecimento. O problema é que para cada documento da Web temos que escrever um código específico, tornando o trabalho manual. Como os documentos da Web não possuem uma semântica agregada às informações que contêm, e nem mesmo um padrão de como apresentar essas informações ao utilizador, temos que aprender acerca da estrutura individual de cada documento e escrever código para essa estrutura em particular. Daí a dificuldade de estendermos ou generalizarmos o mesmo código para outros documentos.

Vários métodos foram desenvolvidos para a extracção de informação tanto em documentos desestruturados quanto em semi-estruturados. [Kushmerick, 1997], por exemplo, descreve vários aspectos e técnicas da extracção de informação, [Freitag, 1998] fala sobre a aplicação de algoritmos de aprendizagem de máquina para a extracção de informação de documentos HTML e [Soderland, 1999] fala sobre a aprendizagem de regras para a extracção de informação de documentos semi-estruturados e texto comum. É importante salientar a diferença entre as fases de recuperação e extracção de informação.

As técnicas de extracção de informação procuram derivar conhecimento de documentos recuperados segundo a forma como um documento está estruturado e representado, enquanto as técnicas de recuperação de informação visualizam o documento apenas como um conjunto de palavras [Palm, 2002].

2.4 Generalização

Após as informações terem sido extraídas e algum modelo de representação das informações ter sido construído, são utilizadas técnicas de mineração de dados e aprendizagem de máquina para descobrir novo co-

nhecimento a partir do que já existe. É nessa fase que os algoritmos de mineração vão descobrir novo conhecimento em cima do que já existe. Um exemplo que nos daria uma ideia de como seria uma saída de um desses algoritmos é dado abaixo.

a) 70% das pessoas que acedem a secção sobre natação também acedem a secção sobre artes marciais; - Neste exemplo, a saída poderia dar uma indicação ao responsável pela manutenção da loja virtual sobre as preferências e perfis dos seus clientes, de forma a montar estratégias de vendas que possam induzir o utilizador a comprar mais.

b) 80% dos sites que abordam o tema Fórmula 1 possuem links apontando para sites que falam da vida de Ayrton Senna; - Neste outro exemplo, a saída descobre uma relação interessante entre os sites podendo dar novos caminhos aos utilizadores interessados nestes tópicos.

O maior problema em aprender ou descobrir novos conhecimentos da Web é a falta de marcação semântica das informações. Muitos algoritmos de mineração de dados requerem como entrada exemplos positivos ou negativos de algum conceito. Se, por exemplo, tivéssemos um conjunto de páginas na Web marcadas como exemplos positivos e negativos do conceito portal, seria fácil modelar um algoritmo classificativo para a classificação automática de novas páginas como portais ou não portais. Embora a Web actual dificulte o processamento das suas informações por parte das máquinas, a Web Semântica [Lee, 2001] fornece uma solução para este problema.

Agrupamento ou clustering é uma técnica de classificação que não requer entradas com marcação semântica, e por isso tem sido aplicada com sucesso em grandes conjuntos de documentos HTML [Cutting, 1992]. No clustering, documentos são agrupados de acordo com a sua similaridade, portanto, um novo documento é classificado de acordo com a sua similaridade com algum conjunto de documentos existente. Uma boa referência para clustering no contexto da mineração na Web pode ser encontrada em [Lingras, 2002].

As Regras de associação também podem ser utilizadas nessa fase do processo. Regras de associação são basicamente expressões do tipo X >= Y onde X e Y são conjuntos de itens. X >= Y demonstra que sempre que uma transação T contenha X então ela provavelmente também irá conter Y. A probabilidade ou confiança da regra é a percentagem de transacções contendo Y junto a X comparado com o total de transacções contendo

X. A ideia de utilizar mineração em regras de associação tem origem nos dados de super-mercados e afins, onde regras como 'O cliente que compra o produto x também comprará o produto y com probabilidade (confiança) de c%' [Pal, 2000].

Na sua concepção a Web foi construída de forma a atender as necessidades de visualização e consumo dos seres humanos, onde os textos são quase sempre escritos em linguagem natural sem nenhuma semântica que facilite o seu processamento. Isso instigou o desenvolvimento de um novo conceito para a Web, chamada de 'Web semântica' onde além de outras coisas promete escrever os documentos da Web com uma semântica agregada às informações de maneira a que as máquinas possam compreendê-los e processá-los.

2.5 Análise

Uma vez tendo sido descobertos os padrões, os analistas precisam de técnicas e ferramentas apropriadas de modo a entender, visualizar, interpretar e validar esses padrões. O sistema WEB-MINER [Mobasher, 1997], por exemplo, propõe uma linguagem de consulta estruturada para a consulta do conhecimento descoberto (na forma de regras de associação e padrões sequenciais). Outros sistemas utilizam técnicas de OLAP [Han, 2000] com o propósito de simplificar a análise de estatísticas de uso em logs de acesso.

2.6 Categorias da Mineração na Web

Nesta secção é apresentada uma visão geral das categorias em que se divide a mineração na Web, assim como algumas das técnicas utilizadas em cada uma delas. A mineração na Web divide-se em três categorias de acordo com a parte da Web a ser analisada: mineração de conteúdo, mineração de estrutura e mineração de uso.

A mineração de conteúdo aborda a mineração dos dados contidos dentro dos documentos da Web. A grande quantidade de formatos que os dados podem assumir (textos comuns, páginas HTML, imagens, áudio, vídeo, etc.) acabam por dirigir as técnicas de mineração a ser utilizadas.

A mineração de estrutura, por outro lado, aborda a mineração das informações contidas entre os documentos da Web. Os documentos da Web relacionam-se basicamente através de vínculos de hipertexto, e esses vínculos escondem informações valiosas e interessantes não só sobre a topologia da Web, mas também sobre como os documentos se relacionam.

A mineração de uso, por sua vez, aborda a mineração das informações de uso da Web, que por outras palavras, são as informações sobre como o utilizador interage com a Web. Nessa categoria são tratadas questões como personalização, interfaces adaptativas e aprendizagem de perfis de utilizadores.

2.6.1 Mineração de Conteúdo

A mineração de conteúdo trata da descoberta de informações úteis do conteúdo, dados, documentos e serviços da Web [Pal, 2000]. Convém salientar que o conteúdo da Web não se constitui apenas de texto ou hipertexto, mas abrange uma ampla variação de tipos de dados, tais como áudio, vídeo, dados simbólicos, metadados e vínculos de hipertexto. Apesar de já existir uma área de pesquisa destinada ao estudo da mineração de dados multimédia, o foco ainda são os dados de texto e hipertexto, que na verdade são os que constituem o grosso da Web. Uma boa referência para pesquisa sobre mineração de dados multimédia é descrita por Zaiane [Zaiane, 1998].

Os dados de texto da Web podem ser de três tipos: desestruturados, tais como textos comuns, semiestruturados, tais como documentos HTML, e estruturados, tais como as tabelas de bases de dados. No tratamento de dados desestruturados utiliza-se KDT (Knowledge Discovery in Texts) ou mineração de dados em textos. A mineração em textos é uma área bem amadurecida e a sua cobertura em detalhe está para além do propósito deste trabalho, mas uma boa referência é descrita por Mladenic [Mladenic, 1998].

A extracção de conhecimento da Web e a sua modelagem numa representação simbólica para a aplicação de técnicas de mineração de dados é descrito por Ghani [Ghani, 2000]. Algumas outras abordagens que tratam da mineração de dados em texto sugerem reestruturar os documentos de forma que eles se tornem legíveis para as máquinas, ou seja, técnicas para a inserção de marcas (tags) semânticas nas informações

[Pal, 2000].

A mineração em hipertexto envolve a mineração de páginas HTML, as quais além de texto contêm vínculos em hipertexto. Um excelente tutorial apresentando esse assunto é descrito por Chakrabarti [Chakrabarti, 2000].

A mineração em serviços da Web tais como grupos de notícia, grupos de e-mail, lista de discussão e bibliotecas digitais também é uma área que tem cada vez mais chamado a atenção dos pesquisadores, principalmente pesquisadores envolvidos na área de WI ('Web Intelligence') [Zhong, 2002]. A WI promete, entre outras coisas, transformar os serviços da Web em entidades inteligentes, de forma que elas possam interagir e comunicar entre si através de uma linguagem comum, elevando assim, a Web a um outro nível de tecnologia da informação. Em [Levy, 2000] são discutidos os sistemas de Internet inteligentes em geral, abordando temas como modelagem de utilizadores, descoberta e análise em fontes de informações remotas, integração da informação e gestão de sites da Web.

Há uma linha ténue que separa a mineração de conteúdo e a recuperação de informação na Web. Não há um consenso sobre a relação entre as duas, alguns afirmam que a recuperação da informação na Web pode ser vista como uma instância da mineração de conteúdo, e outros associam a mineração de conteúdo com recuperação inteligente de informação. Isso acontece porque algumas vezes as duas acabam por trabalhar juntas para alcançar determinado objectivo e uma acaba por complementar a outra.

Há basicamente duas estratégias para a mineração de conteúdo: uma realiza a mineração directamente do conteúdo dos documentos e a outra incrementa o poder de busca de outras ferramentas e serviços. Na primeira estratégia, os documentos pretendidos já foram recuperados e já estão prontos a ser analisados (mineração). Na segunda estratégia, a mineração de conteúdo presta um grande 'favor' às ferramentas e serviços de recuperação de informação, pois ajuda a realizar o processo de indexação e categorização dos documentos. Desta forma, percebe-se que quando a mineração de conteúdo utiliza a segunda estratégia, ela complementa o processo de recuperação de informação, sendo utilizada como uma ferramenta pelos motores e serviços de busca, daí nesse caso ser descrito por alguns como recuperação inteligente de informação.

A mineração de conteúdo pode seguir duas abordagens: baseada em

agentes ou baseada em bases de dados. A abordagem baseada em agentes envolve o desenvolvimento de sistemas de inteligência artificial que podem agir de forma autónoma ou semi-autónoma para a descoberta e organização de informações da Web de acordo com os interesses de um utilizador em particular [Kosala, 2000]. Geralmente, a abordagem baseada em agentes pode ser dividida em três categorias: agentes de busca inteligentes, agentes de filtragem e/ou categorização da informação e agentes de interface [Cooley, 1997]. A abordagem de bases de dados focaliza-se nas técnicas para transformar os dados semi-estruturados ou desestruturados da Web em modelos de dados estruturados onde mecanismos de consulta, como, por exemplo, a linguagem SQL, possam ser utilizados, assim como técnicas de mineração de dados para a análise.

2.6.2 Mineração de Estrutura

Enquanto que na mineração de conteúdo da Web estamos interessados no que há dentro dos documentos, na mineração de estrutura o interesse está nas informações que existem de forma implícita entre os documentos.

Esta categoria envolve a mineração da estrutura que há por detrás da interligação entre os documentos da Web. O que liga esses documentos são os vínculos de hipertexto, os quais são os principais objectos de estudo nesta categoria. A Web pode ser visualizada como um grafo orientado, onde os nós representam páginas, e as setas entre pares de nós representam vínculos entre as páginas. Essa representação da Web em forma de grafo apresenta uma forte semelhança com as chamadas redes sociais [Kumar, 2002] que, juntamente com a análise de citações, inspirou a pesquisa dessa categoria de mineração.

A teoria moderna de redes sociais foi desenvolvida a partir do trabalho de Stanley Milgram [Kumar, 2002]. Em 1967, Milgram conduziu experiências onde ele pedia que diversas pessoas residentes em Omaha, Nebraska, conduzissem uma carta para um associado seu que morava em Boston. As pessoas só podiam enviar a carta para outra pessoa que elas conhecessem pelo primeiro nome, e essas pessoas por sua vez só podiam retransmitir a carta para uma pessoa que elas também conhecessem pelo primeiro nome. O objectivo era de que a carta chegasse ao seu associado no menor número de 'passos' possíveis. Milgram descobriu que o número médio de 'passos' ao longo do caminho das cartas que conseguiam chegar

com sucesso era seis, criando a ideia de que quaisquer duas pessoas residentes nos Estados Unidos estavam ligadas numa rede social com 'seis graus de separação'.

Os pesquisadores têm explorado continuamente as similaridades entre a Web e as redes sociais, desenvolvendo técnicas que incrementam o poder dos motores de busca e dos sistemas de gestão do conhecimento. Nas citações bibliográficas quando um artigo é bastante citado isso indica que provavelmente este é um artigo importante e de maior autoridade perante outros que abordam o mesmo tema. Acontece o mesmo com as páginas e documentos da Web. Os vínculos de hipertexto dão indicações interessantes de como as páginas se relacionam entre si, links apontando para uma página, por exemplo, podem indicar a sua importância, enquanto links 'saindo' de uma página podem indicar, entre outras coisas, a continuação ou complemento dos tópicos por ela abordados.

Alguns algoritmos foram propostos para a modelagem da topologia da Web tais como o HITS ('Hyperlinked Induced Topic Search') [Kleinberg, 1998] e o PageRank [Brin, 1998]. Esses modelos são aplicados principalmente para calcular a qualidade ou relevância das páginas da Web. Uma das regras utilizadas é que quanto mais páginas estiverem apontando para uma determinada página, mais relevante ela será. Várias medidas são tomadas para garantir que as páginas que apontam tenham credibilidade. Alguns exemplos são o sistema Clever [Chakrabarti, 1999] e o motor de pesquisa Google [Brin, 1998]. Algumas outras aplicações destes modelos são a categorização de páginas Web e a descoberta de micro-comunidades na Web [Kumar, 1999].

2.6.3 Mineração de Uso

A mineração de uso da Web focaliza-se em técnicas que possam prever o comportamento do utilizador enquanto ele interage com a Web [Kosala, 2000]. Enquanto a mineração de conteúdo e a mineração de estrutura utilizam os dados reais ou primários da Web, a mineração de uso lida com os dados secundários provenientes da interacção do utilizador com a Web. Os dados de uso da Web incluem dados provenientes de logs de servidores web, logs de servidores proxy, logs de browsers, perfis de utilizador, cookies, secções ou transacções de utilizadores, pasta 'favoritos', consultas do utilizador, clicks de rato e qualquer outro dado gerado pela interacção do utilizador com a Web.

O processo de mineração de uso da Web pode ser classificado segundo duas abordagens [Borges, 1998]. A primeira mapeia os dados de uso do servidor Web em tabelas relacionais antes das técnicas adaptadas de mineração de dados serem aplicadas. A segunda utiliza os dados de logs directamente, utilizando técnicas especiais de pré-processamento. Assim como no KDD, a limpeza e pré-processamento dos dados, aqui, é uma parte crucial do processo, pois a qualidade desses dados vai determinar a eficiência dos algoritmos de mineração.

Uma boa referência para a descrição e comparação de métodos de pré-processamento para a mineração de uso da Web pode ser encontrada em [Cooley, 1999]. As aplicações da mineração de uso da Web podem ser classificadas em duas categorias principais: aprendizagem do perfil de utilizador ou modelagem em interfaces adaptativas (personalização) e aprendizagem de padrões de navegação de utilizador. A mineração de uso da Web despertou interesse especial no comércio electrónico, principalmente pela sua necessidade de aprender acerca do comportamento dos clientes, perfis de compra, preferências e padrões de navegação.

Alguns sites populares de comércio electrónico já utilizam estas técnicas não só para a adaptação do site de acordo com o perfil do utilizador, mas também para fazer recomendações de produtos de acordo com compras anteriores, ou baseadas na similaridade entre perfis de utilizadores.

2.7 Problemas de Recuperação de Informação na Internet

O termo 'sobrecarga de informação' (do inglês 'information overload') refere-se a uma enorme quantidade de documentos disponíveis que colocam ao utilizador a difícil tarefa de separar o trigo do joio na busca de informação útil. Com o intuito de minimizar este problema, os mecanismos de busca - como, por exemplo, o Excite, Yahoo!, AltaVista e outros - foram então projectados com base em técnicas desenvolvidas pela área de Recuperação de Informação (RI) [Baeza-Yates & Ribeiro-Neto, 1999]. Eles indexam as páginas da Internet por palavras-chave, e usam métodos e estruturas de dados para recuperá-las rapidamente, devolvendo ao utilizador uma lista de endereços de páginas que contém as palavras solicitadas, ordenadas por frequência destas palavras. Dada a variedade de conteúdo da informação disponível esta era a alternativa

viável, uma vez que os dois principais pilares que dão suporte à existência da Internet, o protocolo HTTP (HyperText Transfer Protocol) e a linguagem HTML (HyperText Markup Language) foram projectados tendo como principal intuito assegurar a apresentação e a navegação na rede. Preocupações sobre como capturar conhecimento específico, ou seja, realizar buscas semânticas, não foram prioritárias. Devido a este facto, os mecanismos de busca caracterizam-se por uma alta cobertura, porém uma significativa falta de precisão, muitas vezes entregando ao utilizador uma grande quantidade de endereços de páginas inúteis ou irrelevantes. Utilizando algoritmos matemáticos para atribuir relevância às páginas, estes mecanismos não conseguem dotar de semântica a busca, porque possuem capacidade de representar as páginas com análises baseadas apenas no nível léxico. Alguns dos problemas que decorrem deste facto estão listados abaixo:

* O utilizador mediano não conhece as linguagens de consulta dos mecanismos de busca, confundindo-se com problemas triviais como o uso de letras maiúsculas, múltiplas palavras-chave, lógica booleana (o emprego do E e do OU), e a possibilidade de modificação de consultas já efectuadas [Baeza-Yates & Ribeiro-Neto, 1999].

* Para achar o que procura, o utilizador deve escolher as palavras mais apropriadas para encontrar a informação desejada, e, mesmo procedendo desta forma, não há garantias de que a informação esteja na lista de páginas retornadas pelos mecanismos de busca como resultado das consultas [Leong, 1996].

* O utilizador perde tempo e paciência no 'trabalho pesado' de colher a informação de que precisa na lista de endereços devolvida, muitas vezes tendo que procurar nas páginas apontadas pelas páginas da lista, ou iterativamente ir refinando a sua consulta com um conjunto de palavras-chave mais apropriado.

* Se o utilizador mediano não conhece a lógica de indexação dos mecanismos de busca, o mesmo não ocorre com os responsáveis por projectos de páginas e sítios: alguns deles mascaram a relevância das páginas que projectam, incluindo um número alto de repetições de palavras-chave muito comuns, para se colocar artificialmente no topo das listas dos resultados dos mecanismos de busca. Este facto configura o 'problema de persuasão dos mecanismos de busca' ou 'Web spamming'.

* Diante de uma lista de resultados muitas vezes bastante hetero-

génea, o utilizador vê-se tentado a dispersar, sendo vítima do chamado 'fenómeno do museu de arte'

* Os mecanismos de busca não conseguem resolver alguns problemas ligados à semântica inerentes aos idiomas, especialmente a polissemia (uma palavra com vários significados).

* A maneira com que os mecanismos de busca adicionam sinónimos como palavras-chave associadas ao termo procurado, apesar de melhorar a precisão, não proporciona uma real busca por conteúdo, semanticamente definida, no sentido que o utilizador deseja.

* A interface genérica e simples oferecida pelos mecanismos de busca muitas vezes não permite que os utilizadores recuperem a informação procurada [Steele, 2001] com a precisão que desejam.

Basicamente, duas características da Internet dificultam o acesso à informação útil, específica e relevante: o volume e a falta de estrutura (e, consequente, falta de semântica) das informações. Por isso, torna-se difícil agregar valor à informação disponível, ou seja, transformá-la em informação útil e facilmente acessível, convertendo informação desestruturada ou semi-estruturada em estruturada, e permitindo ainda processos de inferência sobre a informação capturada.

A próxima geração da Web, a chamada rede semântica (Semantic Web, já mencionada aqui) visa, justamente, preencher a lacuna semântica deixada por HTML, de forma a prover às páginas mecanismos para definir conceitos, atributos, relações, e outras facilidades. Entretanto, ainda estão a ser estabelecidas ferramentas-padrão para a definição semântica das páginas, como o RDF (Resource Description Framework, ou ambiente de descrição de recursos), a XML (eXtensible Markup Language, HTML extensível), e a padronização de conceitos para serem instanciados pelas páginas está a ser ainda discutida. Por isso, o uso destas linguagens ainda não se popularizou em páginas da Internet.

Tendo em conta que uma representação semântica exequível da rede ainda está em andamento e não atingirá todas as páginas da Web - pois isso implicaria resolver os problemas de Raciocínio de Senso Comum e de Processamento de Linguagem Natural (PLN), dois dos maiores desafios ainda não resolvidos na área de Inteligência Artificial - a presença de contexto torna-se necessária na indexação. A negligência deste facto já produziu consequências prejudiciais, como a desastrosa tentativa do

conselho de pesquisa americano em traduzir russo automaticamente nos anos 60, sem ter em conta o contexto, que provocou corte em fundos de pesquisa para o PLN.

Contexto pode ser definido como o conjunto de factores relacionados a um texto que o faça ser compreendido adequadamente, incluindo factos considerados verdadeiros e formas de inferência aplicáveis. Visto sob a óptica de RI, contexto não inclui um significado semântico das páginas, mas uma visão mais ligada à categorização delas em colecções ou agrupamentos, em termos de similaridade de palavras-chave, frases, metadados (como autor, data, tamanho, etc) e estrutura de ponteiros. Para a área, contextos assim definidos desempenham um importante papel na construção de interfaces de visualização das páginas, que, em resposta a consultas solicitadas por utilizadores, mostram graficamente a frequência de cada palavra-chave das páginas da lista devolvida, ou o relacionamento dessas páginas com outras páginas ou com conjuntos de páginas, em função das palavras-chave que elas contêm.

Porém, o processo de indexação perde muita informação contextual essencial à compreensão das páginas [Baeza-Yates & Ribeiro-Neto, 1999]. Contexto para a Web poderia ainda ser definido de uma forma muito próxima a uma rede semântica, como o conjunto de entidades e os seus atributos, relações e restrições presentes numa página. Todavia, esta definição de contextos também não poderia abarcar toda a rede: a Web reúne factos e dados sobre assuntos quotidianos e científicos em contextos muito diferentes.

2.7.1 A Necessidade de Restrição de Domínios

Na realidade, talvez o foco do problema não esteja nas técnicas empregadas, mas sim no objectivo: será que é exequível algum conjunto de ferramentas conseguir ter a capacidade de 'compreensão' mínima de todas as páginas da Web, mesmo que o objectivo seja apenas recuperação?

Uma situação semelhante já preocupou cientistas de Inteligência Artificial nos anos 70, assim que surgiram os primeiros sistemas de representação de conhecimento. Após a demonstração de que estes sistemas poderiam conter conhecimento aplicável, junto com os seus respectivos mecanismos de manipulação e inferência sobre esse conhecimento, acreditou-se que fazer sistemas baseados em dedução de grande porte

e, inclusive, de Senso Comum, seria uma simples questão de formalizar mais conhecimento, o que resultou em categóricos fracassos. Gerou-se uma expectativa muito grande, especialmente com o projecto GPS (General Problem Solver, ou, em português, Corrector Geral de Problemas) [Russel & Norvig, 1995], que se propunha a ser um 'acumulador' de conhecimentos que poderiam ser utilizados durante processos de inferência na resolução de problemas variados. Achava-se que tudo o que pudesse ser logicamente definido poderia ser deduzido por um motor de inferência.

À época destes primeiros experimentos, os estudos sobre complexidade de problemas, e especialmente a classificação desses problemas em NP-completo, NP-difícil e outras classes, ainda não estava formalizada, e sistemas com algumas dúzias de factos simplesmente não conseguiam alcançar os seus objectivos, por subestimar a explosão de combinações causada pelo crescimento do número de factos, entre outros motivos. A frustração das expectativas contribuiu inclusive para formar uma imagem negativa da área como um todo, que só veio a ser retratada com a correcta aplicação das lições aprendidas com estes contratempos. Após isto, os sistemas baseados em conhecimento passaram a ser desenvolvidos:

- Reduzindo o domínio de conhecimento para que o número de regras pudesse ser tratável e,
- Codificando cuidadosamente os aspectos mais importantes do domínio com o emprego de regras para diminuir e acelerar a busca.

Postos estes limites, percebeu-se que eram de grande valia os sistemas com inferências sobre domínios delimitados para resolver problemas complexos, de difícil tratamento por sistemas convencionais, de onde se originaram os sistemas especialistas.

Naturalmente a ideia de domínios restritos aplica-se também aos sistemas que se propõem a tratar a gama de informações contidas na Web, pelo simples facto da rede ter sido originada a partir do senso comum. Nenhum sistema solitário conseguirá atender genericamente às necessidades de informação dos utilizadores, enquanto que, trabalhando dentro de um domínio restrito, por exemplo, a polissemia, ou seja, uma palavra com vários significados, pode ser tratada com maior precisão.

Mesmo os investigadores de RI possuem uma intuição da necessidade de restrição de domínio, sem, no entanto, o divulgarem explicitamente: muitos testes dessa área de pesquisa são realizados sobre 'corpos' homogéneos, cujos textos tratam sobre um mesmo assunto, e provêm, muitas vezes, de uma mesma fonte, e não sobre conjuntos de textos tão variados em conteúdo e estilo como os disponíveis na Web. Além do mais, recentemente uma nova tendência em mecanismos de busca começa a difundir-se, os mecanismos de busca especializados [Steele, 2001], que actuam sobre áreas específicas, como notícias ou sobre sítios específicos, que fornece preços e disponibilidade de livros e CDs de diversas lojas, entre outros serviços.

2.7.2 Extracção de Informações da Internet

À parte da diversidade encontrada na Web, há porções ou regiões dela mais tratáveis e menos gerais, que podem ser investigadas como um corpus de relativa homogeneidade, com o intuito de procurar dados específicos e objectivos. Para reforçar essa ideia, ressalte-se, ainda, o facto de que uma boa quantidade de utilizadores acede à Internet com objectivos claros e específicos, mais interessados em informações relevantes, úteis, focalizadas e agregáveis do que nas páginas em que estão hospedadas. Além do mais, páginas que tratam de um mesmo tópico (cinema, classificados, e tantas outras) costumam apresentar regularidade de formatação, estrutura e principalmente de conteúdo.

Partindo também deste conjunto de hipóteses, surgiu uma nova subárea de Recuperação de Informações conhecida como Extracção de Informações (EI), cujo tema principal é a reorganização e reutilização de regiões da Web [Atzeni, 1997] em bases de dados. A construção de extractores de informação da Internet oferece vantagens: o utilizador é mais bem atendido, livrando-se de processar manualmente as páginas atrás de dados, e, por isso, a rede fica com menor tráfego já que muitos ponteiros inúteis não serão listados nem carregados. Hoje, já existem várias aplicações com o objectivo de construir bases de dados a partir de páginas bem estruturadas na Internet. Bases de dados, diferentemente da Web, podem ser facilmente consultadas, fornecendo ao utilizador consultas semanticamente claras e precisas sobre entidades e relacionamentos entre elas, inclusive combinando e totalizando dados, tarefas que os mecanismos de busca, mesmo os especializados, não conseguem realizar.

Consequências interessantes advêm da existência de extractores. Em primeiro lugar, eles trazem uma noção de memória à Internet. Os mecanismos de pesquisa actuais repetem continuamente o cálculo de relevância das páginas relativas a determinadas consultas, e não há forma de aproveitar trabalho alheio, ou seja, cálculos de consultas anteriores, e nem pesquisas já efectuadas por outros utilizadores. Além do mais, os extractores servem a dois benefícios contextuais [Akman & Surav, 1996]: a implicação contextual, em que uma nova assertiva pode ser usada junto com o conhecimento existente para gerar novas assertivas; e a contradição ou eliminação, em que uma nova assertiva pode modificar ou eliminar algumas das assertivas existentes.

2.7.2.1 Ausência de Extracção Integrada

Os actuais sistemas de extracção actuam sobre páginas de domínios muito restritos, na realidade constituindo sistemas ad hoc, já que visam o processamento de uma classe muito específica de páginas. Esta não é uma abordagem apropriada para a Web, devido a várias razões. A principal delas refere-se à existência das âncoras. Elas contêm elementos indicativos muito importantes, claros e seguros de como as informações da Web estão semanticamente ligadas e não devem ser ignoradas. Esta afirmação também traz à tona interessantes questões sobre uma possível proliferação de extractores: Como integrar as bases de dados, permitindo aos utilizadores combinar as informações? Como aproveitar as informações contidas nas âncoras com o propósito de executar extracção integrada, aproveitando o conteúdo das âncoras? Como devem ser abordadas e vistas as regiões, os domínios e as classes de páginas da Web com esse objectivo?

Uma importante e negligenciada característica a ser explorada das classes de páginas processadas pelos extractores é que elas se inter-relacionam com outras classes, formando conjuntos ou grupos (clusters). Inclusive há informações pertinentes a um conjunto de dados processados por um extractor que podem ser encontradas em páginas processadas por outro. Isto evidencia-se, por exemplo, em páginas de pesquisadores. Nem sempre consta em que eventos científicos eles fizeram parte de comités de programa, entre outras informações ausentes. Assim, um extractor pode ser mais útil se cooperar com outros extractores dentro de um modelo de domínio adequado - não se resumindo a processar informações

apenas de uma classe restrita, como, por exemplo, artigos científicos - montando uma base de dados integrada sobre esse domínio a partir das informações recolhidas pelos extractores.

O obstáculo encoberto que impede extractores de cooperarem uns com os outros reside na representação de conhecimento que se popularizou na área como um todo, com raras excepções, como o sistema Alembic [Villain, 1999]. O uso de métodos simples e rápidos como autómatos finitos e gramáticas, entre outros, dão prioridade à velocidade de processamento e desenvolvimento e à fácil adaptação, aplicando técnicas de aprendizagem automáticas, em detrimento das possibilidades de uso do conhecimento especifico, como os processos de inferência [Villain, 1999] e, por conseguinte, cooperação.

Portanto, para a manipulação integrada de informação, a representação declarativa de factos de um texto, mais de que a simples ocorrência de um padrão, revela-se de fundamental importância para ambos os processos.

2.7.2.2 Recuperação, Categorização e Extracção

Entre as contribuições deste trabalho encontra-se a ideia de que recuperação, categorização e extracção sobre páginas da Web constituem tarefas complementares, já que elas podem ajudar-se mutuamente, conforme explicitado na figura seguinte.

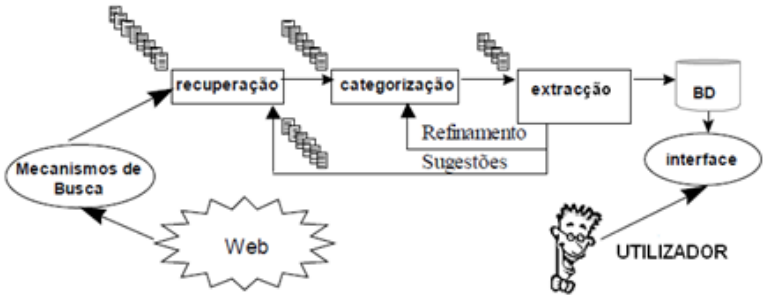

Figura: 2.2: Esboço da arquitectura de um sistema de extracção, evidenciando a complementaridade entre as tarefas de recuperação, categorização e extracção.

Sistemas de recuperação podem fornecer o acesso a um conjunto inicial de páginas que possui alta cobertura e baixa precisão.

Após isto, sistemas de categorização deveriam seleccionar quais as páginas que pertenceriam às classes a serem processadas, e então os extractores poderiam capturar a informação requerida.

Durante o processo de extracção, poderiam ainda ser encontrados, de entre as âncoras das páginas processadas, endereços de outras páginas que pertenceriam a outras classes também processadas.

A extracção poderia ainda refinar a categorização, uma vez que páginas que não contivessem os dados que caracterizam a classe processada seriam desprezadas.

2.7.2.3 Sistemas Multiagentes para Recuperação, Classificação e Extracção Integrada

A falta de mecanismos capazes de captar a semântica do conteúdo das páginas da Web criou uma forte procura de serviços que se ajusta adequadamente à classe de serviços estudada em Inteligência Artificial, e, mais especificamente, aos agentes inteligentes. Os agentes podem caracterizar-se por adaptação ou aprendizagem, apresentando mais robustez às diversas formas como a informação se encontra estruturada.

Por sua vez, a área de Inteligência Artificial já se ressentia dos custosos tempos de resposta que os agentes ou sistemas especialistas consumiam por não beneficiarem da distribuição, como também da dificuldade de especificar conhecimento de áreas distintas num só agente. Minorando estes problemas, a Inteligência Artificial Distribuída ou Sistemas Multiagentes [Alvares & Sichman, 1997] consolidou-se como uma subárea com vida própria, cujos objectos de estudo se centram em comunicação, cooperação e coordenação de agentes em ambientes distribuídos ou não, com objectivos comuns ou não.

Com efeito, já existe tecnologia disponível para a troca de conhecimento declarativo entre agentes, permitindo cooperação entre eles na execução das suas metas comuns ou individuais, alcançando, por conseguinte, distribuição e concorrência.

2.8 A Web

Como já mencionado, a Internet é uma imensa 'Rede de Redes'. No mundo inteiro, centenas de milhares de computadores estão interligados. Às vezes esses computadores pertencem a uma empresa ou universidade e estão interligados com o fim de partilhar recursos, promovendo contactos, informação e armazenamento. Geralmente as redes precisam de partilhar as informações através de distâncias muito grandes. Para isso é preciso ligar os computadores remotos, seja através da rede telefónica ou de alguma outra forma de conexão.

As redes organizam-se de acordo com o tamanho e a complexidade, dependendo do número de computadores envolvidos ou da quantidade de dados que podem ser enviados entre eles. A maioria delas também permite uma forma de transmitir mensagens, denominada electronic mail (ou e-mail), oferecendo a possibilidade de os utilizadores enviarem mensagens através dos seus computadores. A Internet, através da ligação ou conexão, passa tudo isto à frente, interligando milhares de redes menores, e cada vez mais redes e sistemas estão-se associando a ela.

A Internet não pertence a nenhum país ou a qualquer empresa, pois as diferentes partilhas pertencem a diversas organizações, fazendo com que a rede no seu conjunto não pertença a ninguém. A Internet é basicamente auto-regulada em conjunto. Entretanto, algumas regras foram surgindo ao longo dos anos, não de forma coerciva, nem complicada, mas formando um conjunto de princípios com a finalidade de impedir o desperdício dos recursos da própria rede.

Embora a Internet não tenha sido criada com finalidade comercial (mas, sim, exclusivamente para fins de segurança, educação e pesquisa), é cada vez maior a procura e o interesse por acessos comerciais, seja para uso pessoal ou corporativo. Em quase todo o mundo, existem empresas que fornecem acessos comerciais à Internet, denominadas Internet Service Providers.

Dentro do funcionamento dessa imensa rede de comunicação, pode afirmar-se que cada país participante na Internet possui estruturas principais de rede, chamadas backbones, com conectividade através do protocolo TCP/IP - Transmission Control Protocol / Internet Protocol, às quais se interligam centenas ou milhares de outras redes. Os backbones nacionais, por sua vez, são conectados entre si aos backbones de outros

países, compondo, assim, uma enorme rede mundial. Existem, assim, redes não comerciais (compostas por universidades, centros de pesquisa e entidades educacionais) e redes comerciais (mantidas por empresas de telecomunicações e informática, que prestam serviços de conectividade aos seus clientes).

O funcionamento da navegação pela Web ocorre do seguinte modo: cada pedido de HTTP é enviado a um servidor que procura a solicitação e, quando encontra, envia uma resposta ao utilizador. O caminho dos dados começa, portanto, na máquina do 'cibernauta', segue para o fornecedor de acesso que processa o pedido e envia a resposta para o computador que solicitou a informação. Neste caminho, o trabalho do fornecedor é encontrar os dados que o utilizador pede, e o da companhia que oferece serviço de acesso em banda larga é proporcionar o canal para que esse conteúdo possa fluir. Daí resulta a pequena mudança de nome realizada por algumas empresas, que antes eram chamadas de fornecedores de acesso. Na realidade, estas empresas são fornecedores de conteúdo.

Pela sua riqueza, variedade e falta de estrutura do conteúdo, um dos problemas que precedem qualquer solução relativa à Web é a forma de abordá-la ou modelá-la. O tema 'modelagens da Web' costuma estar presente em muitos congressos sobre a Internet, e constitui tema de intensa pesquisa. Basicamente, existem dois tipos de modelos da Web, baseados em abordagens criadas sob o ponto de vista de Bases de Dados:

- Modelos baseados em grafos, onde os ponteiros representam os arcos do grafo;
- Modelos baseados em dados semi-estruturados, em que partes da Web possuem entidades e atributos, cujo esquema não é completamente conhecido ou obedecido, ou seja, são toleradas páginas que não se adaptam completamente ao esquema.

O primeiro modelo adequa-se melhor a problemas de pesquisa, uma vez que pesquisas em grafos são problemas relativamente formalizados. O segundo modelo oferece ferramentas mais acuradas de acesso aos atributos, e costuma ser empregue em tarefas como sumarização e extracção de dados semi-estruturados.

Os modelos baseados em dados semi-estruturados não tratam toda a

Web, mas partes definidas dela, que devem obedecer minimamente a esquemas pré-definidos. A tarefa conhecida como categorização preocupa-se com a identificação de partes ou categorias da Web, e é empregada com esse propósito. A categorização pode ser feita de duas formas:

1. Sobre categorias previamente definidas ou
2. Criando-se categorias de acordo com a semelhança entre as páginas, problema este conhecido como agrupamento (clustering).

Para extracção, a categorização é normalmente efectuada sobre categorias pré-definidas.

Há diversas modalidades de acesso aos serviços oferecidos pela Internet, que são em última instância, modalidades de conexão entre os computadores e utilizadores e uma espécie de 'central' da Internet, chamada Host Internet. Os acessos podem ser classificados em função de vários factores, tais como: recursos de hardware empregues; tipos de software utilizados; forma de conexão física; velocidade de comunicação e custos.

Entende-se por rede de informação qualquer sistema destinado à interligação de computadores ou demais equipamentos de tratamento de dados, por meio electrónico, óptico ou similar, com o objectivo de oferecer, com carácter público ou privado, informações e serviços a utilizadores que liguem os seus equipamentos ao sistema. A estruturação e o funcionamento das redes de informação e a oferta de serviços de conexão e informação são determinados por contrato, no qual é estipulada uma remuneração.

O administrador da Rede e o fornecedor de cada serviço devem ser responsáveis solidariamente pela segurança, integridade e sigilo das informações armazenadas ou em circulação nas bases de dados disponíveis a consulta ou manuseamento pelos utilizadores da rede. A segurança do controlo de acesso deve ser responsabilidade primordial do fornecedor.

2.8.1 Fornecedores

A função de um fornecedor de Internet é caracterizada por diversos factores, de entre os quais a possibilidade de possuir a conexão full time à

rede mundial através de um backbone. Essas conexões são feitas através de circuitos de comunicação ponto a ponto, conhecidas como links.

Geralmente um fornecedor de grande porte faz a ligação com os fornecedores ditos menores, através dos quais os utilizadores de computadores se conectam à Rede Mundial Internet. Para isto é necessário apenas, além do computador, um meio de comunicação entre ele e seu fornecedor, que normalmente é uma linha telefónica ou conexão em banda larga. Tanto o utilizador como o fornecedor devem possuir modems para o acesso, serviço esse que é disponibilizado pelo fornecedor.

Como existem várias denominações e conceitos para o termo fornecedor, entende-se como uma boa sugestão indicar a sua interpretação. Assim, fornecedor de acesso é a instituição que se liga à Internet, partindo de um 'ponto-de-presença' ou outro fornecedor, para obter conectividade IP e transmiti-la a outros indivíduos e instituições, com carácter comercial ou não. O fornecedor de acesso torna possível ao utilizador final a conexão à Internet através de uma ligação telefónica local. Em suma, fornecedor de acesso é aquele que serve obrigatoriamente de elemento de ligação entre o 'cibernauta' receptor e o 'cibernauta' emissor. Não restam dúvidas de que um fornecedor de acesso é também um prestador de serviços técnicos contratado como intermediário entre os utilizadores de Internet.

- Fornecedor de informação é o organismo cuja finalidade principal é recolher, manter ou organizar informações online para acesso através da Internet por parte de assinantes da rede. Essas informações podem ser de acesso público incondicional, caracterizando assim um fornecedor não comercial ou, no outro extremo, constituir um serviço comercial onde existem tarifas ou assinaturas cobradas pelo fornecedor.

- Fornecedor de conteúdo é a organização que tem como finalidade principal recolher, manter e organizar informações online para acesso através da Internet. Estas informações podem ser de acesso ao público, caracterizando assim um fornecedor não comercial, ou constituir um serviço comercial onde existem tarifas ou assinaturas cobradas pelo fornecedor.

- Fornecedor de serviço (Internet Service Provider), objecto principal dos estudos aqui efectuados, engloba tanto o fornecedor de acesso,

> quanto o de informação. É a união do fornecedor de acesso com o de conteúdo. Pode ser definido como aquele que, de maneira habitual e na medida dos seus parâmetros profissionais, recolhe, difunde e transmite dados no âmbito da Internet.

Ainda não há consenso a respeito desta classificação, pois em muitos casos é difícil o enquadramento de um destes organismos apenas num tipo como, por exemplo, quando um fornecedor de informação disponibiliza acesso via linha telefónica aos seus clientes, caracterizando-se também como fornecedor de acesso.

Nos Estados Unidos, o termo ISP é usado de forma geral para denominar o que acima classificamos como fornecedores de acesso, sendo em alguns casos também usado para fornecedores que se aproximam, em conteúdo e finalidade, dos classificados aqui como fornecedores de backbone (acesso a redes locais, através de 'pontos de presença').

Vale a pena lembrar que a compatibilidade da operação depende do perfeito funcionamento de todos os elementos envolvidos, pois a incompatibilidade de qualquer um deles pode afectar a situação dos demais. Assim, se alguém, alguma máquina ou programa utilizado na operação não estiverem totalmente adequados ao sistema, irá ocasionar, com certeza, defeitos na prestação desses serviços e, consequentemente, prejuízos aos participantes nessa relação.

2.8.2 Conteúdo e Informação através dos Fornecedores

Na categoria Information Providers (IP) estão incluídos todos aqueles que oferecem informação através de uma página ou de um site. Por vezes, o proprietário da página ou site é também o organizador do conteúdo, seja próprio, seja de terceiros.

Esta definição de propriedade do conteúdo é importante para a definição de responsabilidades. Conteúdos próprios ou directos resultam das informações elaboradas por quem também é o realizador da página ou site, a exemplo de notas ou artigos publicados, cujos autores pertencem ao próprio fornecedor. Por sua vez, conteúdos indirectos, ou de terceiros, estão relacionados com links existentes nas páginas ou sites, não sendo a informação que ali consta gerada pelos próprios mentores desses locais.

Algumas doutrinas estrangeiras analisam a responsabilidade legal

deste tipo de fornecedor sob dois aspectos: o primeiro, por haver incluído a informação e o segundo, pelo conteúdo desta. Sem pretensão de alongamento deste assunto, deve-se aqui destacar, de modo especial, a situação dos links no contexto da responsabilidade do fornecedor de Internet. Entende-se por links o conjunto de indicações que constam de páginas ou sites, capazes de levar o navegador 'cibernauta' a ampliar o seu raio de informação ou navegação. Há links estritamente ligados ao conteúdo da página ou site e outros que figuram como mero material informativo, sobre os quais o fornecedor não tem qualquer responsabilidade. No primeiro caso, havendo dano, haveria responsabilidade objectiva; no segundo, subjectiva.

2.8.3 Hostings

Os Hosting Service Provider têm como função principal alojar páginas ou sites, ou seja, são uma espécie de 'hospedeiros tecnológicos virtuais'. Ao início, os hospedeiros, ao indicarem um meio através do qual os utilizadores possam conectar-se com outros, não têm qualquer responsabilidade no conteúdo das matérias inseridas nesses locais.

O hosting, apesar de aparentar pouca influência no mundo virtual, tem uma carga enorme de responsabilidade sobre a navegação do 'cibernauta', seja pelas indicações de formas de acesso, seja pelo aparato técnico que deve orientar as suas acções. Do mesmo modo que o Internet Service Provider, o Hosting tem todas as possibilidades económicas e técnicas para delegar, no sentido de realizar controlo e supervisão sobre os sítios e páginas sob o seu comando. O fornecedor de hospedagem seria assemelhado ao locador, já que concede o uso e o gozo de um site em troca do pagamento de um preço.

Segundo posição dominante nos EUA, dificilmente se responsabilizaria o fornecedor hospedeiro. Até agora entendeu-se que ele não administra o uso e gozo dos sites e páginas, não sendo responsável pelos conteúdos porque, além de não ser autor, não teve a oportunidade de aferir a ilegalidade desse conteúdo ou das informações capazes de causar danos a terceiros. Já em relação às implicações do uso de links, é necessário que se estabeleça uma diferença entre as empresas fornecedoras de conteúdo (responsabilidade objectiva) e aquelas que disponibilizam links, que seriam responsabilizados subjectivamente. Tanto os fornecedores de serviço como os de hospedagem teriam a responsabilidade avaliada de

forma subjectiva, derivada de sua falta de controlo sobre o conteúdo de páginas ou sites.

Capítulo 3

Ferramentas de Pesquisa

3.1 A Informação

Com a evolução da Internet e a busca intensa dos utilizadores por informação, surgiu mais esta forma inusitada de pesquisa através dos sites de busca. Muitas vezes o fornecedor de acesso, e mesmo o de conteúdo, não oferecem meios suficientes para a pesquisa. Então o utilizador recorre às várias espécies de ferramentas de busca.

Os métodos de busca são hoje uma das formas mais utilizadas para a procura e recuperação de informações na Internet. Várias pesquisas são feitas nessa área, procurando incrementar os processos automáticos de indexação e classificação da informação utilizados, a fim de melhorar a relevância e a velocidade da recuperação de dados na Web.

Nota-se que esses processos estão longe da perfeição e dependem, em grande parte, do preparo prévio dos documentos a serem indexados, uma tarefa para especialistas. Se a 'página' (documento HTML referenciado por uma URL) a ser indexada tiver as suas informações organizadas de maneira a que as máquinas lhes possam aceder e 'compreendê-las', os processos de indexação e classificação são então potencializados, resultando em índices de melhor qualidade.

Para se produzir uma página com estas características é necessário ter conhecimento de como as máquinas de busca realizam essa indexação, ou seja, quais factores são considerados por elas no momento em que analisam uma página. Sites de buscas são diferentes de livrarias ou

bibliotecas, onde a informação é ordenada e catalogada objectivamente, facilitando o contacto físico do utilizador. Nesses sites, mais parecidos com listas telefónicas, em que é normal uma empresa pagar para aparecer com mais destaque do que os seus concorrentes, o utilizador fica desmotivado absorvido por tanta informação, muitas vezes desordenada.

O problema é que, na busca por informação, o consumidor depara-se com links não gratuitos, os quais lhe irão causar sérios problemas no futuro. Em rigor, links pagos em sites de pesquisa não constitui um tema novo. Surgiu, pela primeira vez, em meados de 2001, nas páginas de resultados do Google, o 'balcão de informações' mais frequentado em todo o mundo. Na época, alguns grupos organizados criticaram a forma como os links pagos apareciam nas páginas. Diante das reclamações dos utilizadores, o Departamento de Comércio dos Estados Unidos interveio, implementando algumas directrizes na tentativa de regulamentar o serviço. A maioria dos sites de pesquisa enquadrou-se e passou a diferenciar os endereços publicitários nas listas de respostas. Os consumidores perderam a confiança nos sites de pesquisas desde que descobriram que uma parte dos resultados apresentados por eles não passam de publicidade paga.

Não se pode esconder a facilidade que estes mecanismos de busca trouxeram principalmente aos que se utilizam com frequência de fontes de pesquisa. Achar algo realmente importante em biliões de páginas que formam o mundo virtual tornou-se impossível sem a ajuda de uma boa ferramenta de busca. No documento da Consumer WebWatch destaca-se a insatisfação dos consumidores depois de descobrirem as armadilhas da publicidade embutida nesses sites.

3.1.1 Tipos de Busca

Já existem ferramentas, tais como AltaVista, Yahoo e Google (talvez reconhecidas como as mais utilizadas ultimamente) que auxiliam a busca de informações na Internet. Ferramentas como estas permitem a localização de documentos textuais a partir de palavras ou catálogos de assuntos. Os utilizadores fornecem as palavras desejadas ou escolhem assuntos do seu interesse, e as ferramentas devolvem os documentos correspondentes, bem como os sites onde se encontram estes documentos.

Nas principais ferramentas, duas técnicas de recuperação são utiliza-

das para este processo de busca: 1ª. a indexação de termos ou palavras que geralmente é full-text, ou seja, todas as palavras dos documentos tornam-se disponíveis no índice e o utilizador recebe como resposta os documentos que contêm as palavras fornecidas como entrada;

2ª. a catalogação de documentos: que ocorre quando alguma pessoa define o assunto do documento; o utilizador precisa então de escolher um dos assuntos já predefinidos para então receber os documentos relativos àquele assunto (catalogados dentro do assunto).

Algumas variações da primeira técnica são comuns. Por exemplo, em algumas ferramentas, há uma linguagem própria para consulta, que utiliza símbolos lógicos para eliminar documentos com determinados termos ou para recuperar somente documentos que contenham obrigatoriamente certos termos (na falta de informações mais precisas, bastará ao documento, para satisfazer a consulta, conter um dos termos fornecidos).

Apesar da incontestável utilidade destas ferramentas, alguns problemas podem ocorrer. Primeiro, a maioria dos utilizadores que utilizam as ferramentas de localização é inexperiente ou leiga, tanto no assunto que procuram quanto na utilização da ferramenta em si. Portanto, têm dificuldades em definir o contexto da informação que necessitam utilizando palavras e símbolos lógicos.

Também ocorre que as diversas ferramentas agrupam as informações de diferentes maneiras. Aqueles que conhecem o funcionamento interno da ferramenta, e possuem mais experiência com a linguagem de consulta (que é também específica da ferramenta) têm mais facilidade de encontrar informações úteis.

Além disto, as ferramentas que utilizam a técnica de indexação devolvem grandes volumes de documentos sem a certeza de que a informação desejada se encontra num deles.

Isto acontece porque a técnica de indexação é baseada unicamente na presença de termos nos documentos. Assim, podem ser devolvidos documentos que contêm as palavras fornecidas, mas que se referem a outro contexto, devido à possibilidade de as palavras terem vários significados diferentes.

Outro problema é que poderão deixar de ser recuperados documentos relevantes para o assunto escolhido, justamente porque não possuem os termos fornecidos.

Quando a segunda técnica é utilizada, podem ocorrer problemas quando o especialista cataloga documentos de forma errada (por exemplo, interpretando equivocadamente o conteúdo de um documento) ou quando o utilizador não consegue encontrar um assunto (dos pré-definidos) que represente precisamente os seus interesses de pesquisa (já que as pessoas podem utilizar termos diferentes para associar a mesma ideia ou termos iguais para ideias diferentes - este problema é conhecido como 'abismo semântico').

Além disso, uma vez que as ferramentas recuperam um grande número de documentos (na ordem de milhares, em média), haverá a necessidade de refinamentos sucessivos até que o utilizador encontre as informações desejadas. Isto ocorre porque, geralmente, o utilizador fornece inicialmente apenas alguns poucos termos para pesquisa.

À medida que os documentos vão sendo devolvidos ao utilizador, deverão ser feitas novas análises sobre os mesmos (ou pelo próprio utilizador ou por ferramentas), para filtrar um conjunto menor de saída.

Uma das alternativas para este último problema é listar os documentos de saída de maneira ordenada segundo algum critério.

Por exemplo, os documentos com maior frequência (presença) dos termos poderão estar no topo da lista.

Quando o utilizador escolhe palavras ou tópicos para descrever os assuntos pelos quais se interessa, dois problemas principais podem ocorrer:

1) o utilizador pode ter escolhido termos ou tópicos não adequados para representar as suas ideias e interesses;

2) a ferramenta de busca pode entender equivocadamente o significado das palavras fornecidas.

Isto ocorre frequentemente porque pessoas diferentes podem utilizar os mesmos termos para ideias diferentes ou então utilizar termos diferentes para as mesmas ideias.

A causa de tais imprecisões e ambiguidades está no centro do processo de comunicação: é impossível transmitir significados; só unidades linguísticas mínimas podem ser transmitidas.

O significado está na mente das pessoas e não nas marcas gráficas.

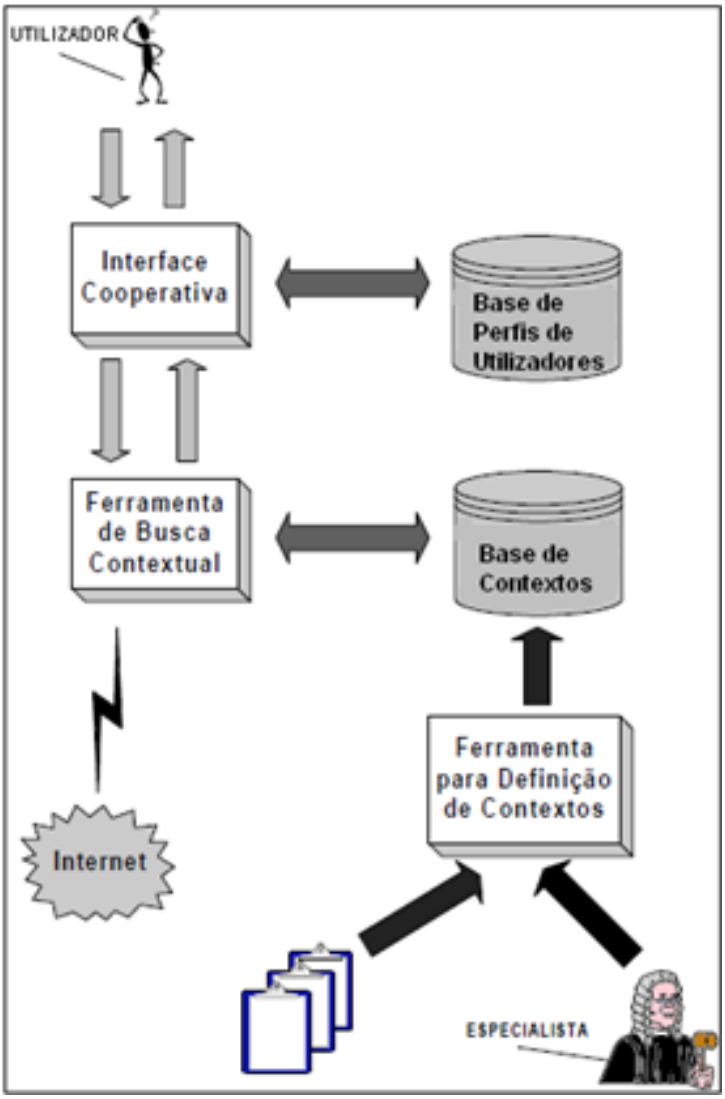

Figura: 3.1: Definição de contexto na busca.

Para minorar tais problemas, deverá ser analisado também o contexto em que os termos são usados, isto é, o resultado da análise do texto pela pessoa. A comunicação obtém sucesso quando são analisados não só os elementos físicos envolvidos, mas também os aspectos sociais, humanos, emotivos, etc, que acompanham o processo de comunicação.

O contexto, é portanto, tudo aquilo que envolve um processo de comunicação, seja ou não transmitido explicitamente. As técnicas para indexação devem gerar índices mais intimamente relacionados com o real significado de um texto em particular e não baseados na presença de termos sem identificação do contexto.

Há três ferramentas. A primeira (interface cooperativa) interage com o utilizador recebendo as suas consultas (expressas em palavras para busca), apresentando os documentos resultantes e refazendo o processo de interacção através de novas selecções de palavras pelo utilizador. Esta ferramenta também documenta todo o processo de interacção, bem como os documentos da Web escolhidos pelo utilizador para visualização. A segunda (e mais importante) ferramenta (a de Busca Contextual) realiza a comunicação com algum software de indexação já disponível na Internet (exemplo o AltaVista). Antes, porém, esta ferramenta deve determinar o espaço de busca, ou seja, o contexto dos termos fornecidos, a fim de evitar interpretações erradas de significados. Para tal, será utilizada a Base de Contextos. A última ferramenta é a que permite a Definição dos Contextos (na Base de Contextos), com a intervenção de um especialista humano ou automaticamente.

A Ferramenta de Busca Contextual é responsável por identificar, a partir dos termos fornecidos pelo utilizador, o conjunto de termos que serão utilizados para a pesquisa de documentos na Internet. Este conjunto de termos deve ser o que melhor define o Contexto do assunto desejado e será extraído da Base de Contextos.

Para encontrar os documentos na rede, esta ferramenta tira partido dos sistemas de indexação já disponíveis (como AltaVista, Yahoo, etc), passando como parâmetros as palavras do contexto (com os símbolos lógicos correspondentes) e recebendo como resposta as URL's dos documentos candidatos.

Na situação mais simples, a Ferramenta de Busca segue os seguintes passos:

1) recebe uma palavra do utilizador; 2) procura na Base de Contextos as palavras relacionadas com aquela; 3) envia os parâmetros de busca a um sistema de indexação (sendo que os termos serão associados por conjunção, ou seja, basta a presença de um dos termos); 4) recebe as URL's dos documentos candidatos a resposta; 5) procura directamente os documentos apontados; 6) realiza a análise do conteúdo destes do-

cumentos (verifica a frequência dos primeiros termos); 7) apresenta ao utilizador (pela interface cooperativa) os documentos com os seus termos mais frequentes e a sua pontuação.

A pontuação é determinada pela presença dos termos de busca. A fórmula utilizada é simples: cada vez que o termo aparece no documento, soma-se um ponto (técnicas mais sofisticadas, usando lógica fuzzy, estão em estudo). Uma lista será então apresentada ao utilizador, que poderá consultar directamente os documentos ou então poderá seleccionar um novo conjunto de palavras (fornecendo um feedback) para uma filtragem dos documentos candidatos.

Tal processo pode-se repetir inúmeras vezes, convergindo para subconjuntos cada vez menores de documentos, até que o utilizador esteja satisfeito. Em casos mais complexos, quando o termo fornecido como entrada pelo utilizador estiver relacionado com mais de que um contexto diferente, será necessário escolher um dos contextos para a busca, determinando qual dos significados do termo é o mais adequado. Para tal, o contexto do utilizador (um conjunto de termos representativos) será consultado na Base de Perfis de Utilizadores, através da Ferramenta de Interface Cooperativa. Nesta segunda situação, este conjunto de termos será usado para determinar qual o contexto do interesse do utilizador. A técnica utilizada é a da referência cruzada entre os conjuntos com pontuação pela presença dos termos.

Actualmente, nem todos os passos da ferramenta estão implementados (por exemplo, a parte do feedback e dos refinamentos sucessivos por interacção com o utilizador).

As técnicas empregues na extracção de relações automáticas entre palavras de um mesmo contexto é similar à técnica utilizada na montagem de Thesaurus. São métodos estatísticos, que se baseiam na análise de ocorrência das palavras nos documentos.

Ao todo o processo de montagem automática de contextos possui três etapas distintas:

- a identificação de palavras nos documentos, - a determinação do grau de relação entre as palavras e o documento que as contém, - a análise das relações entre as palavras.

As palavras que aparecem repetidamente num único documento e as palavras que aparecem em muitos documentos são boas candidatas.

É claro que nem todas as palavras devem ser indexadas. As palavras conhecidas como 'Stop-words não devem ser adicionadas' [Chen, 1994].

As Stop-words são palavras comuns a todos os textos (por exemplo, artigos e preposições) e portanto não são específicas do assunto tratado pelo documento e podem variar.

Dependendo do domínio a ser analisado, verbos ou até mesmo expressões podem ser desprezadas. Após seleccionadas as palavras que devem fazer parte do processo, é realizada uma análise de co-ocorrência das palavras nos documentos. É através desta análise que é possível definir o grau de relação de cada palavra com o contexto em questão. Duas fórmulas são utilizadas para esta análise: a fórmula que analisa o grau de relação entre uma palavra e um documento, e a fórmula que analisa as relações entre palavras (definindo assim os contextos).

A fórmula abaixo define a relação entre uma palavra e o documento em que ela aparece, onde dij é o valor combinado da palavra j no documento i:

$$d_{ij} = tf_{ij} \times \log\left(\frac{N}{df_j}\right)$$

N representa o número total de documentos na BD, tfij é a frequência da palavra j no documento i e dfj é a frequência inversa de documentos (número de documentos em que a palavra j aparece).

A segunda fórmula avalia os resultados gerados pela fórmula anterior, detectando as relações entre as palavras:

$$\text{Valor combinado} = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}}, \text{ onde } d_{ijk} = tf_{ijk} \times \log\left(\frac{N}{df_{jk}}\right)$$

Sendo que tfijk representa o número de ocorrências de ambos as palavras j e k no documento i (o menor número de ocorrências entre as palavras deve ser escolhido), dfjk representa o número de documentos (numa colecção de N) no qual as palavras j e k ocorrem ao mesmo tempo.

Desta forma, é possível identificar as relações entre as palavras em diversos contextos, criando uma estrutura (a Base de Contextos) capaz de indicar o quanto uma palavra está relacionada com outra em determinado contexto. Tendo-se esta informação é possível identificar a que contexto determinado documento pertence.

É possível também estabelecer um grau de pertinência do documento a determinado contexto, isto é, caso se identifique mais de um contexto num único documento, é possível estabelecer quanto este documento está relacionado com um e com outro contexto.

É importante lembrar que duas palavras podem estar relacionadas entre si em mais de um contexto, e, portanto, em cada contexto existe um grau de relação diferente.

A Base de Contextos é uma estrutura que armazena os relacionamentos entre as palavras de um mesmo contexto. Assim, é possível percorrer esta estrutura e identificar quais são as palavras que pertencem a um determinado contexto.

É possível também saber o quanto uma palavra está relacionada com o contexto ou também o quanto uma palavra está relacionada com outra palavra em determinado contexto.

A implementação actual da base de contextos, como está sendo utilizada pela ferramenta de Busca Contextual, é muito simples. Cada contexto é nomeado e identificado por uma palavra e contém um conjunto de palavras que o representam.

Alternativas para melhorar tal estrutura estão a ser testadas. Uma implementação possível é estruturar a Base de Contextos como uma rede semântica, onde os 'nodos' são as palavras e os elos representam relações entre palavras de um mesmo contexto.

Entretanto, como pode haver o caso de uma palavra estar relacionada com duas outras em contextos diferentes, há a necessidade de se caracterizar o contexto de cada elo.

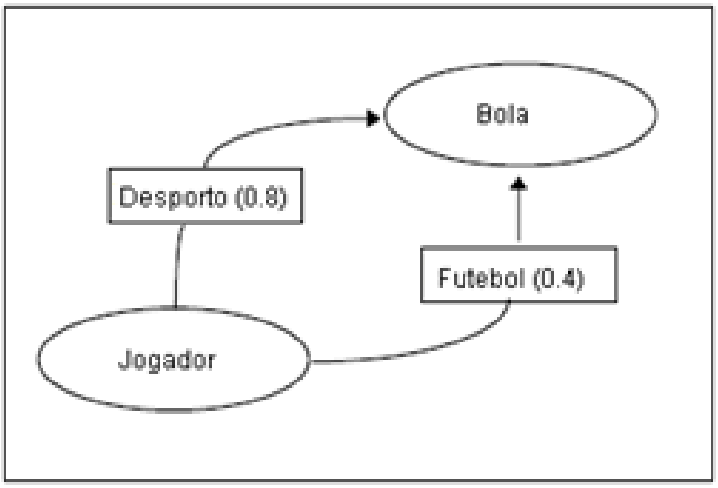

Figura: 3.2: Associação de Contexto

Duas palavras podem estar relacionadas entre si nos mais diversos contextos. Cada elo possui uma indicação do contexto da relação (inclusive determinando o tipo da relação; por exemplo, causa, efeito, sinónimo, etc.) e um valor, o qual caracteriza o grau de associação entre as palavras correspondentes (como visto nas fórmulas mencionadas).

3.1.2 Resultado de Buscas

Partindo da inserção de uma determinada palavra ou um conjunto delas pelo utilizador, o sistema de pesquisa escolhido pelo utilizador ('cibernauta') realiza um vasto varrimento nas páginas disponíveis, identificando em ordem significativa (ou por número de acessos) as ocorrências de maior semelhança para que o utilizador restrinja assim as possibilidades de erro.

Mas, somente a escolha correcta das palavras não é garantia de uma busca bem sucedida. O grau de refinamento atingido pelos grandes sites de busca permite realizar verdadeiras proezas, desde que se conheçam os critérios e operadores específicos para satisfazer uma determinada pesquisa. Para exemplificar o uso de uma ferramenta de busca, será utilizado o Google - bastante conhecido por parte dos utilizadores e muito procurado actualmente - (www.google.com), que com sua simplicidade

e objectividade atingiu o topo da lista nessa área, passando por mais de oito biliões de páginas nas suas pesquisas. A procura começa com a escolha da palavra que melhor define o assunto da pesquisa.

Nesse primeiro momento, os resultados devolvidos são muito amplos e podem não atender ao objectivo: pode usar como exemplo uma busca (páginas de Portugal) a partir da palavra revolução. Esta devolve perto de um milhão de ocorrências tornando muito amplas as possibilidades e incondicionais. Suponha-se que o utilizador queira saber sobre a revolução industrial: acrescentando-se a palavra industrial, os resultados caem para menos de setenta mil ocorrências. Caso o desejo seja encontrar a ocorrência exacta e nessa ordem da expressão: revolução industrial coloca-se entre aspas desta forma: 'revolução industrial', resultando em cerca de quarenta mil ocorrências. O uso da letra 'e' entre as duas palavras, revolução e industrial, funciona da mesma forma, retornando os mesmos resultados.

Suponha-se que o utilizador queira ir ainda mais fundo na sua busca e deseje pesquisar sobre a revolução industrial apenas na Inglaterra, basta apenas acrescentar um espaço após a última aspa e o sinal de soma seguido da palavra Inglaterra: 'revolução industrial' + Inglaterra, para que se tenham cerca de seis mil e quinhentas ocorrências.

Uma outra maneira bastante útil para reduzir o trabalho de busca é pesquisar a ocorrência de palavra(s) numa página específica, principalmente quando se tem um conhecimento prévio da existência daquele assunto em tal base de dados, como é o caso de um livro, uma tese ou um paper numa biblioteca. Para isto, digita-se a palavra que se quer pesquisar, seguida de um espaço e a palavra site seguida de dois pontos, seguida do endereço electrónico que vai servir de fonte da pesquisa. A busca devolverá as ocorrências da palavra revolução dentro do domínio solicitado. Podem-se usar, da mesma forma, aspas para buscar várias palavras dentro de uma página. Os operadores disponíveis para aumentar o número de palavras pesquisadas e também restringir o espaço pesquisado são muito delimitadores e possibilitam uma redução brutal no tempo despendido na filtragem dos resultados.

Diversos outros sites de busca possuem estratégias de pesquisa ainda mais facilitadoras para o utilizador, pois, ao invés de empregarem operadores, disponibilizam diversas entradas que devem ser preenchidas com as palavras que se pretendam pesquisar e campos para que, se marca-

dos, definam como os resultados devem ser mostrados: com todas as palavras, a expressão exacta, qualquer uma destas palavras, qualquer domínio, apenas domínios específicos.

Diante de tantas facilidades, porque é que tantas pessoas têm dificuldades de utilizar as ferramentas de busca? Pode-se dizer que isto se deva, principalmente, ao facto de muitas pessoas não procurarem (antes da pesquisa) ler o tópico de ajuda da ferramenta de busca ou não entrarem no campo de pesquisa avançada para se inteirarem sobre as suas particularidades. A justificação mais usada habitualmente por parte dos utilizadores seria a falta de tempo, acompanhado pelo cepticismo em utilizar os mecanismos de ajuda, talvez uma herança dos programas e sistemas operativos, cuja ajuda muitas vezes complicava mais ainda a cabeça do utilizador.

3.1.3 A Interface Cooperativa e a Base de Perfis de Utilizadores

A Interface Cooperativa registará todas as interacções do utilizador com o sistema, mantendo um histórico na chamada Base de Perfis de Utilizadores. Devem ser documentadas todas as pesquisas realizadas, os documentos recuperados, aqueles que foram lidos e os que foram rejeitados, as palavras utilizadas pelo utilizador para a pesquisa, etc, além de algum tipo de identificação do utilizador.

Estas informações serão transmitidas à ferramenta de busca contextual para que possa deduzir objectivos e planos do utilizador, a fim de 'filtrar' os documentos recuperados, apresentando apenas as informações relevantes ao interesse do utilizador.

Uma questão que surge sempre quando existe a utilização de um modelo do utilizador, é referente ao seu conteúdo inicial. As alternativas, com algumas variações, resumem-se a iniciar com o modelo vazio ou com um conteúdo predeterminado, igual para todos os novos utilizadores. Iniciar com o modelo vazio reduz o poder de dedução da ferramenta nas interacções iniciais. Iniciar com um conteúdo predeterminado pode levar a problemas como a inadequação do modelo ao utilizador. Ou seja, é um risco.

No momento, a implementação da base de perfis apenas contempla o conteúdo das interacções. Ou seja, estão a ser armazenados somente a

identificação do utilizador e um conjunto de palavras que define seus interesses. Portanto, não estão a ser armazenados os tipos de informações (se documentos retornados ou termos fornecidos como entrada), mas somente o seu conteúdo descrito por palavras. Também não estão a ser considerados aspectos de tempo (consultas antigas, consultas mais recentes) e os resultados rejeitados. As palavras que definem o perfil ou contexto do utilizador estão a ser extraídas dos termos fornecidos pelo utilizador e dos termos mais frequentes dos últimos documentos seleccionados pelo utilizador. Caso haja repetidos processos de feedback, estão a ser desconsiderados os aspectos intermediários da interacção ferramenta-utilizador, na implementação actual.

A ferramenta de Busca Contextual tem-se mostrado útil ao escolher um conjunto maior de termos para busca e por ir refinando o espaço de busca, através da convergência dos documentos que vão sendo localizados. Ao oferecer documentos candidatos e auxílios ao utilizador para a tomada de decisão, a ferramenta gera resultados melhores, mesmo que mais demorados. Além disto, o significado dos termos pode ser melhor interpretado com a ajuda das relações entre os termos (tanto na base de contextos, quanto na base de perfis), diminuindo assim os erros por ambiguidade. Experimentos com textos retirados de artigos e reportagens de jornais e sobre manuais médicos conduzem a estas conclusões. Uma avaliação mais rigorosa deverá ser feita para determinar o grau de acerto na recuperação dos documentos.

Para tanto, pode-se utilizar os critérios de Abrangência (recuperar todos os documentos relevantes) e Precisão (recuperar somente os documentos relevantes). Cabe salientar que o sucesso desta abordagem depende em muito de como a base de contextos é criada. Uma boa base permitirá melhores interpretações dos interesses do utilizador, enquanto que uma base pobre ou mal-definida ocasionará erros no processo de busca (retorno de documentos não desejados ou falta de documentos importantes).

Quando a Base de Contextos é criada por um especialista a probabilidade de erros pode ser maior (por razões já discutidas anteriormente). Quando a ferramenta de Definição dos Contextos faz esta definição automaticamente a partir de documentos predeterminados, diminui-se a incerteza, pois são utilizadas técnicas já consagradas na literatura para determinar as palavras representantes de um assunto e podem ser usados volumes maiores de documentos para análise. Entretanto, tais técnicas

somente terão resultados satisfatórios se o conjunto-amostra para extracção das relações entre os termos for bem escolhido. De novo, recai-se na dependência de um especialista humano. Também poderá haver problemas se o termo escolhido para definir o contexto (núcleo do conjunto) não for apropriadamente escolhido.

Em parte, tal situação pode ser contornada com o uso de sinónimos. Uma solução é a de incrementar e refinar a base de contextos automaticamente a partir de entradas de vários especialistas ou por análise dos documentos seleccionados por vários utilizadores diferentes. Assim, os contextos seriam definidos de maneira a combinar o conhecimento de vários especialistas, tornando a ferramenta também um Sistema Colaborativo.

Problemas também ocorrem quando uma palavra aparece como núcleo num contexto e como elemento noutro.

Já a Base de Perfis de Utilizadores, por sua vez, pode levar a ferramenta a conclusões equivocadas. Isto pode ocorrer quando o utilizador procura informações num contexto diferente daquele que a ferramenta deduziu ou quando o utilizador realmente quer alterar seu assunto de busca. Uma das alternativas possíveis é consultar o utilizador todas as vezes que houver algum conflito a ser resolvido. Desta forma, a ferramenta tornaria-se um 'assistente de consulta'.

Outra limitação da abordagem exposta é que as consultas não têm em conta a sintaxe-semântica entre os termos, mas apenas o contexto no qual se inserem. Por exemplo, se forem fornecidas como entrada as palavras 'ferramentas' + 'Internet', podem ser recuperados documentos que tratam de 'ferramentas exclusivamente para o uso da Internet' ou de 'quaisquer ferramentas disponíveis na Internet'. Os refinamentos sucessivos através do feedback do utilizador podem compensar estas desvantagens.

3.2 As Categorias Funcionais para Extracção Integrada

Uma visão alternativa da Web diz respeito à funcionalidade das páginas, dividindo-as de acordo com o seu papel na ligação entre páginas e na apresentação e armazenamento de dados relevantes.

Esta visão baseia-se na verificação de listas de resultados devolvidos pelos mecanismos de busca para o processamento de uma classe.

Assim, dado este propósito e visando a extracção integrada, as categorias funcionais estão assim divididas:

- Páginas-conteúdo, que são páginas que pertencem à classe que está a ser processada, e de onde serão extraídas a(s) entidade(s) em questão.

- Páginas auxiliares, que, apontadas exclusivamente por páginas-conteúdo, hospedam atributos específicos da(s) entidade(s) da página que a aponta.

- Listas de páginas-conteúdo, também chamadas de directórios ou índices, de grande utilidade na localização segura e contextualizada de páginas-conteúdo, por ser composta basicamente de âncoras para páginas-conteúdo.

- Mensagens ou Listas de Mensagens, que contêm correspondências ou listas delas sobre assuntos correspondentes à entidade que está sendo extraída, que usualmente não possuem utilidade para extracção integrada, por serem páginas muito longas, por apontar para páginas trazendo informações muito inadequadas e com relacionamentos difíceis de serem identificados.

- Recomendações, que são páginas-conteúdo que pertencem a outra classe, podendo ser aproveitadas quando ocorre o processamento desta outra classe, acelerando o processo de busca desta classe.

- Simplesmente lixo, ou seja, páginas sem qualquer utilidade para a extracção, por não pertencerem a nenhum dos itens anteriores.

Cabem algumas notas a respeito destas categorias funcionais.

Em primeiro lugar, convém salientar que é possível a existência de listas em páginas-conteúdo.

Por exemplo, a bibliografia de um artigo científico pode constituir uma lista de links para outros artigos.

Podem inclusive existir listas noutras páginas de outras classes, chamadas de eventos científicos frequentemente fornecem uma lista de ân-

coras de pesquisadores nos seus comités de programa, por exemplo.

Há classes de páginas em que cada entidade pode representar uma lista de outra classe.

Uma instância de Publicação Divisível, com subclasses como livros e proceedings, pode conter uma lista de links para artigos científicos, que, não por acaso, é subclasse de Publicação Parte, dentro de uma taxonomia sobre o meio científico.

Observe-se ainda que páginas auxiliares podem ajudar a encontrar a entidade a que se referem, através de um ponteiro para a página autoritária (como 'Home') ou do prefixo imediatamente superior do seu endereço electrónico.

Ressalve-se ainda que o comportamento e utilidade das categorias funcionais, e também o relacionamento entre elas para a extracção integrada permanece fixo, conforme definido acima.

Assim, durante a extracção, a classificação de páginas resultantes de consultas a mecanismos de busca com relação a estas categorias, assim como a identificação de páginas autoritárias - neste contexto, usa-se o termo 'página autoritativa' para denotar a página principal de uma entidade qualquer, seja ela uma página de um pesquisador ou de chamada de trabalhos, que contenha ponteiros para páginas onde estão os atributos da entidade, diferente do termo usual, que designa exclusivamente a página principal de uma instituição ou empresa, com nome de domínio - com relação à entidade processada, proporciona um refinamento fundamental no reconhecimento de páginas contendo dados realmente pertinentes.

As categorias funcionais e os seus relacionamentos encontram-se detalhados na figura seguinte.

A área sombreada indica categorias sem utilidade para o processamento de uma classe.

As setas entre as categorias indicam relacionamentos.

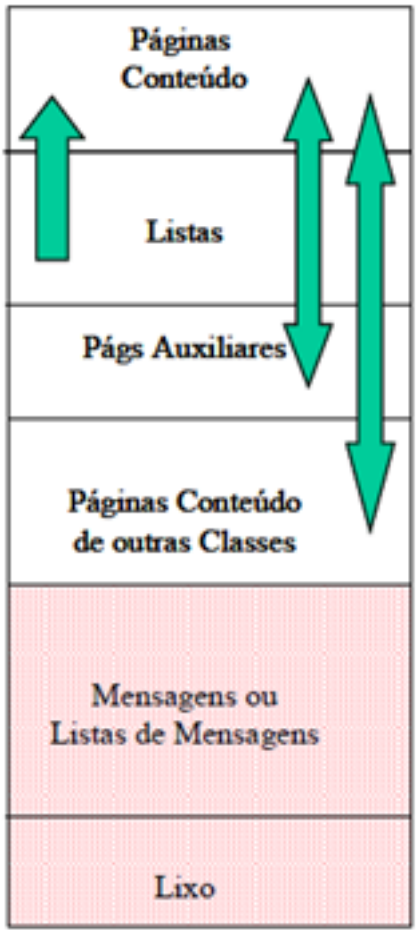

Figura: 3.3: As categorias funcionais e seus relacionamentos.

3.3 Inteligência Artificial

A noção de agente inteligente (ou racional) há muito tempo tem servido à comunidade de IA (Inteligência Artificial) como paradigma na construção de sistemas inteligentes [Minsky, 1986]. Devido a seu alto grau de abstracção, a noção de agente permite que se 'descreva' um sistema em termos de ambiente, acções, dados perceptivos e objectivos, evitando considerações prematuras sobre a maneira como ele foi ou será implementado (linguagens de representação do conhecimento, mecanis-

mos de inferência, etc).

Actualmente, a tecnologia de agentes inteligentes vem extrapolando as fronteiras da Inteligência Artificial, atraindo interesse de vários ramos da Ciência da Computação, e ganhando cada vez mais notoriedade na resolução de uma vasta gama de problemas complexos, e, em especial, daqueles relacionados com a Internet.

Devido a esta proliferação de tipos e áreas de aplicação, há actualmente uma certa controvérsia que reina sobre o que é realmente um agente. Entretanto, existem algumas características consensuais. Os agentes inteligentes devem tentar atender a uma ou mais metas de forma autónoma, decidindo que acções devem tomar pelas percepções do ambiente, provocando mudanças nesse ambiente. Devem ainda apresentar robustez face a situações não previstas de antemão (o que entra em confronto directo com a forma tradicional de programação), muitas vezes possuindo uma representação, dentro de si, do mundo em que se encontra submerso.

Diferentemente dos sistemas tradicionais de Inteligência Artificial, que possuíam competências especializadas e justificadas pela necessidade de robustez, os agentes inteligentes devem apresentar múltiplas competências integradas, especializadas ou não, que lhes garantam versatilidade. Tanto quanto possível, devem tentar aprender e melhorar a sua performance através de uma avaliação das suas acções. Se o ambiente assim o permitir, mobilidade é uma qualidade desejável, e alguns deles chegam a exibir personalidade, os chamados agentes credíveis (believable agents).

Existem três tipos de ambiente nos quais os agentes inteligentes podem actuar: o mundo físico, o mundo computacional - a Internet ou outros ambientes computacionais menores - e o mundo da realidade virtual, uma simulação de um mundo qualquer com características definidas a que o agente se deve adequar. Reciprocamente à utilidade dos agentes na Internet, a própria Internet, enquanto mundo computacional, tornou-se um ambiente bastante propício aos agentes inteligentes, por tratar-se de um ambiente de menor complexidade que o mundo físico, mais acessível e fácil de controlar. Em robots - agentes submersos no mundo físico - a problemática de lidar com o ambiente releva-se de complexidade muito superior, pois o robot defronta-se com situações inesperadas com muito mais frequência.

3.4 Inteligência Artificial Distribuída

Na medida em que mais agentes começaram a surgir, e tarefas mais complexas lhes eram ministradas, as qualidades de sociabilidade e comunicabilidade fizeram-se relevantes, inaugurando a Inteligência Artificial Distribuída, baseada em sociedades de agentes.

A Inteligência Artificial clássica baseava-se numa metáfora psicológica da inteligência, onde uma pessoa ou entidade resolvia os problemas propostos, e a inteligência era vista como atomizada, pois restringia-se aos micro-aspectos da sua própria racionalidade. Inspirada em áreas tão diversas como linguística, sociologia, economia, filosofia e biologia, a Inteligência Artificial Distribuída complementa a metáfora psicológica com uma metáfora sociológica, referente aos macro aspectos dos agentes enquanto sociedade. As soluções dos problemas propostos emergem de acções e interacções produtivas entre os agentes membros de uma mesma sociedade. Uma das mais fortes inspirações dos sistemas multiagentes partiu de uma modelagem da própria mente humana. O livro 'Sociedade da Mente' [Minsky, 1986] traça mecanismos do cérebro e do pensamento. Segundo a obra, existe um grande número de agências por mente, muito variada e rica em termos de tipos de conhecimento que armazenam metas, representações e estímulos a que estão sujeitas. As acções e soluções para a tomada de decisões apareceriam das interacções entre estas agências, através de conflitos, comunicação, hierarquias e outros mecanismos bastante similares aos sistemas multiagentes de hoje.

3.5 Comunicação entre Agentes Cognitivos

A comunicação directa entre agentes abrange dois modelos: O modelo Cliente-Servidor e o modelo peer-to-peer.

O modelo Cliente-Servidor baseia-se em chamadas a procedimentos remotos, enquanto internamente efectua uma comunicação do tipo pedido-resposta com os parâmetros do procedimento solicitado.

Já o modelo peer-to-peer baseia-se na Teoria dos Actos da Fala. A comunicação humana tem sido modelada por esta teoria, que considera que a linguagem falada tem por objectivo engendrar acções e provocar mudanças no ambiente. Os Actos da Fala, anteriormente estudados em

Processamento de Linguagem Natural, uma subárea de Inteligência Artificial Simbólica, são classificados como assertivos (informar), directivos (pedir ou consultar), comissivos (prometer ou comprometer-se), proibitivos, declarativos (causar eventos para o próprio comunicador) e expressivos (emoções). A comunicação entre agentes dotados de autonomia e inferência tem seguido este modelo, que sob o ponto de vista de semântica é muito mais claro e abrangente do que o modelo Cliente-Servidor. O modelo peer-to-peer propõe uma comunicação proactiva - ou seja, qualquer agente pode iniciá-la -, baseada em actos de fala (que expressam as intenções dos agentes) e com conteúdo preferencialmente baseado em conhecimento declarativo, sendo por isso chamada de comunicação em nível de conhecimento.

As condições para que conhecimento possa ser trocado e compreendido pelos agentes em comunicação em nível de conhecimento são:

* A intenção pragmática de cada mensagem, definida pelo acto de fala correspondente (como informar, pedir, recrutar, etc), deve fazer parte do protocolo de comunicação;

* O conhecimento contido na mensagem deve estar escrito em algum formalismo de representação de conhecimento que garanta que a sintaxe da mesma possa ser entendida pelos os agentes, com a possibilidade de emprego de mecanismos de tradução entre estes formalismos;

As mensagens devem referir-se a um contexto e vocabulário comuns, sobre o qual a troca de mensagens possa ser efectuada dentro de uma semântica bem definida e sem ambiguidades. Esse contexto é normalmente fornecido pelo que chamamos de ontologia partilhada ou reutilizável. A figura 3.4 mostra agentes em comunicação em nível de conhecimento. Assim, já existe tecnologia disponível para a troca de conhecimento declarativo entre agentes, permitindo cooperação entre eles na execução das suas metas comuns ou individuais, alcançando, por conseguinte, distribuição e concorrência. A necessidade de comunicação ao nível do conhecimento é tão importante na resolução de problemas complexos na Internet que traz à tona a antiga controvérsia declarativo-procedimental, dados os imensos benefícios produzidos pela comunicação ao nível do conhecimento.

Com efeito, para um ambiente do porte da Internet, a inteligência na resolução de problemas reside na possibilidade de cooperação entre softwares autónomos distintos, ou agentes inteligentes, baseados em co-

nhecimento explícito e com capacidade de comunicação ao nível do conhecimento, o que significa a possibilidade de enviar conhecimento definido num formalismo de representação de conhecimento compreensível por outros agentes.

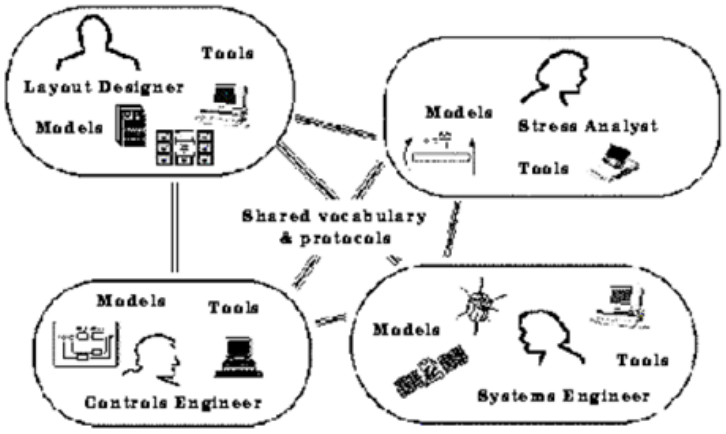

Figura: 3.4: Comunicação ao nível do conhecimento, através de protocolos e vocabulário comum, apesar de cada componente ter o seu próprio agente ou sistema especialista

A necessidade de melhores ferramentas de busca para a Web terminou por disseminar o termo 'agente' para praticamente qualquer aplicação que se propusesse a manipular ou procurar informação na rede. Neste trilho, o novo termo da moda, que vem herdando essa popularidade, face às promessas de melhoria destas actividades, é o termo ontologia, uma vez que elas estão presentes em muitos sistemas, ferramentas e produtos de manipulação de informação e comércio electrónico, representadas como hierarquias de palavras-chave, conceitos, e muitas outras formas, chegando a haver, inclusive, vagas no mercado para desenvolvimento de ontologias, ou 'ontologistas'.

Conforme dito anteriormente, os sistemas abertos, as WANs (Wide Area Networks, ou redes de abrangência alargada), a Internet, o crescimento da capacidade de conectividade, e a impossibilidade de estruturação de redes cada vez maiores e os seus respectivos documentos propiciaram uma revisão às técnicas de Inteligência Artificial, como a alternativa mais viável para um melhor tratamento aos problemas destes ambientes. Basicamente, dois tipos de solução foram propostos, que não são

mutuamente exclusivas:

* Dotar os sistemas de inteligência e autonomia para percorrer e seleccionar informação relevante na imensidão da rede, o que veio a originar os chamados agentes inteligentes;

* Dotar as redes e a própria Internet de inteligência, fazendo com que as páginas possuam uma semântica mais clara e definida, já que as ferramentas que lhes dão suporte, a linguagem HTML (HyperText Markup Language) e o protocolo HTTP (HyperText Transfer Protocol) preocupam-se, no seu estado actual, com apresentação e navegação, e só por isso podem ser aproveitados em buscas lexicais e mensuração de proximidade de texto, e não em buscas semânticas com contexto delimitado.

Vale a pena salientar que as ontologias representam um papel fundamental em ambos os tipos: no primeiro como elemento de comunicação de agentes, organização, reutilização e disseminação de conhecimento, e no segundo como parte intrínseca das linguagens propostas para a definição de páginas com semântica, como será visto logo a seguir. A arquitectura a ser apresentada neste trabalho enquadra-se no primeiro tipo; esta decisão será justificada, após ser explanado o segundo tipo.

3.5.1 Esclarecendo Ontologias

Usando um componente de RDF, o RDFS (RDF Schema), ontologias podem ser representadas, e assim, pode ser efectuada inferência sobre páginas e consultas mais complexas ainda. Contudo, este trabalho adopta uma estratégia baseada em agentes, levando em conta que organizar toda a Internet ontologicamente embate em problemas de várias naturezas:

* O utilizador comum, que navega na rede e publica páginas, mesmo com interfaces gráficas teria dificuldade em formular consultas que envolvessem regras de lógica e ontologias, e também de lidar com as complexidades em especificar ontologias ou instanciá-las nas suas próprias páginas, ainda mais em padrões que se sobrepõem em várias camadas, como o trio OIL-RDF-XML;

* Novos problemas surgiriam, relacionados com a veracidade e correcção das especificações contidas nas páginas;

* As próprias páginas possuem, às vezes, conteúdo ambíguo e vago,

e isso pode fazer parte da própria informação, seja pelo seu conteúdo (como, por exemplo, em poesia) ou pela localidade do vocabulário empregue, mantendo o problema da linguagem natural;

* Dificilmente um padrão ontológico para a codificação de páginas, tanto no que se refere à linguagem de markup, como de ontologias a referenciar, será adoptado pela 'rede da liberdade' num curto espaço de tempo;

* Ainda que o fosse, problemas de escalabilidade tanto das ontologias como da indexação ontológica estariam presentes;

* A adaptação da arquitectura desta proposta frente a uma Internet semântica não invalida e até confirma e facilita a confirmação das principais hipóteses levantadas, como a de que as tarefas de recuperação, categorização e extracção devem ser integradas e direccionadas a grupos de classes de páginas;

* Além do mais, a maior parte das páginas que já estão publicadas dificilmente será alterada, ou sê-lo-ão dentro de um tempo suficiente para manter um mercado propício exclusivamente aos agentes que processam ontologias dentro de páginas.

Vários editores de ontologias estão a surgir, com o objectivo não só de facilitar a sua construção, como também de disponibilizar ontologias públicas para reutilização e extensão, integrando diferentes grupos de pesquisa, frequentemente distantes geograficamente, que pesquisam sobre as mesmas áreas ou áreas afins.

A concepção de ontologias deve ser conduzida como qualquer outro projecto de software, no sentido de serem tomadas decisões de projecto que determinam a sua qualidade baseada em critérios como eficiência, legibilidade, portabilidade, extensibilidade, interoperabilidade e reutilização. Por isso, deve basear-se no seu futuro emprego, e não em aspectos filosóficos do conhecimento acerca do domínio representado. Alguns princípios, se usados com precisão, garantem a sua qualidade:

* Clareza: Os programas usam diferentes modelos e abstracções na resolução dos seus problemas. Na definição do conhecimento deve-se ter a objectividade de definir apenas o que se presume ser útil na resolução da classe de problemas a ser atingida. Definições completas, com condições necessárias e suficientes, devem ter precedência sobre definições parciais.

* Legibilidade: As definições devem ter correspondência com as definições correntes e informais. A ontologia deve usar um vocabulário partilhável - normalmente a terminologia usada por especialistas do domínio.

* Coerência: As inferências derivadas da ontologia definida devem ser correctas e consistentes do ponto de vista formal e informal com as definições.

* Extensibilidade: A ontologia deve permitir extensões e especializações com coerência, sem a necessidade de revisão de teoria, que consiste na revisão lógica automática de uma base de conhecimento em busca de contradições.

* Mínima codificação: Devem ser especificados conceitos genéricos - essa generalidade limitada pela clareza - independente de padrões estabelecidos para mensuração, notação e codificação, garantindo a extensibilidade.

* Mínimo compromisso ontológico: Para maximizar a reutilização, apenas o conhecimento essencial deve ser incluído, gerando a menor teoria possível acerca de cada conceito, permitindo a criação de conceitos novos e mais especializados.

3.5.2 Agente Inteligente em Ambiente Virtual Adaptativo

O agente tem por objectivos: auxiliar na navegação pelo ambiente; fornecer ajuda na localização de informações de interesse; auxiliar o especialista do domínio na organização espacial das informações. A adaptação do ambiente está relacionada com as possibilidades de reorganização do mesmo (conforme inserção, remoção ou actualização das informações) e de personalização da apresentação dos conteúdos, conforme o perfil do utilizador.

O ambiente consiste num mundo 3D, disponibilizado através da Web, onde dois tipos de utilizadores interagem: o requerente e o fornecedor da informação. O requerente, representado por um avatar, poderá navegar, visualizar informações e interagir com o agente. O fornecedor, responsável pelos conteúdos a serem disponibilizados, poderá explorá-lo e interagir com o agente. Durante a interacção com o requerente, o

agente fornecerá auxílio à navegação e à localização de informações de interesse e, durante a interacção com o fornecedor, ajudará na disposição espacial dos conteúdos no ambiente. Nesta disposição, informações e conteúdos referentes a um mesmo domínio deverão ser colocados fisicamente próximos.

Conforme o perfil do requerente, é feita a personalização da apresentação das informações e da estrutura do ambiente. O perfil contém informações sobre os interesses, as preferências e os comportamentos do requerente e é preenchido utilizando as seguintes abordagens: aplicação de questionários; observação da navegação; e verificação da interacção com o agente. A primeira abordagem é adoptada para a aquisição do perfil inicial, sendo as outras usadas para a actualização deste perfil. A monitorização da navegação e o recebimento das informações extraídas pelo agente, durante a interacção com o requerente são feitas através de sensores. Um módulo, gestor de perfil, é o responsável pela inicialização e actualização dos perfis, a partir das informações transmitidas pelo requerente e sensores, respectivamente.

Um perfil do fornecedor de conteúdos também é mantido, com informações sobre a(s) sua (s) área(s) de interesse, obtidas a partir de questionários e utilizadas para a apresentação adaptada dos conteúdos no ambiente. Este perfil também é gerido pelo módulo gestor de perfil. Os conteúdos serão geridos pelo módulo gestor de conteúdo e mantidos numa base de conteúdos. Cada conteúdo contém um perfil com informações sobre o domínio ao qual pertence, tipo de formato, entre outras. O fornecedor actua na definição deste perfil, armazenado na base de perfis de conteúdo. A representação das informações é feita através de componentes 3D, tais como objectos gráficos, ícones e hyperlinks para os conteúdos. Um conjunto de arquivos em VRML, correspondendo à definição de estruturas e objectos 3D, será mantido numa base de dados e utilizado na construção dos ambientes.

Nas interacções entre agente-requerente e agente-fornecedor, o ambiente é adaptado conforme o perfil do utilizador e conteúdos envolvidos. Para isso, um módulo, gerador de ambiente, é responsável pela geração de diferentes ambientes. Além disso, este módulo transmite ao agente as informações pertinentes aos perfis e aos conteúdos apresentados, de forma que este possua informações suficientes para a interacção com os utilizadores. Tais informações constituirão o conhecimento inicial que o agente possuirá do utilizador e do ambiente onde deverá actuar.

O agente possui as seguintes características: percepção, capacidade social, conhecimento e adaptação. A percepção contempla as observações do agente durante a interacção com os utilizadores e a sua capacidade social está relacionada à comunicação com os mesmos. O conhecimento do agente estará representado pelas informações que possui sobre o utilizador e o ambiente, o qual poderá ser actualizado durante a interacção.

Quanto à adaptação, o agente é capaz de aprender sobre o utilizador e adaptar-se a partir do que aprendeu.

O conhecimento do agente, armazenado numa base de conhecimento, será obtido a partir de duas fontes de informação: fonte externa e percepção da interacção com o utilizador. A fonte externa contemplará as informações sobre o ambiente e o utilizador, e serão provenientes do módulo gerador de ambientes. A observação da interacção com o utilizador será realizada pelo módulo de percepção, e as informações obtidas desta observação utilizadas na actualização do conhecimento do agente. É através do módulo de percepção que o agente detectará as solicitações de auxílio à navegação, localização e disposição de informações, vindas dos utilizadores. Com base na sua percepção e no conhecimento que possui, o agente decidirá como agir no ambiente. O módulo de decisão será o responsável por esta actividade. As decisões tomadas são transmitidas ao módulo de acção, responsável pela execução das decisões indicadas pelo módulo de decisão e pela manipulação da interface gráfica do agente.

Um agente inteligente para um ambiente virtual adaptativo, tem por funções: auxiliar o requerente na navegação pelo ambiente e na recuperação de informações, e ajudar o fornecedor na disposição espacial dos conteúdos.

3.6 Manipulação Integrada de Informação da Web

É necessário promover a cooperação, tanto entre as tarefas de extracção, categorização e recuperação, como entre agentes de um grupo de classes de páginas, de forma a que a cooperação resulte e se torne compensatória para o processamento das classes de páginas. Em ambos os casos, o factor precisão determinará a consistência da cooperação, se os relacionamentos entre as classes se mostrarem úteis - ou, por outras palavras, se as recomendações dos agentes a outros agentes exibirem uma precisão mais alta do que os resultados das consultas aos mecanismos de

busca, contendo menos lixo e facilitando a procura de páginas-membro das classes dos grupos procurados.

Também a hipótese dos grupos funcionais pode revelar-se preciosa nesse sentido: a classe das listas, por exemplo, se identificada apropriadamente, conduz directamente a um grande número de páginas-membro das classes, confirmando as hipóteses da abordagem de extracção integrada e de complementaridade entre as tarefas de extracção, categorização e recuperação e a de cooperação entre agentes distintos dentro de um mesmo grupo.

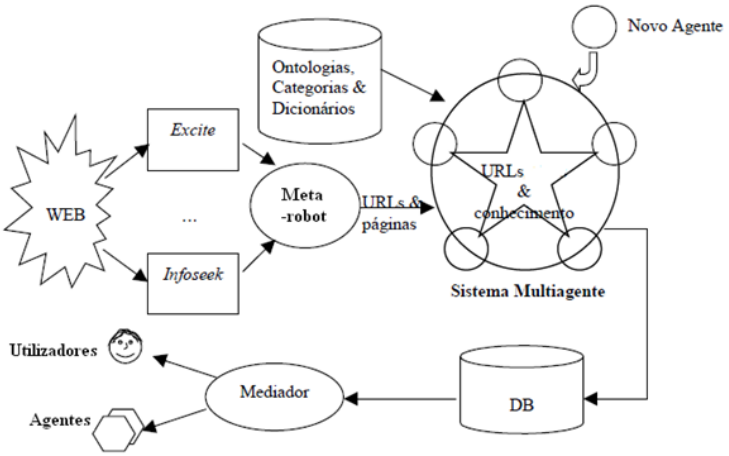

Figura: 3.5: Arquitectura de um sistema multiagente cognitivo para extracção integrada de dados da Internet.

Cada agente é um especialista no reconhecimento de páginas que correspondem a instâncias da classe que ele processa (por exemplo, páginas de pesquisadores, chamadas de eventos científicos - 'Call for Papers' -, artigos e outras do grupo científico) e na extracção de atributos dessa entidade (por exemplo, áreas de pesquisa e instituição de investigadores), procurando também identificar páginas e ponteiros úteis a outros agentes. Uma vez que os agentes possuirão responsabilidades distintas, praticamente sem intersecção, cooperando uns com os outros pela troca de sugestões de endereços de páginas candidatas a membros das classes que processam, a arquitectura baseia-se na abordagem de Resolução Distribuída de Problemas (RDP).

A granularidade da classe de páginas a ser processada por um agente,

ou seja, se um agente deve processar a classe Documento-Científico ou Publicação-Parte, depende da similaridade entre os padrões de suas classes. Se os padrões diferem muito, com certeza uma solução mais eficaz é separar as subclasses entre agentes distintos.

A qualquer momento, um novo agente pode ser introduzido no sistema, e os agentes já existentes trocarão mensagens com ele, de forma a estabelecerem uma cooperação no desempenho das suas tarefas individuais (maiores detalhes na secção seguinte). Utilizadores e agentes externos podem aceder aos dados extraídos pelos agentes através de um mediador.

3.6.1 Meta Robot

Uma questão relevante no desenvolvimento de sistemas para a Internet consiste na construção de robots de recolha. Todos os mecanismos de busca usam robots para recolherem os documentos e os representarem com palavras-chave e as suas respectivas frequências em bases de índices, que servem para responder aos utilizadores quais os documentos mais relevantes para as suas consultas. A proliferação destes robots ameaçava inviabilizar o tráfego na rede e sobrecarregar os servidores durante a recolha das páginas para os índices. Para sanar o problema, foram criadas convenções para robots 'bem-comportados', que, entre outras características:

* Aproveitam índices de outros mecanismos de busca e serviços de outros robots, evitando redundância de esforços,

* Alternam vários servidores, evitando sobrecarga,

* Processam apenas os dados que interessam (tipos de arquivos, data dos arquivos, etc),

* Sabem fugir de loops na procura de ponteiros e de ponteiros repetidos,

* Possam ser controlados interactivamente,

* Mantenham um log ou base de dados pública com estatísticas de sucesso, facilitando ao utilizador a escolha do robot ou mecanismo de busca adequado à sua aplicação.

Apesar de parecer na figura que há apenas um meta-robot na arqui-

tectura, na verdade cada agente possui o seu meta-robot. Um meta-robot é um robot que pode conectar-se a múltiplos mecanismos de busca - como Altavista, Excite, Infoseek e outros - aproveitando os seus índices, uma vez que não é necessário, para a recuperação de página de uma classe, indexar ou percorrer toda a Web. O meta-robot proposto segue todas as directrizes enumeradas acima.

Ele funciona da seguinte forma: efectuam-se consultas aos mecanismos de busca com palavras-chave que garantam cobertura das páginas retornadas em relação à classe de páginas processada pelo agente. Os termos 'call for papers' e 'call for participation' asseguram a cobertura das páginas de chamadas a eventos científicos para o agente 'Call for Papers', por exemplo. Devido à falta de precisão, o conjunto de páginas resultante das consultas recai em vários grupos funcionais além do grupo de páginas-conteúdo associado ao meta-robot, apresentando muitas listas, mensagens, páginas-conteúdo de outras classes, e lixo.

O uso do meta-robot pode ser encarado como o disparo de actividade de um agente, já que os agentes devem dar prioridade ao processamento das 'dicas quentes' enviadas pelos outros agentes, páginas que, presumivelmente, por resultarem dos relacionamentos entre as classes ou de outras páginas reconhecidas pelo próprio agente como listas, supostamente devem apresentar uma precisão significativamente maior em relação à classe processada. Portanto, cada agente continuamente irá aceder a duas filas de URLs: o conjunto sugerido pelo meta-robot é colocado numa fila de baixa prioridade de processamento, enquanto os 'palpites' de outros agentes e listas ficam numa fila de alta prioridade.

Deve-se salientar que o meta-robot foi projectado de forma parametrizada, permitindo a inclusão de novos mecanismos de busca como registos numa tabela da base de dados, abstraindo alterações de código ou recompilações. A sintaxe aproximadamente padronizada como os mecanismos de busca chamam as suas consultas através de um comando CGI e dispõem as URLs na página foi descoberta no melhor estilo RUDE, e propiciou a criação do primeiro extractor do sistema, retirando das páginas de resposta dos mecanismos de busca apenas as URLs resultantes.

3.6.2 Mediação

Na medida em que os sistemas e servidores crescem, uma gama diversa e heterogénea de bases de dados e sistemas passa a incorporar um agregado, com dados e funcionalidades potencialmente acessíveis aos utilizadores (de agora em diante, utilizador designa pessoas, agentes ou outras entidades de software). Tanto quanto possível, os utilizadores devem ser poupados ao conhecimento de detalhes de acesso e funcionamento, como linguagens de base de dados, parâmetros, entre outros. Um mediador propõe-se justamente a fornecer uma fácil interacção entre os utilizadores com os serviços de um sistema ou servidor, actuando como relações públicas dos serviços junto aos utilizadores. A sua existência pode ser justificada de várias maneiras:

- Apenas parte do conjunto de informações e funcionalidades é relevante para os utilizadores. Devido à normalização, a estrutura de bases de dados, por exemplo, tornaria as consultas difíceis e complexas para o utilizador mediano;
- Os dados em bases de dados distintas muitas vezes possuem formatos distintos, dificultando a sua integração;
- Há ainda serviços e sistemas que possuem funções similares, e os seus utilizadores, para abrir mão deles, têm dificuldade de distinguir qual deles se irá adequar melhor à sua tarefa, e também de formular pedidos com o grau de granularidade correcto;
- Integrar cada novo servidor consome muito tempo, e não fornece a flexibilidade e legibilidade desejadas;
- Um requisito fundamental para facilitar o uso automático destes sistemas ou agentes é que os sistemas ou agentes utilizadores possam aceder facilmente a informações relativas à semântica das bases de dados e dos serviços disponíveis, necessitando, portanto, de entidades de software ou agentes que desempenhem adequadamente as tarefas de disponibilizar esta semântica e proceder à comunicação sobre elas com segurança.

Actualmente, pesquisas em duas classes de sistemas suportam os serviços de mediação:

- Facilitadores, que implementam a transparência de localização em programação baseada em componentes ou agentes, i. e., um serviço de nomes e rotação de mensagens e;
- Mediadores, que podem ser vistos como camadas de middleware que fornecem serviços intermediários, tentando tratar a mensagem recebida, entender e processar o pedido.

A mediação pode reunir entidades múltiplas e heterogéneas, como sistemas, agentes, componentes, bases de dados, bases de conhecimento, sites, documentos, bibliotecas digitais e sistemas geográficos entre outros. Especialmente em bases de dados, pode proporcionar uma semântica ao conteúdo das informações contidas nos esquemas, já que esses esquemas trazem apenas uma descrição lógica das informações que guardam, e não conhecimento sobre elas, sem a possibilidade de inferência ou raciocínio. Existem pesquisas sobre bases de dados dedutivas, mas ainda está longe a existência de um padrão de sintaxe para SQL que incorpore essa facilidade. São tarefas típicas de mediação ao utilizador:

* Filtragem e extracção de informação, entregando ao utilizador apenas o que lhe interessa;

* Transformação da informação para um formato compreensível para o utilizador - este formato pode ser da estrutura física da informação ou, no caso de agentes, de uma ontologia para outra;

* Combinação ou integração de informação;

* Transmissão (brokering), quando mediadores começam a conversar com outros mediadores, o que, aliás, confere transparência de localização à informação;

* Notificação, isto é, aviso ao utilizador que há informação que lhe pode ser útil, entregando-a sob sua concordância.

A existência de mediadores adiciona valor aos serviços fornecidos, actuando como uma recepção e interface amigável dentro de uma arquitectura que reúne esses serviços, refinando os pedidos para o grau de granularidade adequado, traduzindo-os e optimizando-os. Os mediadores podem ainda executar tarefas ligadas à segurança e permissões, modelar o cliente de forma a identificar recursos relevantes e interessantes, anunciar serviços numa rede para assinantes e potenciais utilizadores, e até planear a execução optimizada de funções típicas de agentes de

informação, como busca, recuperação de informação e outras.

Qualquer área de aplicação que necessite integrar informação heterogénea pode servir-se de mediadores. Entre as áreas já testadas com sucesso, incluem-se comércio electrónico, sistemas hospitalares, data warehousing, transporte aéreo, hotelaria, tráfego de estradas e sistemas corporativos.

3.6.3 Técnicas para a Recuperação de Informação na Internet

Para recuperar uma informação nas bases de dados da Internet o utilizador necessita utilizar algumas técnicas de pesquisa bem simples, que orientarão os seus passos e reduzirão o tempo de busca. Ao utilizar-se um mecanismo de busca na Internet, a pesquisa limita-se à base de dados daquela ferramenta. O sucesso da pesquisa depende da habilidade em encontrar o melhor mecanismo de busca para o objectivo pretendido e a capacidade de extrair todo seu potencial utilizando formas de refinamento.‘

O sucesso da busca depende do que o utilizador necessita e de saber qual é a informação que se quer encontrar. Assim, ele precisará direccionar melhor a sua pesquisa escolhendo formas de refinamento mais apropriadas. Para se encontrar o que se procura, há algumas técnicas:

1) URL (Uniform Resource Locator): por meio de um endereço de site que conhecemos e de que dispomos; conseguem-se estes sites através de revistas, amigos, jornais ou indicação de algum professor.

2) Por Assuntos Específicos: o utilizador utiliza nas ferramentas de busca palavras-chave (termos que identifiquem e definam a palavra) ou frases que caracterizem o que quer pesquisar.

3) Por Assuntos/Categoria: a busca nesta opção é feita por meio de tópicos que são indexados por categorias e subcategorias de assuntos.

Para ter um melhor resultado numa busca e conseguir obter informações de fiabilidade e consistência, o pesquisador deverá saber refinar a sua pesquisa de maneira que venha a utilizar alguns operadores os quais podem ser diferentes de ferramenta para ferramenta. Através desses operadores é possível definir o objecto de interesse e tornar assim a pesquisa mais eficiente;

O conhecimento desses operadores, aceites pelas diferentes ferramentas de busca, melhora a eficiência da pesquisa. É possível, determinando os termos para a busca, avaliar quais os operadores mais adequados e que, portanto, podem contribuir de facto para o sucesso da pesquisa.

Algumas ferramentas que trabalham com o sistema chamado lógica booleana que utiliza conceitos matemáticos e podem ser também alguns operadores de proximidade, operadores de existência e de exactidão. Os Operadores Lógicos ou Booleanos são utilizados para auxiliar ou filtrar uma pesquisa e podem ser demonstrados como: AND (e), NOT (não), OR (ou). Estes operadores ajudam a refinar a pesquisa, pois recuperam-se as fontes desejadas com mais rapidez e relevância.

Com a Internet o processo informativo (e comunicativo) tem tomado novos contornos, por esta razão é aconselhável prudência e maior investigação com as fontes de informação obtidas na Internet; o pesquisador deve ter certas precauções quanto à credibilidade das informações disponíveis na Web para não saírem prejudicados. Isto considerando os indivíduos que têm acesso às novas tecnologias, e que, de facto, utilizam este veículo para aumentar o seu stock mental e cognitivo.

Deve ser levado em consideração o seguinte: num delineamento da realidade contemporânea temos um sector da sociedade preocupado com a produção de informação e outro preocupado com a produção de coisas (efémeras e passageiras).

O ser humano que não tem o domínio completo/total do sistema será sempre controlado e manipulado, portanto não interage, não se conecta ou isso acontece de maneira distorcida ou precária. Do outro lado estão aqueles que conhecem, que compreendem, que actuam, que usam a máquina com eficácia, inteligência e competência, absorvendo quase toda sua complexidade. E como falta capacidade de apreensão de informação e de conhecimento por parte do outro elemento do jogo é óbvia a vitória do sistema.

A sociedade nesta hiper-modernidade passa, então, por uma possível fase confusa e paradoxal, diante de um excesso de informação, imagens, novidades tecnológicas e outras infinidades de imposição dos tempos pós-modernos, que deixam os indivíduos vacilantes entre o tradicional, o moderno, o inovador, o consumo, a individualidade, o ser e o ter. Numa espécie de paraíso perdido onde se procuram referências a todo o momento a respeito desta nova sociedade, deste novo homem e destas novas

formas sociais e comunicacionais que se alteram num tempo e espaço virtual de dimensão nula.

3.6.4 Principais Tarefas na Recuperação de Informação

A Internet e a sua fonte de informação desestruturada deram margem ao nascimento de outras tarefas ligadas à busca de informação relevante e útil, além da recuperação de informação.

Surgiram a tarefa de extracção e várias outras decorrentes da tarefa de classificação (ou categorização), a saber:

* Filtragem: No fundo, pode-se considerá-la como uma categorização de documentos em duas classes; a dos documentos seleccionados ou considerados relevantes, e a dos desprezados ou irrelevantes.

* Agrupamento (Clustering): Consiste em separar os documentos em grupos por similaridade, medindo-os e comparando-os matematicamente por um limiar. Neste caso, as classes ou grupos nos quais os documentos serão organizados não são pré-definidas. Esta tarefa possui várias utilidades; funciona, por exemplo, como base para a expansão de consultas para a busca de documentos similares e potencialmente relevantes em sistemas de recuperação. Serve também em sistemas de recomendação de itens a utilizadores, e ainda para a criação de hierarquias de classificação.

* Roteamento: A tarefa de notificação - ou seja, o aviso a um utilizador específico sobre uma nova informação - originou esta tarefa, que perfaz uma classificação de documentos em classes pré-definidas para que eles possam ser encaminhados aos utilizadores interessados nestas diversas classes, de acordo com perfis de utilizador preenchidos por eles de antemão.

A ideia de integrar as três tarefas possui antecedentes, e inspira-se na necessidade de um processamento adicional após a recuperação de documentos, para determinar a real relevância de seus conteúdos.

Considerando as três tarefas (recuperação, extracção e categorização), as diversas propostas de solução para problemas na área de recuperação de informação foram separadas em dois tipos de abordagens: as abordagens estatísticas e as baseadas em processamento de linguagem natural.

3.6.4.1 Abordagens Estatísticas

São aquelas que seguem sempre as seguintes premissas básicas:

* Os documentos (ou partes deles) são representados em vectores de palavras-chave e frequências, podendo ainda conter outras particularidades, como, por exemplo, ordem das palavras, que ajuda a identificar a presença de n-gramas (termos com n palavras-chave em sequência, como 'redes neurais sem peso').

* A representação não inclui palavras muito frequentes, que atrapalhariam a indexação, e são especificadas num dicionário próprio pelo qual são reconhecidas, chamado de stoplist.

* As palavras são reduzidas aos seus morfemas por algoritmos de stemming; por exemplo, engineer e engineering são reduzidas para engine.

* A tarefa a ser realizada (recuperação, categorização e/ou extracção) buscará palavras-chave nos vectores que representam os documentos, ordenando os documentos por frequência destas palavras-chave, opcionalmente utilizando outras palavras-chave - cuja probabilidade de estarem presentes nos documentos é alta - e/ou seus respectivos pesos ajustados pela relevância destas novas palavras-chave.

Portanto, a tarefa a ser realizada é efectuada através de um algoritmo matemático que actua sobre os vectores que representam os documentos, indicando como solução o conjunto ordenado de documentos que melhor atende à tarefa solicitada. A heurística matemática mais comum em sistemas de representação é conhecida como algoritmo TF-IDF (do inglês term frequency - inverse document frequency). O algoritmo atribui pesos a palavras-chave na representação de um documento de acordo com duas medidas:

* A frequência da palavra-chave, chamada na fórmula abaixo de TF(p), onde p é a palavra-chave,

* A frequência em documentos da palavra-chave, chamada sob a mesma convenção de DF(p), que significa o número de documentos no qual a palavra-chave p está presente.

A fórmula que representa o peso de uma palavra-chave num documento será dada por TFIDF(p) = TF(p) * log(d/ DF(p)), onde d é

o total de documentos do conjunto procurado. O algoritmo visa salientar palavras pouco frequentes em outros documentos, que, portanto, são mais representativas do documento representado.. Para uma consulta de busca com várias palavras-chave, por exemplo, são totalizados os pesos de cada uma em cada documento, e os documentos com peso maior que zero são ordenados de forma decrescente e apresentados como o conjunto resposta.

Os mais diversos tipos de algoritmos matemáticos, probabilísticos e de aprendizagem automática já foram testados em RI, com o objectivo de conferir aos sistemas vantagens que os peculiarizem, como adaptabilidade, usados por algoritmos que implementam aprendizagem, e busca por conceito, como o algoritmo LSI (do inglês Latent Semantic Indexing). Há abordagens das mais diversas, com algoritmos envolvendo trigonometria para verificação de similaridade de documentos, probabilidades de documentos serem relevantes em relação a consultas, lógica difusa, algoritmos genéticos, redes neurais, redes de crença e redes de inferência.

3.6.4.2 Abordagens Baseadas em Processamento de Linguagem Natural

As técnicas de Processamento de Linguagem Natural (PLN) são capazes de processar textos escritos num subconjunto de um idioma. Têm como característica um alto custo computacional, principalmente porque as várias fases empregam representações diferentes; devido a isto, soluções de PLN costumam ser empregues em tarefas de background, como categorização e extracção. Contudo, existem sistemas que as empregam de forma superficial, sem aplicar todas as tarefas e técnicas possíveis, visando optimizar a precisão de sistemas de recuperação com um tempo de resposta aceitável. Este tipo de solução foi aplicado com sucesso em bibliotecas digitais, onde o facto de os documentos guardarem semelhança estilística e temática constitui um factor preponderante.

3.6.5 Extracção de Informação

Extracção de informação ‘é a tarefa de identificar os fragmentos específicos de um documento que constituem o núcleo de seu conteúdo semântico‘. A exemplo dos sistemas de PLN e como qualquer aplicação

de Inteligência Artificial, aqui também é obedecido o requisito de restrição de domínios; ou seja, os extractores actuam apenas sobre classes de documentos que guardam entre si regularidade de formatação, estrutura e conteúdo. Os extractores processam textos em geral pequenos e ricos em dados. Existem basicamente dois tipos de sistemas de extracção: os sistemas baseados em processamento de linguagem natural e os wrappers. Ambos os tipos beneficiam da aprendizagem automática para garantir escalabilidade e portabilidade, proporcionando um desenvolvimento rápido de novos extractores pela automatização parcial ou total do processo de aquisição de conhecimento. Após serem detalhados os dois tipos, serão abordados os papéis, benefícios, vantagens e abordagens da aplicação de ontologias em sistemas de extracção.

Os extractores baseados em processamento de linguagem natural trabalham sobre textos escritos sobre um determinado assunto, como terrorismo, micro-electrónica e sucessão de cargos em empresas. Os textos possuem regularidade de estrutura e conteúdo, mas de não formatação, e muitos deles provém de uma mesma fonte. Integrantes de um mesmo domínio, os textos usados em extracção baseada em PLN são classificados entre as classes de semi-estruturados e desestruturados.

Um grande problema para qualquer sistema de PLN é o facto de que um facto pode ser expresso de várias maneiras. Por isso, a modelagem do domínio resulta na tarefa mais cuidadosa destes sistemas. A análise semântica, para a qual servirá esta modelagem, irá demonstrar resultados consistentes na medida em que:

1º. Forem escolhidas as partes adequadas de um dicionário semântico, como o WordNet, que se adaptem ao domínio, e;

2º. Sejam criados templates suficientes para instanciar as diversas formas de expressão dos factos que se deseja extrair.

Após a análise semântica, a análise do discurso ajuda a inferir a extracção de atributos, e também a minimizar problemas de co-referência, eliminando instâncias de templates que contêm factos inverosímeis. Extractores baseados em PLN representam o conhecimento explicitamente, facilitando qualquer processo de inferência que se siga durante ou após a extracção.

Outro tipo de extractores, os wrappers, foram projectados com o pressuposto de que existem documentos com muita informação útil, es-

pecialmente na Internet, estruturados ou semi-estruturados em forma de tabelas, registos, e que apresentam forte regularidade de formatação, estrutura e conteúdo. Os wrappers visam, portanto, preencher bases de dados com informações extraídas a partir de seus delimitadores, que podem ser sinais de pontuação ou tags em HTML.

Naturalmente, este tipo de extractor requer uma estrutura das páginas até certo ponto rígida, não conseguindo capturar o contexto com a flexibilidade desejada. Algoritmos de aprendizagem também se revelaram úteis aos wrappers, e são empregados no intuito de resgatar, compreender e explicitar a formatação das páginas. A aprendizagem ou indução de wrappers [Kushmerick, 1997] possui a vantagem de se adaptar imediatamente a diferentes formatos de documentos, exigindo apenas a anotação das páginas que servirão como exemplos para o algoritmo de aprendizagem.

3.6.6 Mineração de Conteúdo na Web

Embora a principal área de aplicação das técnicas de mineração de dados ainda seja na área comercial, muitas outras áreas que também lidam com grandes volumes de dados vêm cada vez mais utilizando essas mesmas estratégias para outras aplicações. A Web, com o seu enorme volume de recursos e utilizadores é naturalmente uma fonte de dados para um grande número de aplicações.

Há duas abordagens que são usualmente utilizadas para obter os dados que alimentam o processo de mineração de dados em aplicações Web. Algumas das soluções desenvolvidas obtêm dados interferindo directamente sobre os recursos, acrescentando pequenos programas que capturam e armazenam informações de uso. Outras procuram interferir minimamente com os conteúdos, geralmente aproveitando os registos de uso (logs) presentes nos servidores Web.

Considerando os tipos de dados que estão presentes na Web, as quatro principais áreas de aplicação das técnicas de mineração de dados para a Web são:

· Mineração de conteúdo, usada para descobrir informações diversas a partir do conteúdo das páginas Web, principalmente (mas não limitada a textos e gráficos).

· Mineração estrutural, que analisa a organização do conteúdo. Pode ser intra-página, quando utiliza a informação da estrutura interna do documento (por meio da análise de rótulos HTML ou XML) ou inter-páginas, quando foca os hyperlinks e na forma como se conectam diversas páginas.

· Mineração de uso, que observa a exploração de dados gerados pelos acessos dos utilizadores aos recursos, tais como origem da solicitação, referências solicitadas e momento dos acessos.

· Mineração de perfil de utilizador, que explora informações demográficas sobre os utilizadores da Web por meio de dados de registos e informações de perfis disponibilizadas pelos utilizadores.

3.6.7 Mineração de Dados em PDS (Processo de Desenvolvimento de Software)

As técnicas de mineração preditivas [Tan, 2005], como a classificação e regressão, têm sido empregues no contexto de PDS com o intuito de facilitar estimativas e planeamento de projectos, estabelecer causas de problemas e prever falhas em módulos de software [Nayak, 2001] [Khoshgoftaar, 2001] [Nagappan, 2005]. Os trabalhos relacionados apresentam a aplicação de técnicas de mineração em dados de PDS para descoberta de conhecimento em bases de dados de projectos de software. Contudo, o número de trabalhos que apresentam mineração sobre dados de projectos e produtos de software, ainda, é muito pequeno. As principais limitações encontradas pelos pesquisadores estão relacionadas ao baixo volume de dados, espaçamento entre os dados e, consequentemente, falta de regularidade nos mesmos.

A classificação é uma técnica de mineração de dados capaz de atribuir objectos a uma entre várias categorias pré-definidas. Os modelos de classificação reflectem como os atributos preditivos se relacionam para determinar o atributo alvo. Através da classificação, é possível analisar dados e extrair modelos que descrevam classes e predigam tendências futuras em novos dados. Entre as técnicas de classificação mais utilizadas pode-se citar: Árvore de Decisão e Rede Bayesiana. A Árvore de Decisão é um gráfico de fluxo em estrutura de árvore, onde cada nó interno denota um atributo de teste, também chamado de preditivo, e cada aresta representa um resultado para o teste. Os nós (ou nodos) 'folhas'

representam classes alvo. Uma das vantagens da técnica da Árvore de Decisão é a facilidade de interpretação dos modelos gerados.

A técnica de Rede Bayesiana gera modelos gráficos que representam o relacionamento probabilístico de um conjunto de variáveis [Hec, 1997a]. Os dois elementos básicos de uma Rede Bayesiana são:

- Grafo acíclico orientado que representa os relacionamentos de dependência entre um conjunto de variáveis;
- Tabela de probabilidades associada a cada nó e o seu nó imediatamente relacionado.

Assim, uma Rede Bayesiana é composta de uma parte quantitativa (tabelas de probabilidade condicional) e outra qualitativa (o grafo acíclico orientado). Cada nó da rede representa um atributo do domínio e os arcos representam as relações de dependência entre dois nós. A probabilidade entre dois nós da rede representa a força do relacionamento causal entre eles. Desta forma, o relacionamento entre as variáveis do domínio é explorado através de probabilidades condicionais.

3.7 Navegação usando Redes Bayesianas

Num ambiente de hipertexto projectado, normalmente existem dois 'actores': o autor, projectista do sítio (site) e o leitor, representando o conjunto de indivíduos que interagem com o ambiente virtual apresentado. Neste contexto, páginas, links, possibilidades de interacção e estrutura do ambiente, são de responsabilidade do autor. O conteúdo e estrutura resultante reflectem a opinião dos seus autores, que projectaram o sítio, direccionando-o para um perfil de sujeitos.

A decisão do que constitui uma boa hiper-estrutura depende de muitos factores: domínio da aplicação, familiaridade do leitor com a informação apresentada, se a informação é lida sequencialmente ou se há busca por uma informação em particular. O problema da estruturação do sítio ocorre sistematicamente, pois o único método para fazê-lo são as contribuições individuais dos seus autores, cada um adicionando o seu próprio conhecimento e estruturando, sob o seu 'ponto de vista', as vinculações entre os conteúdos.

O princípio da inteligência colectiva, aqui aplicado na Web, parte do pressuposto que os links escolhidos pelos indivíduos durante uma sessão representam um conhecimento colectivo implícito. Este conhecimento pode ser capturado e utilizado na auto-organização da estrutura de links de um sítio. A hipótese aqui é que a 'soma' dos conhecimentos parciais de cada indivíduo representa, no seu conjunto, um conhecimento mais abrangente e superior ao do mentor intelectual do sítio. Portanto, se bem aproveitado, este conhecimento colectivo possibilitará uma estrutura de navegação melhor do que a estrutura originalmente concebida pelo autor do sítio.

É utilizada a hipótese de que o 'comportamento navegacional' de um indivíduo, durante a sua interacção com um ambiente na Web, fornece subsídios suficientes para um sistema gerar um modelo deste sujeito. Para isto, está sendo proposta uma topologia de rede Bayesiana com a capacidade de recomendar 'caminhos mais adequados' de navegação.

O 'comportamento navegacional' é aqui compreendido como: a) os caminhos percorridos pelo indivíduo no sítio e; b) o tempo de permanência em cada página acedida.

A partir destas duas variáveis propõe-se uma função geradora de evidências para a rede Bayesiana, que então criará a estrutura navegacional de acordo com o 'perfil cognitivo' deste indivíduo.

Entende-se como 'caminhos mais adequados' uma estrutura de links de hipertexto que facilite a pesquisa do sujeito, considerando o seu conhecimento prévio do assunto e respeitando os seus desejos e aspirações.

Pressupõe-se que o modelo proposto possa contribuir para a transformação da Web, como nós a conhecemos, numa rede com eficácia associativa, sendo capaz de absorver o conhecimento implícito dos seus utilizadores e descobrindo novas relações entre peças de informação.

3.7.1 Sistema Adaptativo

Um sistema adaptativo consiste de um hipertexto e um modelo de utilizador, sendo adaptativo no sentido que adapta o hipertexto usando o modelo de utilizador. Denomina-se sistema de Hipermédia Adaptativa a 'todo o sistema de hipertexto e/ou hipermédia que reflicta algumas características dos seus diferentes utilizadores em modelos e aplique tais

modelos na adaptação de diversos aspectos visíveis do sistema às necessidades e desejos de cada utilizador' [Brusilovsky, 1998].

A adaptação poderá ocorrer ao nível de estrutura do sítio (links são criados, eliminados ou modificados) e/ou ao nível de conteúdo. Será considerado um hipertexto adaptativo se as adaptações ocorrerem dinamicamente sem a necessidade de interferência consciente dos seus utilizadores.

3.7.2 Probabilidade Bayesiana

A abordagem clássica da probabilidade supõe que as probabilidades são inerentes à natureza física do mundo. Por exemplo, ao lançar uma moeda os valores da probabilidade de que caia cara ou coroa são valores inerentes às propriedades físicas da moeda. Sob esta interpretação, as probabilidades são chamadas 'frequentistas' e com base em experimentos pode-se estimar estas probabilidades.

Como alternativa, as probabilidades Bayesianas consideram as probabilidades como subjectivas e associadas ao conhecimento de cada indivíduo. A probabilidade de um evento é, sob a abordagem Bayesiana, um grau de crença na probabilidade de que o evento ocorrerá, sob o ponto de vista de algum individuo. Uma vantagem da probabilidade Bayesiana é que não é necessário associar experimentos para estimar a probabilidade associada a eventos.

3.7.3 Redes Bayesianas

As Redes Bayesianas são tipos específicos de redes de conhecimento. A ideia principal é que, para descrever um modelo do mundo real, não é necessário usar uma enorme tabela de probabilidades conjuntas na qual são listadas as probabilidades de todas as combinações possíveis de eventos. A maioria dos eventos é condicionalmente independente da maioria dos outros, portanto suas interacções não precisam de ser consideradas. Em vez disso, usa-se uma representação mais local, que descreve agrupamentos de eventos que interagem.

As redes Bayesianas foram desenvolvidas nos anos 70 com o objectivo de modelar o processamento distribuído na compreensão da leitura, onde as expectativas semânticas e evidências perceptivas deveriam ser

combinadas para formar uma interpretação coerente. A habilidade para coordenar inferências bidireccionais preencheu uma lacuna na tecnologia de sistemas especialistas no início dos anos 80, e as redes Bayesianas têm emergido como um esquema de representação genérico para conhecimento incerto .

Uma rede bayesiana é um Grafo Direccionado Acíclico (DAG) onde os nós representam as variáveis (de interesse) de um domínio e os arcos representam a dependência condicional ou informativa entre as variáveis. A força da dependência é representada por probabilidades condicionais que são associadas a cada cluster de nós pais-filhos na rede. Para se representar que um facto X é independente de um outro facto Y, caso um terceiro facto Z seja conhecido, todos os caminhos do grafo que ligam X e Y devem passar por Z:

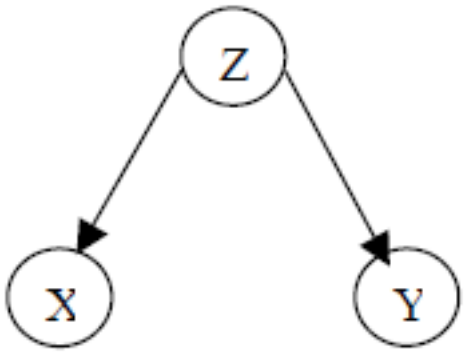

Figura: 3.6: Base Bayesiana

Assim, se a variável Z é conhecida, então X e Y são independentes para um dado Z. Isto possibilita a redução dos parâmetros numéricos das condicionadas que fazem parte da distribuição em questão. O conhecimento causa/efeito de um domínio permite estruturar suposições de independência condicional. Suponha que os grafos com que se esteja a trabalhar sejam fracamente ligados. Isto significa que ao remover qualquer arco do grafo este se divide em dois.

O tempo de permanência em cada página e o comportamento de navegação dos utilizadores constituirão evidências para a rede Bayesiana subjacente. Diante das evidências fornecidas pelo estilo de navegação dos

educandos, a rede Bayesiana actualizará as suas crenças. O resultado esperado é uma estrutura de hiperlinks adequada ao conhecimento e interesse dos utilizadores.

Especificamente, será elaborado um sítio onde à medida que os utilizadores navegarem por ele haverá uma monitorização das suas acções e essas acções serão parâmetros de entrada para uma rede Bayesiana. Esta rede gerará uma lista de links com probabilidades associadas, permitindo a adaptação dinâmica dos links a cada passo de navegação. O comportamento navegacional dos visitantes fornecerá valores para algoritmos específicos que converterão esses valores em evidências para a rede Bayesiana implementada no servidor.

As redes Bayesianas são formadas - em relação às variáveis que a compõem - por uma parte estrutural (qualitativa) e por uma parte quantitativa. A parte qualitativa é um modelo gráfico (grafo acíclico direccionado) onde as variáveis são os nós. Além dos nós, a parte qualitativa inclui as regras, que são as relações de dependência entre variáveis, representadas pelos arcos direccionados. A parte quantitativa de uma rede Bayesiana é o conjunto de probabilidades condicionais associadas aos arcos existentes no modelo gráfico e as probabilidades estimadas a priori das hipóteses diagnosticadas. Após definida a topologia da rede, basta especificar as probabilidades condicionais para os 'nodos' que possuem dependências directas e utilizá-las para processar qualquer outro valor de probabilidade.

Mineração de dados e o processo de descoberta de conhecimento em bases de dados baseados na tecnologia de mineração de dados em redes Bayesianas são uma das áreas de pesquisa da inteligência artificial mais requisitadas nas últimas décadas, com um grande número de aplicações. As redes Bayesianas oferecem uma estrutura unificada e intuitiva, onde é possível comparar diferentes hipóteses, de acordo com os nós da rede, tornando-as um dos melhores métodos analíticos para a tomada de decisão.

Dessa forma, a fim de minimizar o tempo gasto no processo de aquisição de conhecimento, a mineração de dados para construção de redes Bayesianas é capaz de estimar os valores das probabilidades e também identificar os nós da rede a partir de bases de dados, tornando esse processo mais rápido e provavelmente mais eficiente.

Para que a estrutura do sítio seja dinâmica, utilizam-se mecanismos

de monitorização. O comportamento monitorizado será o tempo que o sujeito permanecerá em determinada página e também o mapa da navegação percorrida. Essas informações, uma vez analisadas, servirão como base para a actualização das conexões entre os links, provendo a adaptação automática da estrutura de links das páginas.

As acções do utilizador originarão um modelo do seu comportamento navegacional, e este modelo será construído de forma implícita, o que significa que o educando não necessitará em nenhum momento de responder questões de teste ou fazer qualquer outra intervenção directa na construção deste modelo. Isto fará com que a interacção do utilizador com o sítio seja absolutamente tradicional. O utilizador não precisará de nenhum treino ou conhecimento extra e/ou específico. As adaptações na organização e apresentação dos links ocorrerão automaticamente, oferecendo ou limitando opções de navegação.

O processo de descoberta de conhecimento em redes bayesianas utiliza a mesma metodologia do KDD clássico, sendo composto também de três etapas responsáveis pelo pré-processamento, a mineração de dados propriamente dita e o pós-processamento.

Figura: 3.7: Processo de KDD para mineração de dados em redes Bayesianas.

Após a escolha de uma base de dados num domínio de aplicação, o processo de KDD inicia-se com a etapa do pré-processamento, onde são aplicados alguns procedimentos, entre eles, tratamento de valores ausentes, repetidos; redução do volume de dados; selecção dos atributos principais entre os existentes na base; limpeza e integração.

A etapa seguinte, chamada de mineração de dados é a principal do KDD, e em redes bayesianas, utiliza algum mecanismo para a construção automatizada da rede pela aprendizagem das suas estruturas e probabilidades, a partir da base de dados tratada na etapa anterior.

Por fim, a etapa do pós-processamento, segundo [Goldschimidt, 2005],

contempla a avaliação, análise e selecção do conhecimento extraído. Dessa forma, a rede bayesiana, gerada na fase de mineração de dados, pode ser analisada pelo especialista e, caso a aquisição de conhecimento tenha sido realizada de maneira correcta, a rede bayesiana poderá ser utilizada na construção de um sistema especialista probabilístico.

Redes bayesianas constituem uma forma natural para representação de informações condicionalmente independentes. Além de mais, possibilitam a representação compacta de uma tabela de junção de probabilidades. Por outras palavras, redes bayesinas oferecem uma boa solução a problemas onde conclusões não podem ser obtidas apenas do domínio do problema, onde o uso de probabilidades é exigido.

Inferências sobre redes bayesinas podem ser executadas em tempo linear, porém, para a maioria das topologias de rede, inferências possuem complexidade NP-hard. Algumas técnicas podem ser aplicadas para se obter tempo linear, mesmo em topologias que impossibilitariam este facto. Entretanto, este ainda é um dos grandes desafios ao desenvolver-se uma rede.

3.8 Sistemas de Recuperação de Informação

Os Sistemas de Recuperação de Informação têm como objectivo a realização das tarefas de indexação, busca e classificação de documentos (expressos na forma textual), a fim de satisfazer a necessidade de informação do indivíduo, geralmente expressa através de consultas. A necessidade de informação pode ser entendida como a busca de respostas para determinadas questões a serem resolvidas, a recuperação de documentos que tratam sobre determinado assunto ou ainda o relacionamento entre assuntos.

Hoje em dia, a localização de documentos através de engenhos de busca, é feita, geralmente, com a utilização de buscas por palavras-chave ou expressões contidas nos documentos. O sucesso em encontrar documentos relevantes depende do casamento dos termos fornecidos pelo utilizador numa consulta, com os utilizados como índices na indexação da base de dados de documentos.

Com o crescimento das colecções de documentos digitais, os sistemas de recuperação de informação que localizam documentos utilizando

buscas por palavras-chave e expressões simples têm-se tornado cada vez menos eficientes. Este insucesso está relacionado com os seguintes motivos: a dificuldade do utilizador em expressar o que realmente procura através de uma consulta; a forma desorganizada como os documentos resultantes da busca são mostrados; o número excessivo de documentos devolvidos.

Com a vasta quantidade e variedade de documentos disponíveis, formular uma consulta efectiva para uma busca é uma tarefa difícil, e examinar uma lista resultante de uma pesquisa onde os itens são muitos e estão ordenados de forma claramente não significativa pode ser tediosa. Assim, tornam-se necessários métodos que sejam capazes de realizar uma organização automática dos documentos em conjuntos, evidenciando o relacionamento entre os conteúdos desses documentos, e as relações de proximidade entre os conjuntos de documentos de forma visual.

Existem, na Web, classes de páginas com conteúdo e estrutura similar (por exemplo, páginas de chamadas de trabalhos, referências bibliográficas, etc); algumas delas têm sido tratadas por agentes extractores. Porém, estes sistemas negligenciam o facto de que algumas destas classes se inter-relacionam, formando grupos (por exemplo, o meio científico).

Torna-se necessária uma organização ou arquitectura de sistemas multiagentes cognitivos para a recuperação, classificação e extracção integradas de informação, a partir destes grupos. Para a realização destas tarefas, uma visão da Web que incorpora estas classes (visão por conteúdo) e também a funcionalidade de apresentação das informações mantidas nas páginas torna-se necessária. Cada agente processa uma classe, empregando ontologias do domínio e ontologias estratégicas para reconhecer páginas e extrair delas as possíveis informações úteis, comunicando entre si e cooperando com os outros agentes.

As indicações sobre páginas e links, trocadas entre os agentes, normalmente contêm menos lixo do que os resultados das consultas dos mecanismos de buscas tradicionais (por exemplo, Google, AltaVista e Excite). A arquitectura de agente apresenta várias formas de reutilização: código, esquema da base de dados, conhecimento e serviços dos mecanismos de busca. Resultados promissores da recuperação e classificação funcional e de conteúdo foram obtidos para agentes que processam eventos e artigos científicos, empregando uma ontologia do domínio científico criada especificamente para este fim, sugerindo que a arquitectura

é realizável.

A problemática referente á manipulação de informação em grandes redes como a Internet colocou questões de difícil solução para tornar fácil o acesso à grande fonte de informação disponibilizada pelos utilizadores. Áreas como recuperação de informação, agentes inteligentes, ontologias, classificação e extracção, e modelagens da Web subitamente incluíram-se entre os temas de maior pesquisa no campo da informática. A pesquisa nestas áreas tem, continuamente, tentado fornecer soluções adequadas, mas ainda se encontram em fase de maturação e longe de soluções gerais de alta performance.

Diversos factos devem ser considerados para uma maior capacidade de mineração de dados na Internet:

*As tarefas de recuperação, categorização e extracção podem e devem ser integradas, o que pode ocasionar uma melhoria de performance em todas elas.

*Formulação do problema da manipulação integrada de informação.

*Formulação de uma visão da Web combinando conteúdo e funcionalidade, que facilitem a resolução do problema.

*É necessário que o domínio que abarca as classes de páginas interrelacionadas, esteja representado num formalismo lógico de representação de conhecimento. As definições do domínio servirão como vocabulário de comunicação na cooperação entre agentes de classes de páginas distintas de um mesmo domínio.

*Os requisitos de comunicação, não apenas em sistemas de manipulação integrada de informação, mas também em ambientes abertos distribuídos em geral, pedem um modelo de comunicação com mais habilidades que o modelo cliente-servidor, e que só modelos de comunicação em nível de conhecimento ('peer-to-peer'), baseados em ontologias reutilizáveis como vocabulário da comunicação, são capazes de oferecer.

*Os mecanismos de busca tradicionais, baseados em técnicas de recuperação de informação e indexação por palavras-chave, podem constituir a base e o suporte para outros mecanismos de busca, aplicativos, agentes e extractores mais refinados, precisos e focados em domínios restritos, baseados em conhecimento explícito reutilizável e comunicável, e habilitados à cooperação.

*A elaboração de uma arquitectura, e sua respectiva realização, num framework de sistemas multiagentes cognitivos para a solução do problema de manipulação integrada de informação, aplicável a grupos de classes identificados na Web, com a possibilidade de cooperação entre os agentes componentes e reutilização massiva de componentes, independente do domínio tratado, conectividade a vários mecanismos de busca, e reutilização de conhecimento, para a eventual construção de novos agentes.

*Agentes que procuram, filtram e classificam páginas da Web para as classes de artigos científicos e para a classe de páginas chamadas de trabalho para eventos científicos.

A Web tem participado, efectivamente, nas actividades rotineiras de milhões de utilizadores dos sistemas computacionais e a sua aplicação aos locais de trabalho tem sido indispensável em muitos casos, principalmente nos meios educacionais. Estas actividades podem ser prejudicadas quando se desperdiçam horas em buscas ineficientes, portanto análises devem ser empreendidas para se compreender melhor os processos de pesquisa implementados pelos utilizadores.

O entendimento dos processos de busca são primordiais para a melhoria da efectividade das pesquisas, pois o tempo consumido com estas actividades chega a 70% do total de acesso à Internet. Uma forma de melhorar o entendimento sobre o processo de busca é estudar o comportamento do pesquisador, analisando as habilidades e condições necessárias para uma busca de sucesso.

As formas de busca indicam estratégias empregues pelos utilizadores, sendo definidas aqui como um plano contemplando uma série de acções visando encontrar uma informação. Como exemplo, uma simples estratégia de busca seria a utilização de um site de buscas (Google, Yahoo!, ...) onde se digita determinado termo e se recebe uma listagem das páginas referenciadas e que contém algum relacionamento com aquele termo. Continua a seguir-se para alguma página recebida e, assim por diante, até encontrar o que procura ou desistir.

Quanto à tendência da globalização do mercado da produção intelectual, pode-se arguir que, dentro do quadro de mudanças estruturais porque vem passando o mundo, a disseminação de padrões culturais globalizados vem assumindo proporções sem limite. Tal situação tem-se acentuado principalmente porque o modo de produção industrial capita-

lista tornou-se hegemónico ou vantajoso na produção e distribuição de produtos intelectuais, e através de seus mecanismos de distribuição - os média em geral - interfere poderosamente nos processos económicos, políticos e culturais das sociedades. Enquanto processo de desenvolvimento de complexas interconexões entre sociedades, culturas, instituições e indivíduos, a globalização estimula e favorece a remoção dos nossos relacionamentos e de nossas referências de vida de contextos locais para contextos transnacionais.

A convergência tecnológica tem vindo a eliminar os limites entre os meios, tornando-os solidários em termos operacionais, e desgastando as tradicionais relações que mantinham entre si e com seus utilizadores. Na verdade, com a tecnologia digital torna-se possível o uso de uma linguagem comum: um filme, uma chamada telefónica, uma carta, um artigo de revista, qualquer deles pode ser transformado em dígitos e distribuído por fios telefónicos, microondas, satélites ou ainda por via de um meio físico de gravação, como uma fita magnética ou um disco. Além disso, com a digitalização o conteúdo torna-se totalmente plástico, isto é, qualquer mensagem, som, ou imagem pode ser editada, mudando de qualquer coisa para qualquer coisa.

A convergência tecnológica parece tender a cancelar a validade de fronteiras entre diferentes tipos de produtos intelectuais, serviços informativos e serviços culturais, e a suprimir as linhas divisórias entre comunicação privada e de massa, entre meios baseados em som e em vídeo, entre texto e vídeo, entre as imagens baseadas em emulsão e as electrónicas, e mesmo a fronteira entre o livro e o ecrã. Uma das maiores consequências disso é a observável tendência de integração de diversos aspectos das políticas públicas para informática, electrónica e telecomunicações, com alguns aspectos das políticas relativas aos média e à cultura. A Internet, a imprensa, a indústria gráfica, o rádio, a televisão, as bibliotecas, os livros e as revistas científicas, as telecomunicações e a informática estão a ficar mais interconectadas e interdependentes, de tal forma que uma política de governo para uma delas pode ter significativas implicações para as outras.

A sociedade, actualmente, pode ser considerada, de modo geral, sociedade da informação, devido ao seu grande envolvimento com o meio da informatização. Independentemente dos caminhos que adoptemos, caberia levar em consideração os seguintes conceitos na abordagem do tema:

a) a imprevisibilidade dos caminhos da inteligência humana fará estas estruturas seguir. O vertiginoso desenvolvimento das tecnologias de informação e comunicações tem sido um poderoso instrumento para a rotina, reorganização e automatização do trabalho intelectual. O fenómeno tecnológico tem operado como libertador de energia cognitiva, que será necessariamente aplicada na área de conhecimento de cada ser humano, não importa seu nível de educação. E dado que, além de libertar energia o fenómeno tecnológico também disponibiliza um fantástico arsenal de ferramentas de concepção e desenvolvimento de produtos e processos, torna-se impossível prever os conteúdos em si mesmos e, mais que isto, as formas que tais conteúdos tomarão, e a maneira como os elementos estruturais se organizarão e se relacionarão entre si e com os utilizadores.

b) a incontrolabilidade dos conteúdos que circulam, sob várias formas, através dos serviços de informações e comunicações. É da própria natureza dos elementos estruturais, sobretudo pelo avanço extraordinário da convergência tecnológica entre informática, comunicações e electrónica, a incontrolabilidade da produção e circulação de conhecimento.

O desenvolvimento tecnológico equilibra a equação social inventando dispositivos de relativo controlo de consumo, pelo menos enquanto se necessita de máquinas lógicas para aceder ao conhecimento circulante. Mais importante que isto, contudo, é o facto incontestável da incontrolabilidade da produção e circulação do conhecimento ser parte constitutiva, estruturante mesmo, da cultura contemporânea. Ela, através das tecnologias de informação e comunicações, realiza e radicaliza o sonho humano libertário (sem restrição às liberdades individuais).

c) a inevitabilidade de acção no sector, seja regulando - ou desregulando - a organização, a gestão e a produção, na intenção de garantir o atendimento do interesse público, a ordem democrática, os valores morais e éticos, a livre competição e a busca contínua da universalização do consumo dos serviços de informação e comunicações.

Um dos principais indicadores do desenvolvimento da sociedade da informação é a penetrabilidade das tecnologias de informação na vida diária das pessoas e no funcionamento e transformação da sociedade como um todo. Em âmbito geográfico, a penetrabilidade é medida principalmente pelo número de utilizadores da Internet numa determinada população.

Outro indicador fundamental da sociedade da informação que com-

plementa a penetrabilidade das tecnologias de informação, constitui o nível de operação ubíqua, num determinado contexto, de recursos, produtos e serviços de informação na Internet por parte dos seus utilizadores, representando indivíduos, governos e as mais diferentes organizações sociais de carácter público ou privado. Esta operação ubíqua representa a consecução de inovações muitas vezes radicais no funcionamento da sociedade actual, especialmente nas actividades e processos que requerem o acesso à informação.

Os recursos, produtos e serviços de informação são identificados na Internet com o nome genérico de conteúdos. Em resumo, conteúdo é tudo o que é operado na Internet. Uma das contribuições mais extraordinárias da Internet é permitir que qualquer utilizador, com carácter individual ou institucional, possa vir a ser produtor, intermediário e utilizador de conteúdos. O alcance dos conteúdos é universal, resguardadas as barreiras linguísticas e tecnológicas do processo de difusão. É através da operação de redes de conteúdos de forma generalizada que a sociedade actual vai mover-se para a sociedade da informação. A força motora para a formação e disseminação destas redes reside na eficiência das decisões colectivas e individuais.

Os conteúdos são, portanto, o meio e o fim da gestão da informação, do conhecimento e da aprendizagem na sociedade da informação. Resumindo, a sociedade da informação desenvolve-se através da operação de conteúdos sobre a infra-estrutura de conectividade.

Pretende-se ressaltar a importância do processo de selecção de conteúdos e dos níveis de qualidade relativa - no sentido interpessoal - que deverão ser objecto de análise e controlo por parte dos sistemas intermediários de informação, mediante instrumentos adequados nas etapas de formação de stocks, processamento técnico e disseminação. Da acção normalizadora e do tratamento parametrizado dos conteúdos vai depender a sua melhor difusão e uso pela sociedade.

Podemos ir mais além e prever que será o volume de conteúdos operados por um país que determinará o seu desenvolvimento económico e social e a qualidade de vida dos seus habitantes. Num contexto globalizado, o volume de conteúdos operados por um país passa também a medir a sua capacidade de influenciar e de posicionar a sua população no futuro da sociedade humana.

Capítulo 4

Aprendizagem Supervisionada

4.1 Classificadores

Na tarefa de classificação, existem algumas técnicas que são utilizadas para a extracção de conhecimento de bases de dados sendo que a seguir serão abordadas somente duas delas: árvores de decisão e redes neurais. Estas duas técnicas baseiam-se na aprendizagem supervisionada, na qual os resultados obtidos necessitam de análise de um especialista que fará a avaliação da relevância dos mesmos. Estas técnicas geram modelos a partir de exemplos de uma base de dados, denominados conjuntos de treino, representando uma amostra dos registos que serão analisados [Gonchoroski, 2007].

Objectiva-se com este estudo a comparação das duas técnicas, observando suas vantagens e desvantagens, de acordo com os resultados obtidos, permitindo a escolha da técnica que revela os resultados mais adequados dentro do contexto da aplicação.

4.1.1 Árvores de Decisão

As árvores de decisão possuem este nome devido a sua estrutura, muito compreensível e assimilativa, se assemelhar a uma árvore. As suas técnicas dividem os dados em subgrupos, baseadas nos valores das

variáveis, sendo que o resultado disto é uma hierarquia de declarações do tipo 'Se...então...' que são principalmente aplicadas quando o grande objectivo da mineração de dados é a classificação de dados ou a predição de saídas. [Martinhago, 2005].

De acordo com Goldschmidt (2005, p. 109), uma árvore de decisão pode ser definida como 'um modelo de conhecimento em que cada nó interno da árvore representa uma decisão sobre um atributo que determina como os dados estão particionados pelos seus nós filhos'. Já de acordo com Sousa (1998), métodos de árvore de decisão representam um tipo de algoritmo de aprendizagem de máquina, que fazem uso de uma abordagem dividir-para-conquistar para classificar casos, representando-os em forma de árvores.

4.1.1.1 Histórico

Muitas pessoas na área de Mineração de Dados consideram Ross Quinlan, da Universidade de Sydney, Austrália, o criador das árvores de decisão. Isto deve-se, em grande parte, à criação de um novo algoritmo chamado de ID3 (Itemized Dichotomizer 3), desenvolvido em 1983. O algoritmo ID3 e versões posteriores como o ID4 e o C 4.5, por exemplo, são estruturados de tal forma que se adaptam muito bem ao serem utilizados em conjuntos com árvores de decisão, visto que eles produzem regras ordenadas por importância. Essas regras são utilizadas na produção de um modelo de árvore de decisão dos factos que afectam os itens de saída [Jeronimo, 2001].

Pode-se dizer que as árvores de decisão são uma evolução das técnicas que apareceram durante o desenvolvimento das disciplinas de machine learning. A partir da aproximação conhecida como Detecção de Interacção Automática, desenvolvida na Universidade de Michigan, as árvores de decisão foram ganhando maior importância no meio científico [Kranz, 2004].

4.1.1.2 Conceitos

Considerada uma ferramenta completa e bastante conhecida para classificação dos dados e apresentação dos resultados na forma de regras, as árvores de decisão são utilizadas, frequentemente, no processo de Des-

coberta de Conhecimento. A classificação é executada, na maioria das vezes, em duas fases no uso das árvores de decisão: construção da árvore e poda [Oliveira, 2001]. Nesta técnica, o utilizador escolhe o atributo que quer avaliar para que o algoritmo procure as variáveis mais relacionadas, gerando uma árvore de decisão com inúmeras ramificações. A árvore criada será utilizada na classificação de novas instâncias, de acordo com os valores dos atributos da nova instância [Araújo, 2006].

Pode-se considerar as árvores de decisão como um algoritmo supervisionado, pois há a necessidade de ser informadas com antecedência as classes dos registos usadas no conjunto de treino. Uma árvore de decisão é formada por um conjunto de nós que são conectados através de ramificações, dividindo-se estes nós em três tipos conforme mostra a figura seguinte, onde o nodo raiz é o início da árvore, os nodos comuns dividem um atributo e geram novas ramificações e o nodo folha possui as informações de classificação do registo [Santos, 2008].

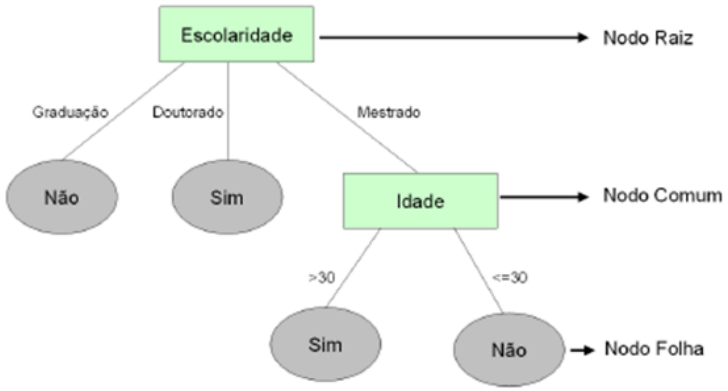

Figura: 4.1: Tipos de nodos de uma Árvore de Decisão

Na fase de construção da árvore, constroem-se ramificações na árvore através de sucessivas divisões dos dados com base nos valores dos atributos. Sendo assim, o processo é repetido recursivamente até que todos os registos pertençam a uma classe [Oliveira, 2001].

Um registo entra na árvore pelo nó (ou nodo) raiz. A partir deste nodo todos os outros nodos são percorridos até ser alcançado o nodo folha. Cada um dos nodos testa o valor de um único atributo e oferece arestas distintas a serem percorridas na árvore a partir deste nodo para

cada um dos seus valores. Assim é determinado o próximo nodo no qual o registo se irá posicionar. Podem ser utilizados diferentes algoritmos na escolha do teste inicial, porém, todos têm o mesmo objectivo: escolher aquele que melhor descreve a classe alvo. Quando o algoritmo chega ao nodo folha, todos os registos que terminam na mesma folha são classificados da mesma forma. É importante salientar que existe somente um caminho da raiz até cada folha, que significa a expressão utilizada para classificar os registos [Borges, 2006].

Após a fase de crescimento da árvore, pode-se encontrar uma estrutura especializada que está super ajustada aos dados, sendo que desta maneira é oferecida mais estrutura que o necessário. Então, a poda passa a ter um papel crucial, fazendo com que sejam consideradas árvores menores e potencialmente de melhor precisão [Sousa, 1998].

Na fase da poda, as ramificações que não têm valor significativo são removidas, a fim de criar um modelo de classificação, fazendo a selecção da sub-árvore que contém a menor taxa de erro estimada [Oliveira, 2001].

Após a fase da poda, a árvore gerada pode representar uma estrutura complexa e de difícil compreensão. Nestes casos pode-se utilizar a extracção de regras como uma fase final, visando extrair regras menores e menos complexas, porém, com precisão similar [Sousa, 1998].

4.1.1.3 Vantagens e Desvantagens

O uso de árvores de decisão possui algumas vantagens em relação às outras técnicas, de entre as quais se pode citar: facilidade de compreender o modelo obtido, uma vez que tem a forma de regras explícitas, possibilitando a avaliação dos resultados e a identificação dos seus atributos chaves no processo; facilidade de expressar as regras como instruções lógicas sendo aplicadas directamente aos novos registos; árvores de decisão são relativamente mais rápidas em comparação às redes neuronais, por exemplo, e na maioria das vezes obtém-se mais precisão nos resultados quando comparadas a outras técnicas de classificação [Oliveira, 2001].

Segundo Sousa (1998), as principais desvantagens no uso de árvores de decisão estão na necessidade de uma considerável quantidade de dados para desvendar estruturas complexas e na possibilidade de haver erros na classificação, no caso de existirem muitas classes, bem como o tratamento de dados contínuos.

Para melhor compreensão dos algoritmos de árvore de decisão, serão apresentados a seguir dois deles: o C4.5 e o Logistic Model Tree (LMT).

4.1.1.4 Algoritmo C4.5

Considerado um dos mais tradicionais algoritmos na tarefa de Classificação, o C4.5 foi inspirado no algoritmo ID3, sendo que o seu método visa abstrair árvores de decisão seguindo uma abordagem recursiva de particionamento das bases de dados [Goldschmidt, 2005].

Também desenvolvido pelo pesquisador australiano Ross Quinlan, em 1993, este algoritmo encontra-se disponível em diversos softwares de mineração. O C4.5 transforma a árvore de decisão num conjunto de regras ordenadas pela sua importância, possibilitando ao utilizador a identificação dos factores mais relevantes nos seus negócios [Oliveira, 2001].

A principal vantagem do algoritmo C4.5 em relação ao ID3, é que ele tem o poder de lidar com a poda (prunning) da árvore, evitando o sobre-ajustamento, com a valoração numérica de atributos e com a presença de ruído nos dados [Borges, 2006]. Na maioria das vezes uma árvore originada do algoritmo C4.5 precisa ser podada pela necessidade de redução do excesso de ajuste (overfitting) aos dados de treino [Martinhago, 2005].

Enquanto o algoritmo ID3 manipula apenas dados nominais, o C4.5 pode manipular também dados numéricos. Entretanto, trabalhar com dados numéricos não é tão simples, pois enquanto os atributos nominais são testados apenas uma vez em qualquer caminho da raiz às folhas, os atributos numéricos podem ser testados diversas vezes no mesmo percurso. Esta característica pode ser considerada uma possível desvantagem do C4.5, pois em alguns casos, a árvore gerada pode ser de difícil entendimento para o utilizador [Borges, 2006].

Neste algoritmo é utilizada a abordagem 'dividir para conquistar', em que o problema original é dividido em partes semelhantes ao original, porém menores, fazendo com que os problemas sejam resolvidos e suas soluções formem uma combinação para o problema inicial. Possui, ainda, a capacidade de aprimorar a estimativa do erro utilizando uma técnica conhecida como v-fold, onde é realizada a validação cruzada com dois ou mais grupos [Gonchoroski, 2007].

De acordo com Oliveira (2001), o algoritmo cria uma árvore com uma quantidade aleatória de folhas por nodo e assume os valores das categorias como um divisor, diferentemente do que se atinge com algoritmos que produzem árvores binárias, por exemplo. Então o prunning é realizado de acordo com a taxa de erro de cada nodo e seus descendentes, sendo que a soma dessas taxas compõem a taxa de erro da árvore. Para a identificação do nodo raiz e dos seus descendentes são realizados os cálculos da entropia e do ganho de informação [Oliveira, 2001].

Após a criação de um conjunto de regras, o algoritmo realiza o agrupamento das regras obtidas para cada classe e a eliminação das regras que não possuem relevância na precisão do conhecimento a ser extraído.

Como resultado final, obtém-se um pequeno conjunto de regras que podem ser facilmente entendidas, criadas pela combinação das regras que induzem à mesma classificação [Martinhago, 2005].

4.1.1.5 Algoritmo LMT

O algoritmo Logistic Model Tree (LMT) aplica os princípios das árvores em problemas de classificação, utilizando para a construção da árvore a regressão logística, que tem como objectivo saber quais variáveis independentes influenciam o resultado, utilizando uma equação para prever um resultado baseado nestas variáveis.

Um processo de adaptação por etapas é empregue na construção dos modelos de regressão nos nodos folhas, realizando uma redefinição incremental àqueles construídos em níveis superiores da árvore [Araújo, 2006].

Este algoritmo normalmente é utilizado para o prognóstico numérico, sendo que os nodos folhas gerados armazenam um modelo de regressão logística para geração do resultado. Após a construção da árvore, é aplicada uma regressão para cada nodo interior, utilizando os dados associados a este nodo e todos os atributos que participam nos testes na sub-árvore. Em seguida, os modelos de regressão logística são simplificados, utilizando a poda. Porém, a poda só acontecerá se o erro estimado para o modelo na raiz de uma sub-árvore for menor ou igual ao erro esperado para a sub-árvore. Após a poda é realizado um processo que forma o modelo final, colocando-o no nodo folha [Landwehr, 2003].

4.1.2 Redes Neurais

Uma Rede Neural Artificial (RNA) é uma técnica computacional que cria um modelo matemático, emulado por computador, simulando um sistema neural biológico simplificado, que tem como principal característica a capacidade de aprendizagem, generalização, associação e abstracção [Araújo, 2006]. São consideradas as técnicas mais comuns utilizadas pelos processos de Mineração de Dados e possuem uma característica que as diferenciam das outras técnicas: podem gerar saídas iguais às entradas, que não existiam durante a fase de treino [Santos, 2008].

Duas características fazem com que as redes neurais sejam semelhantes ao cérebro: o conhecimento é adquirido pela rede através do seu ambiente utilizando um processo de aprendizagem; e as forças de conexão entre os neurónios, mais conhecidas como pesos sinápticos, são usadas para o armazenamento do conhecimento obtido [Haykin, 2001]. De acordo com Sousa (1998), os métodos baseados em RNA proporcionam métodos mais práticos para funções de aprendizagem, que são representadas por atributos contínuos, discretos ou vectores.

A estrutura de uma rede neural consiste em uma quantidade de neurónios interligados que são organizados em camadas. O conhecimento através destas camadas dá-se através da modificação das conexões, que são responsáveis pela comunicação entre as camadas [Araújo, 2006]. A figura seguinte apresenta a arquitectura de uma rede neural simples em que os círculos representam os neurónios, e as linhas representam os pesos das conexões.

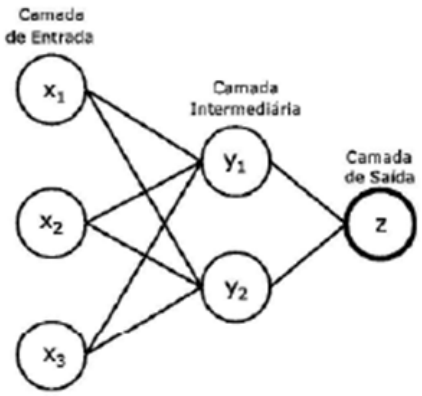

Figura: 4.2: Estrutura de uma Rede Neural Simples

Todas as camadas de uma rede neural possuem funções específicas. A camada de entrada é a que recebe os dados a serem analisados. A camada intermediária é responsável pelo processamento interno das informações e extraem características, permitindo que a rede crie sua própria representação. É importante salientar que uma RNA pode conter várias camadas intermediárias, de acordo com a complexidade do problema. A camada de saída recebe os estímulos da camada intermediária, construindo o padrão que será a resposta para o problema em análise [Araújo, 2006].

O processo de aprendizagem de uma rede neural pode ser realizado de duas formas:

- Supervisionado: é utilizado um conjunto de pares de dados de entrada e saída desejada. A partir dos conjuntos de entrada, a rede neural cria um conjunto de valores de saída desejado. Perante a existência de grande diferença entre as saídas e os pesos sinápticos, são acertados até que a diferença seja diminuída [Santos, 2008].

- Não supervisionado: o treino da rede dá-se apenas através de valores de entrada. Assim, realizam-se processos, chamados de competição e cooperação, entre os neurônios para a classificação dos dados, obtendo-se um reconhecimento de padrões [Santos, 2008].

De acordo com Martinhago (2005), a grande vantagem da utilização de redes neurais é a grande versatilidade que possuem, sendo que o resultado é satisfatório até mesmo em áreas complexas, com entradas incompletas ou imprecisas. Além disso, as redes neurais possuem excelente desempenho em problemas de classificação e reconhecimento de padrões [Santos, 2008].

As desvantagens da utilização das redes neurais estão ligadas à solução final, que depende das condições finais estabelecidas na rede, uma vez que os resultados dependem dos valores aprendidos. Outra desvantagem das redes neurais é o facto de os resultados obtidos não terem uma comprovação, pois todo o conhecimento adquirido pelos neurónios não pode ser representado. Portanto, não é possível comprovar um resultado adquirido através da utilização de redes neurais [Martinhago, 2005].

Em comparação com as árvores de decisão, os algoritmos de redes neurais normalmente necessitam de maior força computacional para serem utilizados. Os tempos de treino variam de acordo com o número de

casos de treino, número de pesos na rede e das configurações dos vários parâmetros do algoritmo de aprendizagem [Sousa, 1998].

4.1.3 Algoritmo de Retropropagação

O surgimento da retropropagação deu-se devido ao interesse por parte dos pesquisadores na resolução de alguns problemas existentes dentro do treino das redes neurais. Após o seu surgimento, este algoritmo tornou-se um dos mais populares para este tipo de treino, sendo considerado um dos responsáveis pelo ressurgimento do interesse nesta área [Araújo, 2006].

O algoritmo de retropropagação utiliza pares para ajustar os pesos na rede, através de um mecanismo de correcção de erros. A sua aprendizagem baseia-se na propagação retrógrada do erro para níveis anteriores da rede, de acordo com o nível de participação que cada neurónio teve na camada superior [Braga, 2000].

O treino através deste algoritmo ocorre em duas fases, chamadas de forward e backward, sendo que em cada uma delas a rede é percorrida em um sentido diferente. Na fase forward um padrão é apresentado à camada de entrada da rede. A actividade resultante percorre a rede, camada por camada, até que uma resposta seja produzida pela camada de saída. Na fase backward é feita a comparação da saída obtida com a saída desejada para este padrão particular. Se o resultado não estiver correcto, o erro é calculado, sendo o erro propagado a partir da camada de saída até a camada de entrada, modificando os pesos das conexões das unidades das camadas internas, de acordo com a retropropagação do erro [Braga, 2000].

A partir das técnicas estudadas busca-se uma relação entre teoria e prática. Sendo assim, a seguir serão apresentados estudos de casos nos quais as soluções para os problemas existentes foram encontradas utilizando técnicas de mineração de dados.

4.1.4 Aplicações de Mineração de Dados

De acordo com o estudo feito através de artigos que utilizam Mineração de Dados e mais especificamente árvores de decisão, foi seleccionado um deles com o objectivo de exemplificar um caso em que o uso de árvo-

res de decisão auxiliou a extracção de conhecimento de bases de dados em áreas distintas.

Assim, é possível perceber que o uso de árvores de decisão se torna cada vez mais comum em diversas áreas, auxiliando a tomada de decisão por parte de homens de negócios e até mesmo por parte de organizações governamentais. O artigo utiliza a Mineração de Dados para criar perfis de utilizadores fraudulentos de empresas de telecomunicações.

- Análise do perfil do utilizador de serviços de telecomunicações utilizando técnicas de mineração de dados [Junior & Perez, 2006]

O crescimento exponencial do prejuízo que as operadoras de telecomunicações absorvem, devido à utilização ilícita dos seus recursos, fez com que as mesmas procurassem alternativas para diminuir este problema.

Assim, percebeu-se que técnicas de mineração de dados, quando aplicadas neste sector, tornam-se um poderoso recurso para identificar o perfil de utilizadores fraudulentos.

Quanto mais rápido forem identificados esses utilizadores, menor será o prejuízo que a operadora de telecomunicações terá e, consequentemente, mais recursos poderão ser oferecidos aos mesmos.

Por outro lado, conhecer o perfil dos bons pagadores através de suas preferências na utilização dos serviços, auxilia as empresas de telecomunicações a realizarem campanhas de marketing, por exemplo, visando promoções e privilégios que as ajudam a manter e agregar clientes novos.

O objectivo principal deste ponto é apresentar o uso de duas técnicas de mineração de dados: redes neurais e árvores de decisão, a fim de avaliar qual delas se torna mais eficaz para identificar o perfil de um utilizador fraudulento.

Como ferramenta de Mineração de Dados para geração de árvores de decisão os autores optaram por utilizar o algoritmo C4.5 do software Sipina, pois o mesmo possui licença para uso educacional, implementa o método de classificação e utiliza árvore de decisão para representar o conhecimento obtido. A figura seguinte apresenta a interface principal do Sipina.

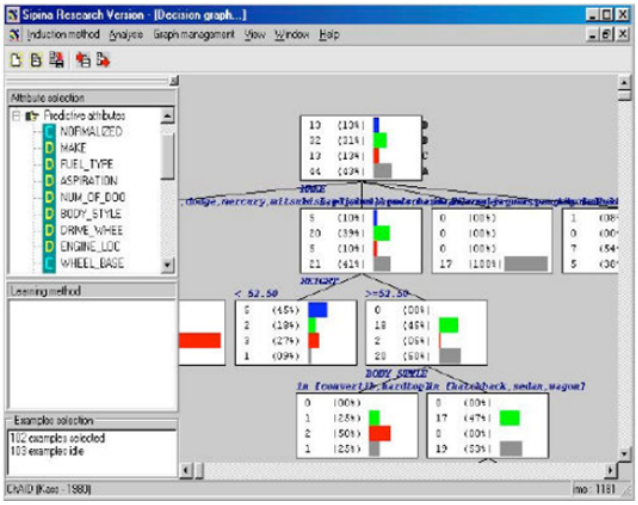

Figura: 4.3: Interface principal do SIPINA

Já para a extracção do conhecimento utilizando redes neurais foi escolhido o software QwikNet, que simula redes neurais executando vários métodos eficientes para treiná-las e testá-las e tem como característica oferecer uma relação flexível e intuitiva, permitindo projectar, treinar e testar redes neurais num ambiente gráfico. A interface do QwikNet é apresentada na figura seguinte.

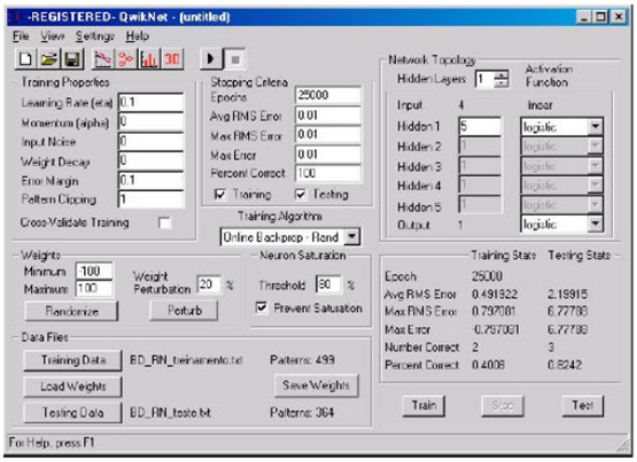

Figura: 4.4: Interface do QuickNet

Inicialmente, foi gerado um arquivo texto contendo os dados das chamadas telefónicas presentes no SGBD da empresa.

Estes dados foram separados por tabulações e, após a limpeza dos mesmos, sofreram uma codificação, utilizando uma folha de cálculo que depois foi guardada em formato '.txt', que os enriqueceram e os prepararam para o processo de descoberta de conhecimento.

Foram disponibilizados 63.534 registos para testes, com dados de chamadas telefónicas realizadas entre 01 de Setembro e 31 de Dezembro de 2005, sendo todas estas chamadas realizadas por assinantes fraudulentos de telefones fixos.

Não foi possível a utilização de um número maior de registos devido ao sigilo.

E também por estes dados serem pontos estratégicos das empresas no combate à fraude.

Estes registos possuem os seguintes dados:

- dia da semana que a chamada foi executada,

- hora inicial da chamada,

- destino da chamada que identifica o tipo do destino (local, DDD, celular, DDI),

- duração das chamadas.

Para a geração da árvore de decisão com o software Sipina, a classe principal foi criada com o atributo 'dia da semana' como nodo principal, com o intuito de descobrir os dias da semana em que os utilizadores fazem mais ligações.

Como nodos filhos foram especificados os atributos hora e destino, com a finalidade de descobrir o horário preferido das chamadas e o tempo utilizado nas conversações.

Como resultado da aplicação deste algoritmo, podem-se considerar as regras apresentadas na figura seguinte.

Regra 01: Os dias da semana com maior número de chamadas são quarta e quinta feira no horário entre 06:00h – 12:00h:
Quarta feira: 22%
Quinta feira: 31%
Na sexta feira o horário de maior tráfego é entre 18:00h – 24:00h: 21%

Regra 02: Nas segundas feiras o horário entre 12:00h e 18:00h concentra chamadas para serviços especiais: 33%

Regra 03: Nas quartas feiras o horário entre 12:00h e 18:00h concentra chamadas:
para telefone fixo (Local): 23%
para telefone celular (DDD): 22%

Regra 04: Nas quintas feiras o horário entre 12:00h e 18:00h concentram-se chamadas para telefone fixo (DDI): 30%

Regra 05: Nas sextas feiras o horário entre 12:00h e 18:00h concentram-se chamadas:
para telefone celular (Local): 22%
para telefone fixo (DDD): 24%

Figura: 4.5: Regras Geradas pelo SIPINA

Conforme as regras obtidas, pode-se definir o perfil geral dos utilizadores, obtendo-se o comportamento generalizado dos fraudulentos.

Com essa definição, pode-se estabelecer um parâmetro comparativo, com o qual é realizada a verificação da semelhança do perfil dos utilizadores individuais com o perfil dos fraudulentos. Analisando os dados de um utilizador específico, é feita a comparação dos resultados com as classes pré-determinadas, verificando se determinado utilizador se encaixa num dos perfis já encontrados.

Na realização dos testes utilizando redes neurais todos os dados foram transformados, conforme realizado no teste com árvores de decisão. A rede neural foi treinada utilizando um arquivo com 499 linhas de dados com informações como dia da semana, horário, destino da chamada e duração, de utilizadores fraudulentos. Para o teste utilizou-se um arquivo com 240 linhas e um arquivo de um único utilizador para comparar a sua semelhança com o perfil de utilizador que a rede neural conseguiu aprender.

Dos 364 registos com os quais foram realizados os testes, representando uma pequena amostra do total de registos, três encaixam-se no

perfil aprendido pela rede neural. Com isso, considera-se que este indicador é bastante baixo quando aplicado para identificação de perfis de utilizadores fraudulentos.

As redes neurais são utilizadas para aprender com o histórico dos utilizadores, analisando o comportamento de cada um em diferentes períodos do dia. Realizando uma análise individual, dão condições às empresas de descobrir em tempo real alguma actividade suspeita, interrompendo-a rapidamente colaborando com o aumento do lucro da empresa.

Com a técnica de árvores de decisão os dados apresentados representam um padrão de comportamento, contudo foi gerado um número elevado de subdivisões, o que fez com que a leitura do resultado se tornasse um pouco demorada, mas de fácil compreensão. A técnica também permite a geração de regras que definem o padrão, o que facilita a procura de novos padrões quando aplicada em outro volume de dados.

4.2 Importância em Organizações

Pode-se afirmar que muitas vezes os dados presentes nas bases de dados das organizações não são aproveitados da melhor maneira possível.

Assim, torna-se fundamental a aplicação de técnicas de mineração de dados nestas bases a fim de transformar informações desconhecidas em conhecimento útil e lucrativo para as empresas.

A partir destas técnicas de mineração de dados é possível criar perfis de clientes, fazendo com que acções possam ser tomadas por parte dos homens de negócios na tentativa de aumentar o potencial da empresa.

Capítulo 5

Agrupamento Incremental e Hierárquico de Documentos

5.1 Era Digital

A chamada 'Era Digital' caracteriza-se como uma época na qual a informação (conhecimento) numa forma electronicamente acessível (i.e. digitalizada) pode ser acedida, partilhada e utilizada de forma fácil, imediata e ampla em actividades económicas. Na 'Era Digital', informação e conhecimento assumem um papel estratégico, alavancando novas possibilidades de crescimento em termos de produtos e serviços em empresas, governos e demais instituições [Lau, 2003]. Os produtos e serviços oferecidos por estas organizações tornam-se, assim, dependentes do capital intelectual dos indivíduos que nelas trabalham. Como consequência, a estruturação, o processamento e a utilização dos dados de uma organização para a geração de conhecimento é uma forma de obter benefícios próprios [Mesquita Mota, 2003]. Segundo Drucker (1987), o funcionamento das companhias é dependente do processamento de informação. Neste contexto, Mesquita Mota (2003) observa que a gestão da informação é um requisito fundamental para o processo de geração de conhecimento.

Outro facto importante é que a utilização da informação como recurso básico para as actividades económicas, em conjunto com a popularização de tecnologias de comunicação tem como consequência directa o aumento da quantidade de informações digitais geradas, seja por utilizadores co-

muns, seja por organizações. Segundo Gantz (2007), se fosse possível imprimir em papel todos os exabytes (1024x1024x1024 Gigabytes) de informação digital gerada, capturada e replicada em 2006, seria possível embrulhar a Terra quatro vezes.

Apesar de parecer extraordinário, no fim de 2009, o tamanho desse universo será de 988 exabytes, com uma taxa de crescimento anual de 57%, sendo possível embrulhar a Terra vinte e cinco vezes.

As informações geradas por uma organização podem ser provenientes de diferentes fontes, tais como documentos de texto, folhas de cálculo, arquivos de imagem, áudio e vídeo gerados pela própria organização; documentos, e-mails e páginas pessoais dos membros; e bases de dados de aplicações corporativas em bases de dados relacionais ou orientados a objectos. A crescente expansão do universo digital, aliada à heterogeneidade das fontes de dados e à falta de controlo e organização dos dados gerados, torna a utilização desses dados e a sua transformação em informação útil cada vez mais complexa. Sem uma solução adequada, muitas das informações geradas por uma organização, que poderiam ser de grande valia para o seu processo de inovação, acabam sendo aproveitadas de modo superficial e insuficiente, ou até mesmo, nem são utilizadas [Gantz, 2007].

Algumas propostas buscam solucionar o problema de acesso a essas informações, sem, contudo, tratar a questão da heterogeneidade do ambiente informacional como problema central. A consequência disso é que o universo de documentos (e, portanto, de informações) a serem recuperados tende a ser limitado. Exemplos de propostas neste sentido são os Sistemas de Recuperação de Informação (SRIs) apresentados por Amorim (2007), Nascimento (2004) e Nunes (2007). Estas soluções direccionam o foco para áreas específicas do conhecimento, o que limita ainda mais a sua abrangência. A solução de Amorim (2007) é direccionada para as áreas de Arquitectura, Engenharia e Construção. A solução apresentada por Nascimento (2004) delimita a recuperação de informações em documentos relacionados com a Construção Civil. Por outro lado, a solução de Nunes (2007) é direccionada para a resolução de problemas na área da Jurisprudência.

Outros trabalhos propõem uma arquitectura específica para a realização de recuperação de informações, porém fazem uso de apenas um motor de indexação (um motor de indexação é um software capaz de

organizar os dados de forma estruturada para facilitar o processamento destes), limitando a abrangência da solução às fontes de dados suportadas por este motor, como no modelo apresentado por Beppler (2005), por exemplo. Outros ainda utilizam a integração de diferentes motores de busca, como a solução de Ma (2005), mas não possibilitam a realização de indexação e busca de informações em diferentes fontes de dados, pois limitam-se apenas ao ambiente Web.

Nenhuma das propostas apresentadas soluciona o problema da heterogeneidade do ambiente, uma vez que todas elas se limitam à indexação e à busca de informações em fontes de dados específicas. Há ainda algumas propostas comerciais para solução do problema da recuperação de informações. Entre elas estão softwares baseados em indexação para ambientes desktop como: o Windows Desktop Search [Microsoft, 2007a] e o Google Desktop Search [Google, 2007a]. Estes softwares oferecem uma interface amigável para o utilizador, mas também não solucionam o problema de recuperação de informações em fontes de dados heterogéneas. O Google Desktop Search oferece possibilidade de extensão limitada, não sendo possível a indexação de informações em fontes que não façam parte de uma das categorias pré-definidas por ele. O Windows Desktop Search, por sua vez, oferece extensão para a indexação e busca de informações em qualquer fonte de dados. No entanto, exige esforço de programação para possibilitar a indexação e busca de informações em algumas fontes, como em bases de dados relacionais, por exemplo.

Diante do panorama apresentado, este trabalho tem como objectivo propor uma solução para o problema de recuperação de informações em ambientes de fontes de dados heterogéneas. A solução está baseada na construção de uma estrutura (framework) capaz de integrar tecnologias de indexação e busca de informações de forma transparente. Esta estrutura possibilita a construção de SRIs capazes de recuperar informações em ambientes de fontes de dados heterogéneas. O desenvolvimento desta estrutura foi baseado em técnicas já consolidadas de sistemas orientados a objectos, bem como padrões de projecto [Gamma, 1994] [Freeman, 2004]. E, por isso, conta com os benefícios que são oferecidos por eles. A solução proposta possui dois pontos fortes: 1) Permite maior abrangência na indexação e busca em um ambiente informacional, o que possibilita a indexação de dados provenientes de diversas fontes, inclusive de bases em bases de dados relacionais, objecto-relacionais ou orientados a objectos; 2) dá maior flexibilidade na escolha dos motores de indexação a serem

utilizados, fornecendo uma maneira simples de substituição dos motores de indexação utilizados, caso necessário.

5.1.1 Metodologia

O estudo exploratório apresentado neste trabalho teve início com uma pesquisa bibliográfica sobre os aspectos e os conceitos relacionados com o tema central, com o intuito de construir uma base teórica que serviria de suporte às etapas posteriores. A segunda fase do estudo consistiu na concepção e desenvolvimento da estrutura de integração. Por fim, foi desenvolvida uma aplicação exemplo, que teve como principal objectivo a validação do estrutura de integração proposto.

5.1.2 Trabalhos Relacionados

A integração de tecnologias em ambientes heterogéneos e a recuperação de informações têm sido estudadas por diversos autores. Alguns deles propõem sistemas que têm como foco a recuperação de informação. Outros propõem modelos de integração de tecnologias para solucionar o problema da heterogeneidade das fontes de dados em diversos contextos. Existem ainda alguns trabalhos que possuem ambas as características, a recuperação de informação e a integração de tecnologias.

Os sistemas de recuperação de informação de Amorim (2007), Nascimento (2004) e Nunes (2007) fazem parte do primeiro grupo. Amorim (2007) apresenta um SRI que tem como objectivo melhorar a precisão e revocação (Precisão - Relação entre o número de documentos relevantes recuperados e o número total de documentos recuperados. Revocação - Razão do número de documentos relevantes recuperados sobre o total de documentos relevantes disponíveis na base de dados [Baeza-Yates & Ribeiro-Neto, 1999]), na busca de informações relacionadas com a área de Arquitectura, Engenharia e Construção (AEC). Para isso, ele utiliza ontologias, que dão significado ao conteúdo dos documentos indexados. Nascimento (2004) também propõe um SRI ligado à área de AEC. No entanto, Nascimento (2004) utiliza algoritmos específicos que têm como base o conhecimento do domínio da construção civil a fim de obter maior eficácia na busca dos dados. Nunes (2007) apresenta um SRI que utiliza uma abordagem muito parecida com a de Amorim (2007), já que faz uso de ontologias e anotações semânticas para auxiliar na pesquisa por

documentos, porém actua no campo da Jurisprudência.

Uchôa (1999) apresenta uma estrutura (framework) para integração de sistemas de bases de dados heterogéneos. Esta estrutura orientada a objectos foi desenvolvida na linguagem C++ e permite que sejam criados sistemas para gerir bases de dados heterogéneos. O trabalho vê a heterogeneidade dos dados como um problema, e procura solucioná-lo dentro do propósito dos sistemas gestores de bases de dados. Para tanto, o autor utiliza integração de tecnologias.

Ma (2005) também apresenta um modelo de integração de tecnologias. No seu modelo, o autor apresenta uma maneira de construir um sistema de buscas na Web fazendo uso de outros sistemas de busca integrados, e, além disso, faz uso de conceitos de inteligência artificial para aprimorar a precisão nas consultas realizadas.

Para a criação do seu modelo, MA (2005) utilizou como base a estrutura de um meta-motor de busca que pode ser observado na Figura 5.1.

A estrutura apresentada pela Figura 5.1 define que o Servidor Web recebe a consulta do utilizador e verifica se os mesmos resultados foram pesquisados recentemente.

Caso tenham sido, o retorno é feito a partir da base de resultados.

Caso contrário, a consulta é enviada para a Interface de Processamento Web que a executa paralelamente nos vários motores de busca que estão a ser utilizados.

Os resultados destas consultas são estruturados e exibidos para o utilizador, e também salvos na base de resultados para consultas posteriores.

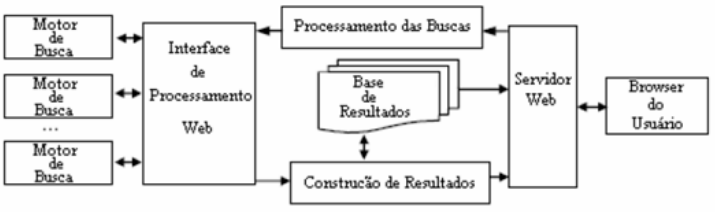

Figura: 5.1: Estrutura de um meta-motor de busca

5.2 Desenvolvimento do Trabalho

5.2.1 Sistemas de Recuperação de Informações

5.2.1.1 Técnicas de Recuperação de Informações

Uma alternativa para o acesso às informações é a navegação pelos documentos, sem a necessidade de indexação prévia [Souza, 2006]. No entanto, segundo Zobel (2006), esta alternativa somente é válida em sistemas onde o número de consultas para busca de informações é considerado pequeno, ou ainda, cujos dados são bastante voláteis. Em ambientes onde muitas requisições de busca são necessárias ou existe uma grande quantidade de dados, a utilização de técnicas de recuperação de informações baseadas em indexação torna-se necessária, visto a economia de tempo em buscas posteriores.

Segundo Baeza-Yates (1999), a recuperação de informação lida com as tarefas de representação, armazenamento, organização e acesso aos itens que contêm a informação. Neste conceito, Souza (2006) define os Sistemas de Recuperação de Informações como sistemas que organizam e viabilizam o acesso aos itens de informação ao desempenhar as tarefas citadas acima.

Os principais modelos de recuperação de informação são o booleano, o vectorial e o probabilístico. No modelo de recuperação booleano, a cada consulta realizada são devolvidos todos os documentos que possuem os dados solicitados pelo utilizador. Com o intuito de aperfeiçoar a busca, é possível que sejam utilizados os operadores OR, AND e NOT para relacionar as palavras da pesquisa. Este modelo é baseado na teoria dos conjuntos e os documentos são qualificados em apenas duas categorias, se possuem ou não o termo pesquisado. Desta forma, não é possível medir grau de relevância dos documentos devolvidos através deste modelo de recuperação.

O modelo vectorial permite que seja determinado grau de relevância para cada documento, o que possibilita a criação de um ranking, e estabelece um critério mínimo de inclusão dos documentos no resultado das pesquisas. Isto é possível devido à representação de cada documento como vectores no espaço de dimensão n, onde n é o número de palavras contidas em todos os documentos que possuem representatividade

no índice. A determinação do grau de relevância pode ser feita com base na distância vectorial entre os documentos e as consultas mapeadas no espaço n-dimensional. A fim de exemplificar, considere-se um índice com apenas três palavras indexadas (universidade, comunidade, digital) pelas quais os documentos presentes no mesmo podem ser pesquisados. Neste caso, o espaço criado possuiria três dimensões. Considere-se também que existem três documentos neste índice (D1, D2 e D3) e que eles contenham, respectivamente, as seguintes palavras: universidade, comunidade; comunidade, digital; e universidade, comunidade, digital.

A figura seguinte mostra como esses documentos seriam mapeados no referido espaço, e como seriam calculadas as distâncias destes para uma consulta (com as palavras universidade e digital), também mapeada neste mesmo espaço. Estas distâncias determinam a relevância do documento para a consulta, sendo que quanto menor a distância entre o documento e a consulta, mais relevante ele é para esta. A figura seguinte evidencia que o documento D3 possui maior relevância para a consulta mapeada (d3 é menor que d1 e d2), e que D1 e D2 apresentam a mesma relevância para esta consulta (d1 é igual a d2).

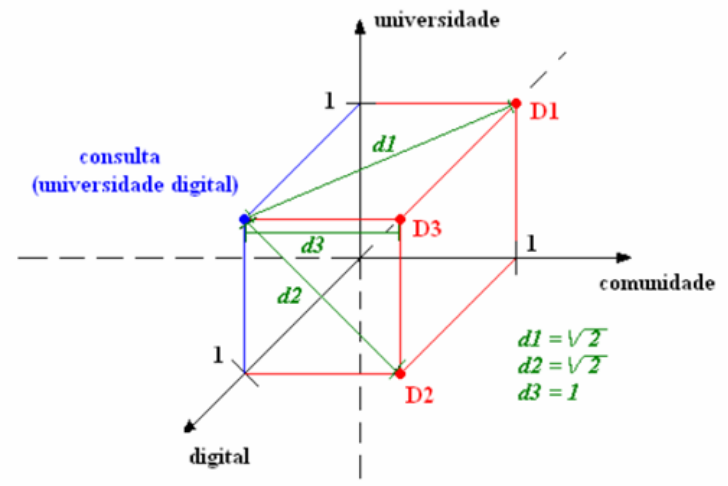

Figura: 5.2: Exemplo de um modelo vectorial de recuperação de informações

No modelo probabilístico de recuperação de informações, supõe-se que exista um conjunto óptimo de documentos para cada pesquisa dos

utilizadores, e que ele seja passível de recuperação. Com o intuito de o obter, este modelo utiliza outro método de recuperação para colher uma lista inicial e, a partir dela, realizando interacções sucessivas com os utilizadores, fazer análises de relevância dos documentos a serem devolvidos.

Actualmente, grande parte dos SRIs é baseada no modelo vectorial de recuperação de informações. Novas técnicas estão sendo estudadas, com o intuito de complementar os modelos de recuperação existentes, possibilitando a criação de SRIs mais eficientes. São alguns exemplos: 1) a identificação de padrões semânticos existentes nos documentos; 2) a utilização de metadados para facilitar a identificação dos documentos; 3) a criação de novas metáforas visuais, de modo a facilitar a extracção de conhecimento das informações recuperadas; 4) a utilização de informações pessoais e comportamentais dos utilizadores, obtidas através de websites de relacionamento e interacções, com o intuito de definir relevância específica para cada utilizador; 5) a inserção de dados semânticos nos documentos, com o intuito de aumentar a precisão no retorno [Souza, 2006].

5.2.1.2 Indexação

A indexação de dados incorpora diversos conceitos multidisciplinares de linguística, matemática, psicologia cognitiva, ciência da computação e outros. Além disso, envolve diversos aspectos relacionados com outras actividades de gestão do conhecimento, de entre os quais podem ser destacadas as técnicas de recuperação e de interpretação de informações a serem utilizadas e a estrutura de armazenamento dos dados indexados. Estes conceitos relacionam-se entre si com o objectivo de possibilitar a recuperação das informações de forma rápida e precisa, poupando tempo e trabalho necessários à execução de tarefas que as utilizem como recurso [Lima, 2003].

O processo de indexação pode ser executado sobre dados provenientes de diversas fontes, desde arquivos de texto até arquivos de vídeo, áudio e imagem. Apesar de ser possível ou necessário o registo de diferentes informações na indexação destes dados, o processo de indexação consiste basicamente do mesmo princípio em todas as ocasiões, ou seja, mapear os termos de um documento numa estrutura de dados específica chamada de índice [Zobel, 2006].

Para efectuar a construção de um índice, quaisquer que sejam os dados a serem indexados são necessários os seguintes passos: 1) Tokenize (Análise Lexical), 2) Analysis (Retirada das palavras sem significado na linguagem natural) e 3) Stemming (Radicalização). O Tokenize consiste na separação e armazenamento de cada token (uma sequência de caracteres que pode ser tratada como uma unidade dentro da gramática de uma linguagem [Appel, 2002]). Nesta etapa, também é possível o registo de informações relevantes sobre cada um deles. Ao fim da etapa de Tokenize, obtém-se uma lista com todos os tokens de um documento. O processo Analysis retira os tokens pouco relevantes para as pesquisas da lista gerada na etapa anterior, como artigos, preposições, pontuações, espaços em branco, entre outros.

Os tokens são considerados relevantes ou não de acordo com a língua utilizada para escrever o documento que os contém [Gospodnetic, 2005]. Além disso, nesta etapa, podem ser retirados os acentos e substituídas letras em maiúsculo de cada token, o que possibilita uma pesquisa mais abrangente nos documentos.

A etapa Stemming consiste em reduzir cada token remanescente das etapas anteriores na sua palavra base. Para isso, são retirados prefixos, sufixos, plural, bem como flexões verbais, e qualquer outra flexão existente na palavra original. Não é necessário que a palavra-base obtida pelo processo seja exactamente a raiz daquela palavra (de acordo com formalidades da língua). É satisfatório que palavras relacionadas ou derivadas possuam a mesma base dentro do índice. Dessa forma, é possível que seja efectuada a recuperação de palavras derivadas da mesma base ao realizar-se uma única consulta [Peng, 2007] [Gospodnetic, 2005].

Os tipos de índices mais comuns são o Índice Invertido e o Índice Sequencial. O Índice Invertido consiste no mapeamento de cada token para a lista de documentos aos quais ele pertence. Este índice pode-se apresentar também como uma árvore binária, o que reduz o tempo das pesquisas. No entanto, a utilização de índices dispostos como árvores binárias aumenta a necessidade de alocação de espaço em memória para o armazenamento [Cormen, 2002]. A criação e actualização de um índice invertido exigem que sejam pesquisados todos os tokens a serem inseridos ou alterados, o que pode gerar um aumento de tempo considerável (overhead) no processo de indexação. Um exemplo de Índice Invertido pode ser observado na figura seguinte que apresenta uma lista de tokens à esquerda, sob o título 'Palavras', apontando, cada token, para a lista

de documentos que o possuem (apresentadas de forma horizontal sob o título 'Documentos').

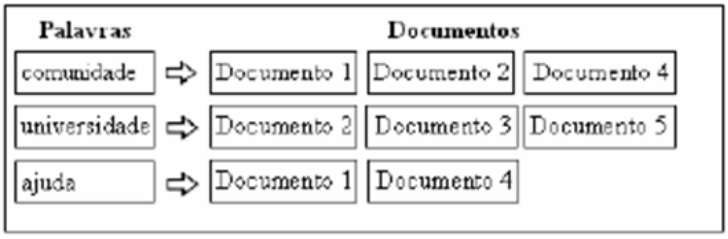

Figura: 5.3: Exemplo de índice invertido

O Índice Sequencial consiste numa lista de pares (documento, token) ou (documento, lista de tokens), ordenados pelos documentos. Esta abordagem faz com que seja reduzido o overhead na inclusão e actualização existente nos Índices Invertidos. No Índice Sequencial, as inserções sempre ocorrem no final, e as actualizações ocorrem directamente no par do documento a ser actualizado. Geralmente, um Índice Sequencial é transformado em Índice Invertido em tempo de execução com o intuito de reduzir o tempo de pesquisa.

Um exemplo de Índice Sequencial é apresentado na figura seguinte que apresenta à esquerda uma lista de documentos indexados que aponta, cada um deles, para a lista de tokens que possui.

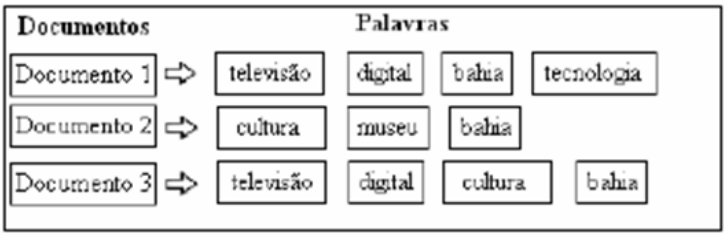

Figura: 5.4: Exemplo de índice sequencial

5.2.2 Motores de Indexação de Dados

Os motores de indexação armazenam documentos em um repositório e mantêm um índice desses documentos. As consultas são enviadas

para este índice e os documentos que possuem as palavras consultadas são devolvidos. Estes documentos são recuperados através de diferentes técnicas e podem não conter necessariamente todos os termos consultados. Além disso, estes documentos são classificados através de alguma métrica com o objectivo de devolver os documentos relevantes para o utilizador [Zobel, 2006].

Com o objectivo de desenvolver o modelo de integração, os motores de indexação Windows Search [Microsoft, 2007a], o Google Desktop Search [Google, 2007a] e o Lucene [Apache, 2007] foram estudados de modo a permitir que o projecto desenvolvido desse suporte às características que lhes são comuns e a outros motores de indexação que seguem o mesmo padrão. Isto possibilitou que a estrutura de integração criada desse suporte ao desenvolvimento de SRIs utilizando alguns dos principais motores de indexação e busca [Cole, 2005].

O primeiro motor de indexação estudado foi o Windows Search, que é o motor de indexação de dados utilizado pelo sistema integrado de busca e organização do sistema operacional Windows Vista. Este motor também é utilizado pelo Windows Desktop Search (WDS), um software independente que pode ser instalado nos sistemas operacionais Windows XP e Windows 2003 Server [Microsoft, 2007a]. O estudo realizado sobre o Windows Search foi baseado no software WDS.

O objectivo do WDS é fornecer administração dos dados digitais, indexando e buscando mais de 200 tipos de arquivos normalmente utilizados. De entre estes estão e-mails e contactos gerados pelos produtos da Microsoft como Outlook Express e Microsoft Office, documentos de texto, apresentações, metadados de arquivos de imagem e música e arquivos de código fonte. Outra característica desta aplicação é permitir a obtenção de dados presentes em arquivos da máquina local ou de alguma outra máquina da rede [Microsoft, 2007a].

O Windows Search disponibiliza um Software Development Kit (SDK) que pode ser utilizado para adicionar a funcionalidade de busca de informações a outras aplicações a serem desenvolvidas. Através deste SDK, o WDS pode ser estendido para possibilitar a indexação e busca em outras fontes de dados. Com o objectivo de facilitar o acesso aos dados indexados pelo Windows Search, o seu SDK torna indiferente tanto o tipo de dado devolvido, como a aplicação que faz uso deste, possibilitando o acesso aos dados de maneira única.

Com o intuito de facilitar o desenvolvimento de eventos que exerçam as principais funcionalidades intrínsecas a um sistema de busca de informações, este SDK fornece implementações de interfaces que permitem efectuar as operações de indexação, pesquisa, reindexação e remoção (desindexação) dos dados presentes no índice [Microsoft, 2007b].

Após o Windows Search, foi estudado o motor de indexação do Google Desktop Search (GDS). Este software é similar ao WDS e tem como objectivo facilitar o acesso a informações. O GDS é capaz de indexar e pesquisar informações em diversos tipos de arquivos, tais como: e-mails do Microsoft Outlook E-mail, Outlook Express, Thunderbird e Netscape Mail; arquivos do Microsoft Word, Excel, PowerPoint e em formato pdf; conversas do Windows Messenger, AOL Instant Messenger e Google Talk; histórico do Internet Explorer, Firefox, Mozilla e Netscape; metadados de arquivos de música, imagens e vídeo. A ferramenta também permite que sejam indexados arquivos de computadores remotos através de uma rede local [Google, 2007a].

Existem também algumas possibilidades de extensão das características do GDS com o uso do seu SDK que permite dois tipos de utilização: 1) o desenvolvimento de plugins para adicionar a capacidade de indexação de novos tipos de arquivos não reconhecidos originalmente; e 2) o uso das funcionalidades de indexação e busca em outras aplicações. No entanto, não é possível a recuperação de informações em algumas fontes de dados através do GDS (como em bases de dados, por exemplo). Apesar de ser extensível, este motor possui extensão limitada a categorias pré-definidas que não englobam todas as fontes existentes [Google, 2007b].

Por fim, foram estudadas as características do Lucene, um projecto de código aberto, licenciado sob a Apache Software License. Diferentemente dos softwares apresentados nas secções anteriores, o Lucene não apresenta uma interface de interacção com o utilizador. Isto ocorre porque o Lucene não é um software destinado ao utilizador final, e sim uma biblioteca de recuperação de informação, com o objectivo de permitir que o programador adicione à sua aplicação a capacidade de indexar e buscar dados. Esses dados podem ser provenientes de diversas fontes devido ao facto deste motor não assumir qualquer característica do que será indexado. A única premissa é que os dados possam ser convertidos para um formato textual. No entanto, esta biblioteca não fornece suporte nativo a nenhum tipo de arquivo. A indexação de qualquer fonte de dados

depende de rotinas de responsabilidade do programador [Gospodnetic, 2005].

Apesar de ser originalmente escrito em Java, actualmente, o Lucene está transcrito em outras linguagens, o que possibilita a sua utilização num maior número de projectos. Algumas dessas linguagens são C++, Perl, Python, .NET e Ruby. Isto significa que as funcionalidades da biblioteca Lucene podem ser exploradas através de qualquer uma dessas linguagens e, inclusive, por mais de uma ao mesmo tempo, já que os índices gerados são padronizados para a utilização por qualquer um desses projectos [Gospodnetic, 2005].

A figura seguinte apresenta uma visão geral das principais características dos motores de indexação Windows Search (WS), Google Desktop Search (GDS) e Lucene. Foram levados em conta a existência de interface com o utilizador, a possibilidade de extensão, a disponibilidade de SDK, as linguagens de programação disponíveis pelo SDK e as fontes de dados que são indexadas por padrão (sem a necessidade de extensão).

	Interface com o usuário	Extensível	Possui SDK	Linguagens de programação	Fontes de dados indexadas por padrão
WS	SIM	SIM	SIM	C++ e .NET(C#)	E-mail (eml, msg), Contatos (vcf), Documentos de texto (doc, dot, htm, html, mht, one, rtf, txt, xml), Planilhas (xls, xlw), Apresentações (pot, pps, ppt, xls, xlw), Pastas e Outros (bat, c, cmd, cpp, cxx, Dif, disco, h, hpp, hxx, idl, inc, inf, inx, js, nws, pl, ppa, pwz, rc, reg, resx, slk, url, vbs, xla, xld, xlt, xlv, xsl)
GDS	SIM	SIM *	SIM	C e Java (não oficial)	E-mail (Gmail, Outlook Email, Outlook Express, Netscape Mail, Thunderbird), Documentos de texto (Word), Planilhas (Excel), Apresentações (PowerPoint), Histórico de navegadores (Internet Explorer, Firefox, Mozilla, Nestcape), Conversas (Google Talk, MSN Instant Messenger, AOL Instant Messenger), Outros (arquivos pdf, zip)
Lucene	NÃO	SIM	SIM	Java, C++, Perl, Python, .NET/C# e Ruby	Todas as fontes cujos dados possam ser convertidos em formato textual **

* Extensibilidade limitada às categorias pré-definidas

** A indexação de qualquer fonte de dados depende de rotinas de responsabilidade do programador

Figura: 5.5: Comparação dos motores de indexação estudados

O estudo sobre estes motores de indexação evidenciou características específicas a cada um deles que limitam a indexação de informações em

ambientes de fontes de dados heterogéneas utilizando apenas um deles.

Algumas fontes de dados comuns em ambientes informacionais corporativos, como bases de dados relacionais, não são suportadas originalmente para indexação e busca de informações por estes motores.

O WDS pode ser estendido através do seu SDK para suportar tal característica ou pode ser desenvolvido um módulo que realize esta operação através do Lucene.

O GDS não suporta este tipo de extensão.

Outras fontes de dados não suportadas para a recuperação de informações através destes motores são imagem, áudio e vídeo. Essas fontes são as principais responsáveis pelo crescimento do universo digital existente actualmente [Gantz, 2007].

Os motores de indexação estudados indexam somente os metadados destas fontes, não processando os dados contidos nelas. Assim, não é possível a recuperação de informações existentes nestes formatos através destes motores.

Apesar de limitar a recuperação de informações em ambientes de fontes de dados heterogéneas, os motores estudados possuem propriedades fundamentais que facilitam a indexação de informações nestes ambientes, pois são capazes de executá-la sobre conjuntos de dados específicos.

5.2.3 O Modelo de Integração

O modelo de integração de motores de indexação proposto tem o intuito de proporcionar o desenvolvimento de SRIs que visem solucionar o problema apresentado: a dificuldade de acesso às informações digitais em ambientes de fontes de dados heterogéneas.

Este modelo propõe o trabalho conjunto e simultâneo de diferentes motores de indexação, e a independência destes em relação às lógicas de negócio das aplicações desenvolvidas sobre o modelo.

Esta secção apresenta o modelo de integração proposto, destacando a arquitectura sobre a qual ele foi projectado e a estrutura de integração que proporciona o uso conjunto de diferentes tecnologias de indexação e busca de informações.

5.2.3.1 A Arquitectura

Com o objectivo de integrar diferentes tecnologias de indexação e busca, o modelo proposto é dividido em três camadas independentes: 1) um universo heterogéneo de dados; 2) os motores de indexação e busca; e 3) o estrutura de integração.

Com isso, a arquitectura proposta é suficientemente flexível.

Na figura seguinte, a primeira camada representa um universo heterogéneo de dados, análogo ao de um ambiente corporativo, onde os dados serão indexados e recuperados.

Este universo pode ser composto por dados provenientes de diversas fontes, tais como documentos de texto ou folha de cálculo, páginas Web, arquivos de imagem, áudio e vídeo, bases de dados relacionais ou orientados a objecto, entre outras.

As diferentes tecnologias utilizadas para o propósito de indexação e busca de dados - os motores de indexação - compõem a segunda camada do modelo.

Apesar de todos os motores serem responsáveis por indexar e pesquisar conteúdo, eles exercem funções distintas dentro dessa camada ao indexar fontes de dados específicas.

Para exemplificar a distinção de tarefas dos motores de indexação, pode-se imaginar um SRI que faça uso do modelo proposto e utilize dois motores de indexação para indexar diferentes fontes de dados.

O primeiro responsável pela indexação de arquivos comuns num disco e o segundo por indexar informação contida numa base de dados relacional.

A terceira camada consiste na estrutura de integração, sendo a mais importante do modelo proposto.

Esta camada possui os objectos que propiciam a actuação conjunta das diferentes tecnologias de indexação utilizadas.

Além disso, ela permite que a indexação e a recuperação dos dados sejam realizadas de maneira transparente para os programadores e para os utilizadores dos SRIs desenvolvidos sobre o modelo de integração.

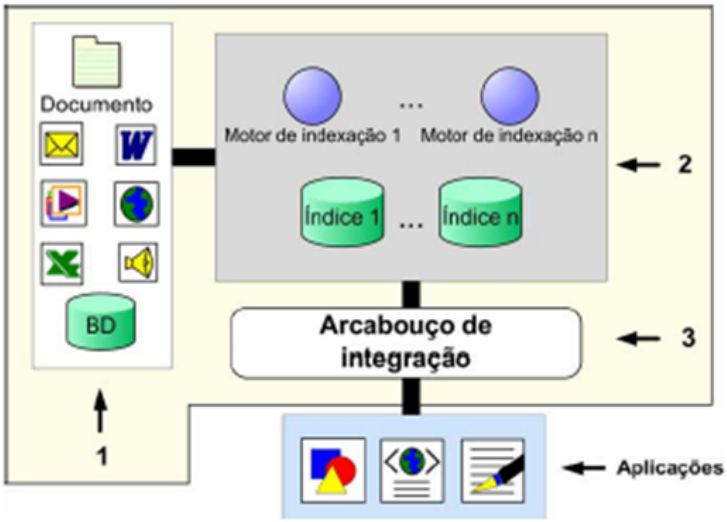

Figura: 5.6: Arquitectura proposta para um Sistema de Recuperação de Informações (SRI)

Abaixo da estrutura de integração podem existir diversas aplicações com o objectivo de indexar e recuperar as informações e, possivelmente, extrair conhecimento destas. Essas aplicações formam uma camada de interacção com os utilizadores, que, propositadamente, não faz parte do modelo proposto. Os dados recuperados podem ser apresentados em diferentes metáforas visuais, processados sob algoritmos de Mineração de Dados com o intuito de efectuar a extracção de conhecimento, ou ainda serem disponibilizados através de serviços Web (WebServices). Essas são apenas algumas das possibilidades de tratamento dos dados recuperados. A definição final do processamento destes dados fica a cargo do programador e dos requisitos dos utilizadores do sistema.

Conforme apresentado, o modelo de integração proposto possibilita a recuperação de informações provenientes de diversas fontes de maneira simultânea. Isto é consequência da integração de dois ou mais motores na indexação e busca de dados. Dessa forma, o modelo proposto apresenta-se como uma solução para a recuperação de informações em ambientes de fontes de dados heterogéneas. Outra propriedade do modelo é a possibilidade de recuperação dos dados de forma padronizada. Isso ocorre independentemente de quais ou quantos são os motores de indexação utilizados, mesmo que eles forneçam as respostas às consultas

em um formato diferente. Com isso, torna-se possível realizar operações sobre os dados sem a necessidade de conhecimento do motor de indexação responsável pela recuperação dos mesmos.

5.2.3.2 A Estrutura de Integração

A figura seguinte apresenta a estrutura de integração que compõe a terceira camada da arquitectura apresentada na secção anterior. A estrutura é composta por três módulos: Indexação, Busca e Factory. Estes módulos foram desenvolvidos sob o paradigma de orientação a objectos, tendo como base conceitos já consolidados da Engenharia de Software, bem como de Padrões de Projecto. Assim, o modelo de integração se aproveita dos benefícios consequentes das melhores práticas de desenvolvimento de software.

O módulo de indexação é composto pelas interfaces IIndex, IDocumentIndex e IDBIndex. As interfaces IDocumentIndex e a IDBIndex são especializadas de IIndex. Além dessas interfaces, o módulo de indexação ainda deve contar com a implementação de classes que cumpram o que é especificado nelas, para que a indexação dos dados seja possível. A principal finalidade dessas interfaces é definir o comportamento dos tipos de motores de indexação, e disponibilizá-los de modo que o programador não necessite ter conhecimento sobre qual motor será utilizado para uma indexação específica.

A interface IIndex especifica um comportamento geral que todos os motores de indexação possuem. Este comportamento é simbolizado, somente, pelo método ClearIndex, que tem como propósito efectuar a limpeza do índice gerado pelo respectivo motor de indexação do que o implementar. Em trabalhos futuros, caso seja observado algum outro comportamento-padrão para todos os motores de indexação, deverá ser possível expressá-lo através desta interface. A interface IDocumentIndex representa um padrão de procedimentos que são executados por motores de indexação de documentos comuns de um sistema de arquivos. Define o método Index que deve indexar o conteúdo de arquivos internos a um directório-raiz que é recebido como parâmetro. É previsto que essa indexação aconteça, inclusive, em todos os arquivos contidos nos subdirectórios a partir deste directório-raiz. Além do método de indexação, esta interface define o método IncrementalIndex que deve possibilitar a indexação incremental de um directório recebido por parâmetro. A in-

dexação incremental deve ser realizada sobre os arquivos que sofreram alterações ou que ainda não foram indexados.

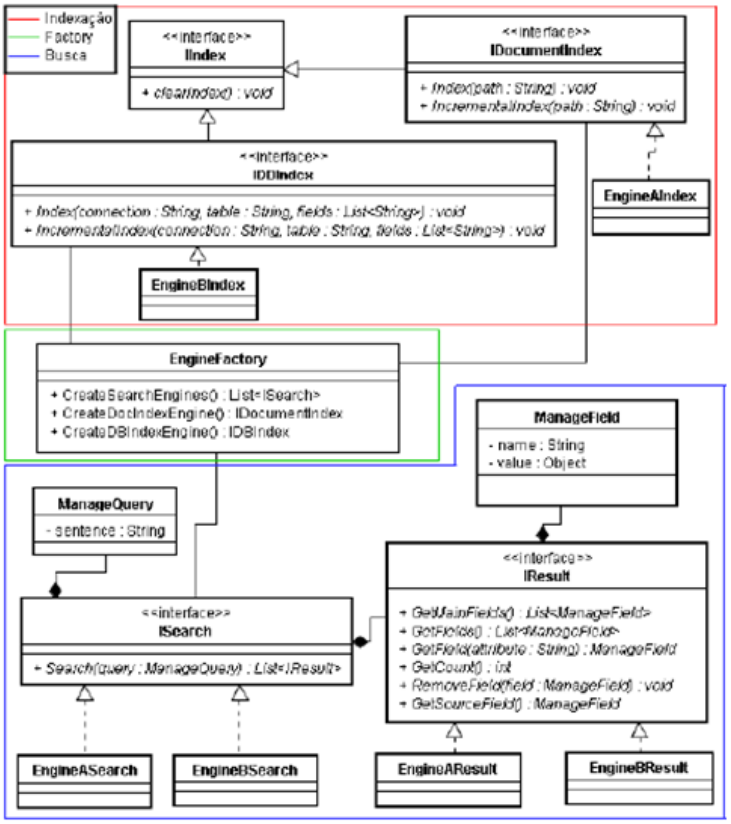

Figura: 5.7: Diagrama de classes do estrutura de integração

A terceira interface deste módulo, a IDBIndex, define o formato de um motor de indexação de informações presentes numa base de dados relacional. Assim como a IDocumentIndex, os métodos Index e IncrementalIndex, que recebem outros parâmetros, têm o objectivo de realizar a indexação de dados presentes em um base de dados relacional.

As implementações das interfaces IDocumentIndex e IDBIndex são representadas na figura anterior por EngineAIndex e EngineBIndex, respectivamente. Estas implementações cumprem o que foi determinado pelas interfaces e delegam aos motores de indexação a tarefa de indexar os dados.

O módulo de busca é o que contém o maior número de classes. Este facto acontece porque ele tem a obrigação de cobrir uma maior quantidade de responsabilidades. Este módulo é formado pelas classes ManageQuery e ManageField, além das interfaces ISearch e IResult e suas respectivas implementações.

Estas interfaces, assim como as interfaces do módulo de indexação, representam abstracções e fornecem ao programador as mesmas características oferecidas pelo módulo de indexação. Ao contrário do que acontece no módulo de indexação, entretanto, os objectos definidos pelas interfaces deste módulo são fornecidos em forma de listas: lista de motores de busca (ISearch) e lista de resultados (IResult) associada a cada um dos motores de busca.

A interface ISearch tem como principais objectivos definir o comportamento dos motores de busca do sistema e possibilitar o acesso aos mesmos na camada seguinte da arquitectura de forma transparente, ou seja, sem que se saiba quais motores estão sendo utilizados para a realização das buscas. Ela também define o método Search, que é responsável por solicitar a pesquisa ao motor de busca relativo à implementação e devolver os valores encontrados no padrão definido pela interface IResult. O objecto que tem representação nessa interface é composto por uma lista desses resultados e por uma consulta (ManageQuery) que serão detalhados a seguir.

A consulta (ManageQuery) é um objecto que encapsula os dados a serem requisitados. Esta classe possui apenas o atributo sentence, que simboliza o texto que efectivamente será pesquisado pelos motores. Porém, a criação dessa classe tem o propósito de facilitar a extensão da estrutura de integração para suportar futuras pesquisas possibilitadas pelo uso de outro motor ainda não conhecido ou pela evolução de algum dos motores estudados.

O padrão do resultado que será devolvido por todos os motores de busca é delineado pela interface IResult, como já foi explicitado. Definir essa padronização é o principal intuito desta interface. Dessa forma, todos os resultados provenientes de motores de indexação diferentes podem ser submetidos ao mesmo tratamento na camada seguinte. Isto retira a importância do motor real e transfere-a para sua abstração, como no princípio da inversão de dependência descrito por Freeman (2004). Um objecto que implementa a interface IResult é composto por uma lista

de campos e possui os métodos apropriados para permitir operações sobre eles. Cada um desses campos representados pela classe ManageField possui um nome e um valor. Esse valor é definido genericamente para suportar os tipos de informações heterogéneas que podem estar contidas nesses campos, tais como: texto comum, números inteiros, números reais, datas, entre outros.

É importante ressaltar que para cada motor de indexação utilizado é indispensável que se tenha uma implementação da interface ISearch e outra da IResult. A implementação da interface ISearch permite a realização de consultas no seu respectivo motor de indexação, enquanto a implementação da interface IResult padroniza os resultados de um motor específico para que eles possam ser devolvidos.

O módulo Factory é constituído exclusivamente pela implementação da classe EngineFactory. Este módulo foi criado com o objectivo de encapsular a criação dos objectos da estrutura de integração em uma única classe, permitindo o desenvolvimento voltado para interfaces e não para as implementações das classes concretas da estrutura, caracterizando o princípio Open-Closed - código aberto para extensões, porém fechado para modificações [Freeman, 2004] [Meyer, 2000].

A criação deste módulo foi baseada nos padrões Abstract Factory e Factory Method [Gamma, 1994] [Freeman, 2004]. Foi utilizada uma adaptação apresentada por Freeman (2004) como Simple Factory, que é uma prática muito utilizada para o encapsulamento da criação de objectos.

A classe EngineFactory é composta de três métodos, o CreateSearchEngines, o CreateDocIndexEngine e o CreateDBIndexEngine.

O CreateSearchEngines devolve uma lista com todas as instâncias dos motores de busca dos motores utilizados. Dessa maneira, a busca pode ser realizada em todos os índices criados por esses motores, mantendo o conceito de índice único com o qual a estrutura trabalha. Isto garante também a transparência na recuperação dos dados pela camada de utilização.

Os métodos CreateDocIndexEngine e CreateDBIndexEngine são responsáveis por instanciar o motor de indexação de documentos e o de base de dados, respectivamente, devolvendo-os para a utilização do programador. Como foi dito anteriormente, estes motores têm o comportamento

delineado pelas interfaces IDocumentIndex e IDBIndex.

Os métodos da classe EngineFactory asseguram a transparência dos motores utilizados pelo programador nas aplicações sobre o modelo. Isto permite que a indexação e a busca de informações sejam realizadas sem o conhecimento prévio sobre quais motores estão sendo utilizados em operações específicas.

5.2.3.3 A Extensão do Modelo

A partir das definições dos módulos da estrutura de integração apresentados na secção anterior é possível desenvolver uma aplicação que faça uso de dois ou mais motores de indexação e busca de maneira integrada. Todavia, podem ser necessárias extensões ou modificações dos motores utilizados nessa aplicação. Exemplos seriam a adição de um motor de indexação a uma aplicação já estruturada com dois ou mais motores e, também, a substituição de um dos motores utilizados.

Há ainda uma terceira possibilidade de alteração que é a introdução da capacidade de indexação de um novo tipo de dados não coberto pela estrutura actual. A estrutura de integração foi projectada para permitir essas alterações de modo a não tornar necessária nenhuma alteração na codificação das classes de negócio da aplicação e garantir, dessa forma, a independência entre as camadas do modelo proposto. A análise de cada uma dessas modalidades de adaptação fornecidas pela estrutura é discutida na sequência do texto.

A incorporação de um motor de indexação à estrutura é feita através de algumas implementações de interfaces já pertencentes aos módulos de busca e indexação e de pequenas alterações no módulo Factory.

No módulo de busca devem ser criadas implementações das interfaces ISearch e IResult relativas ao motor que se deseja adicionar. Já no módulo de indexação é preciso que seja implementada apenas uma das interfaces que defina o tipo de dados que serão indexados com o novo motor: IDocumentIndex ou IDBIndex. Essas alterações são ilustradas na figura seguinte.

Para finalizar a inclusão de um novo motor, devem ser feitas alterações na classe EngineFactory. Em primeiro lugar, deve ser criado um novo método que seja responsável pela criação do objecto utilizado para

realizar a indexação usando este motor. Além disso, o novo motor deve ser adicionado ao método CreateSearchEngines da classe EngineFactory, que é responsável por devolver os objectos que executam as buscas.

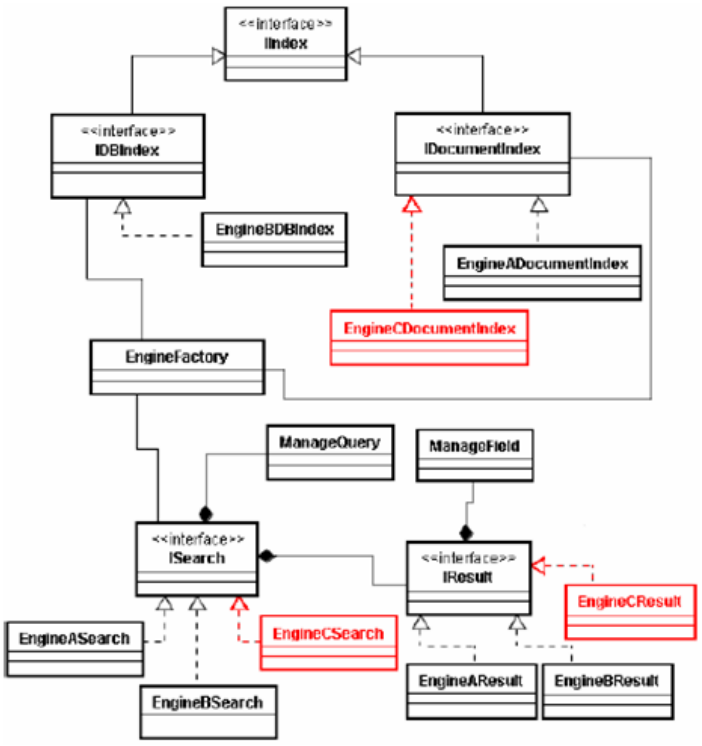

Figura: 5.8: Adição de um motor de indexação à estrutura de integração

A substituição de um dos motores que estão a ser utilizados tem um grau de complexidade baixo para quem possui conhecimento sobre os padrões de desenvolvimento orientados a objectos. É preciso que sejam codificadas as mesmas implementações dos módulos de busca e indexação necessárias à inclusão de um novo motor. Entretanto, no módulo Factory deve-se apenas substituir a criação do motor de indexação que se deseja colocar pelo que dará lugar a ele, bem como adicionar o pesquisador da mesma forma que é feita na inclusão de um novo motor.

Na figura seguinte, são exibidas em vermelho as classes que devem ser adicionadas e, em cinza, as classes que deixarão de ser utilizadas ao efectuar-se a substituição do motor EngineA pelo EngineC.

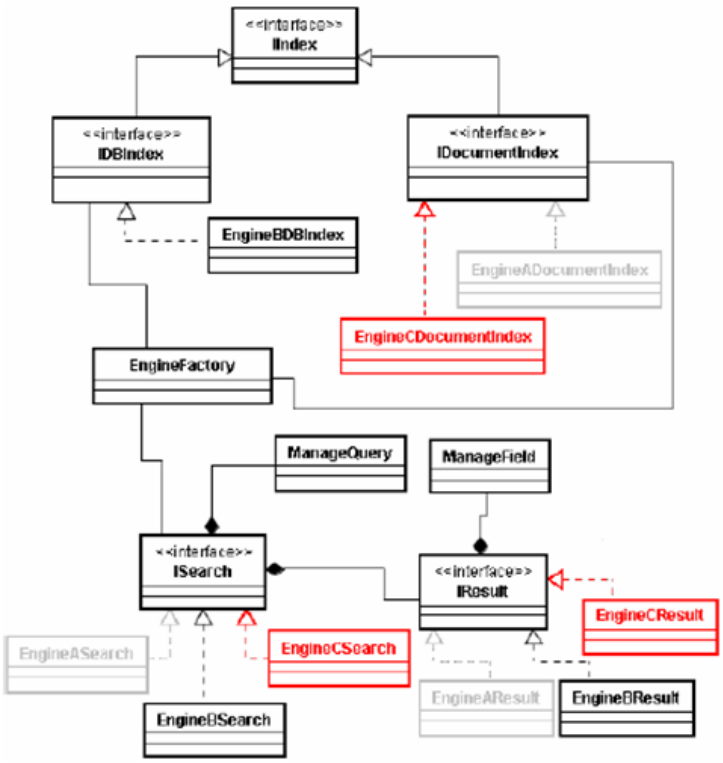

Figura: 5.9: Substituição de um motor de indexação no estrutura de integração

A extensão com maior complexidade é a adição da indexação de novas fontes de dados não definidas inicialmente.

Para tal, é preciso criar uma interface de indexação que herde as características de IIndex e que defina um padrão para a indexação dessa nova fonte de dados.

Feito isso, o restante do processo compreenderia praticamente os mesmos passos realizados para a incorporação de um novo motor à estrutura.

A única diferença é que a implementação do módulo de indexação deverá seguir a nova interface criada.

Novamente a vermelho, são apresentadas, na figura seguinte, as alterações necessárias para realização da inclusão da capacidade de indexa-

ção e busca de informações numa nova fonte de dados.

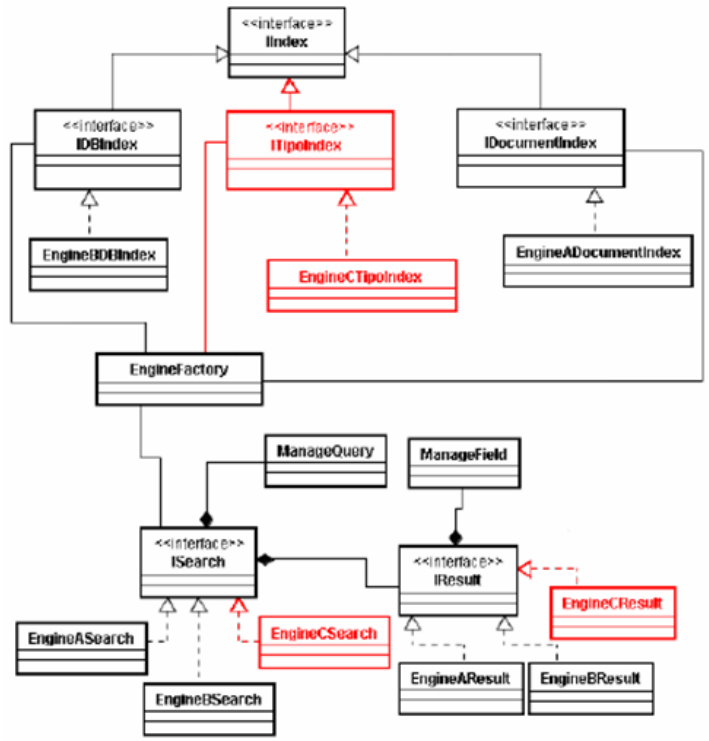

Figura: 5.10: Adição de um tipo de dados a ser indexado ao estrutura de integração

5.3 Um Exemplo de Aplicação

Uma aplicação-exemplo foi desenvolvida com o objectivo de rectificar as características e utilidades do modelo de integração proposto, bem como servir de avaliação prática do que foi apresentado.

A aplicação foi montada sobre a plataforma Web e é capaz de indexar, pesquisar e exibir dados de diversas fontes, de entre outras funcionalidades, como pode ser observado no diagrama de casos de uso apresentado na figura seguinte.

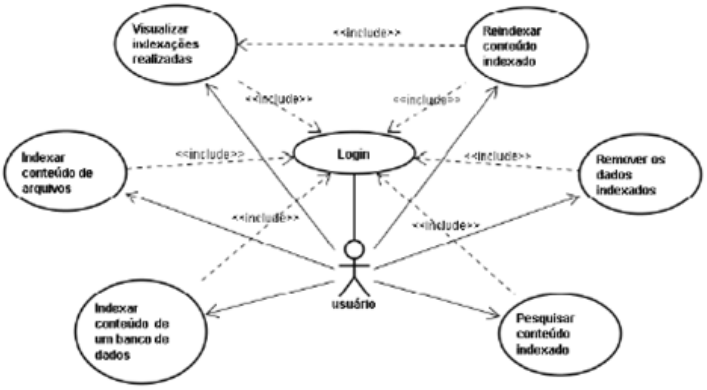

Figura: 5.11: Diagrama de casos de uso da aplicação exemplo

A aplicação exemplo faz uso de dois motores de indexação, o Windows Search e o Lucene. O primeiro é responsável por indexar arquivos comuns (tais como documentos, folhas de cálculo, apresentações, etc) e o segundo, por indexar dados provenientes de bases de dados relacionais.

A aplicação consiste em uma interface Web montada sobre o modelo proposto. O seu desenvolvimento foi dividido em etapas de forma a exemplificar cada passo da construção de uma aplicação sobre o modelo de integração e elucidar a possibilidade de extensão do modelo.

A etapa inicial consistiu na criação das classes do Windows Search, responsável por permitir a recuperação de informações provenientes dos arquivos comuns. Estas classes são WDSSearch, WDSResult e WDSDocumentIndex e implementam respectivamente as interfaces do módulo de busca ISearch e IResult e a do módulo de indexação IDocumentIndex, como pode ser observado a vermelho no diagrama de classes apresentado na próxima figura.

Além destas, também foram implementados os métodos da classe EngineFactory, responsáveis por instanciar os objectos de indexação e busca de informações do motor Windows Search.

A partir dessas implementações, foi concebida uma aplicação Web com o objectivo de proporcionar a interacção com o utilizador final. Esta aplicação permite que sejam realizadas indexações e busca de informações previamente indexadas. Para comprovar tal facto, foram indexados 360.543 arquivos em seis computadores interligados através de uma rede

local. A Figura 5.13 apresenta o resultado dessa mesma consulta realizada nesta aplicação após a indexação dos arquivos.

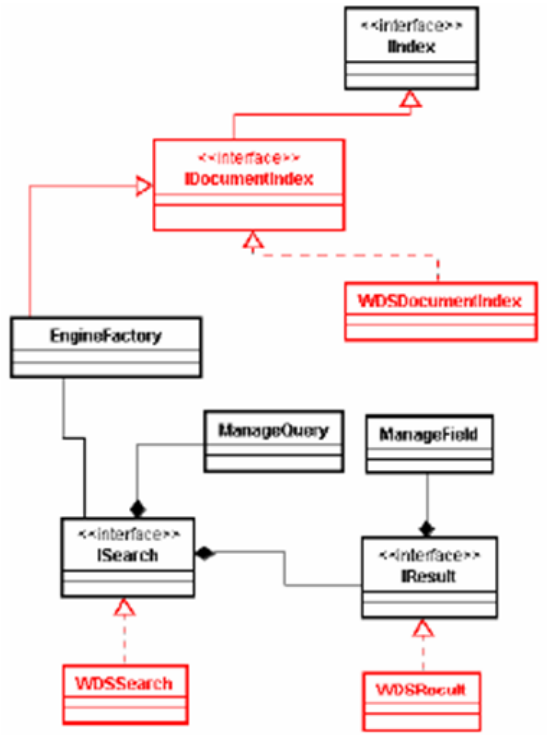

Figura: 5.12: Diagrama de classes da aplicação-exemplo com o motor Windows Search

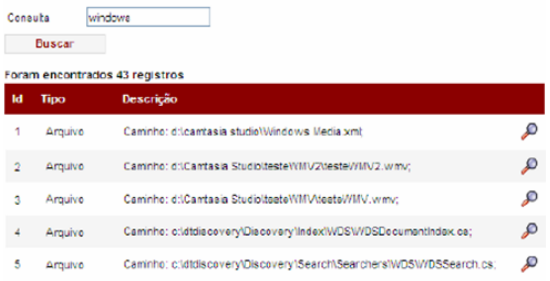

Figura: 5.13: Consulta realizada na aplicação exemplo com o motor Windows Search

A etapa seguinte no desenvolvimento da aplicação consistiu na inclusão do motor de indexação e busca Lucene com o intuito de indexar dados provenientes de um base de dados relacional. Para isso, as classes LuceneSearch, LuceneResult e LuceneDBIndex foram implementadas de acordo com as especificações das interfaces ISearch, IResult e IDBIndex, respectivamente. Estas modificações podem ser visualizadas em vermelho no diagrama de classes apresentado na figura seguinte.

Além disso, foram efectuadas as alterações necessárias na classe EngineFactory para que as novas classes pudessem ser instanciadas.

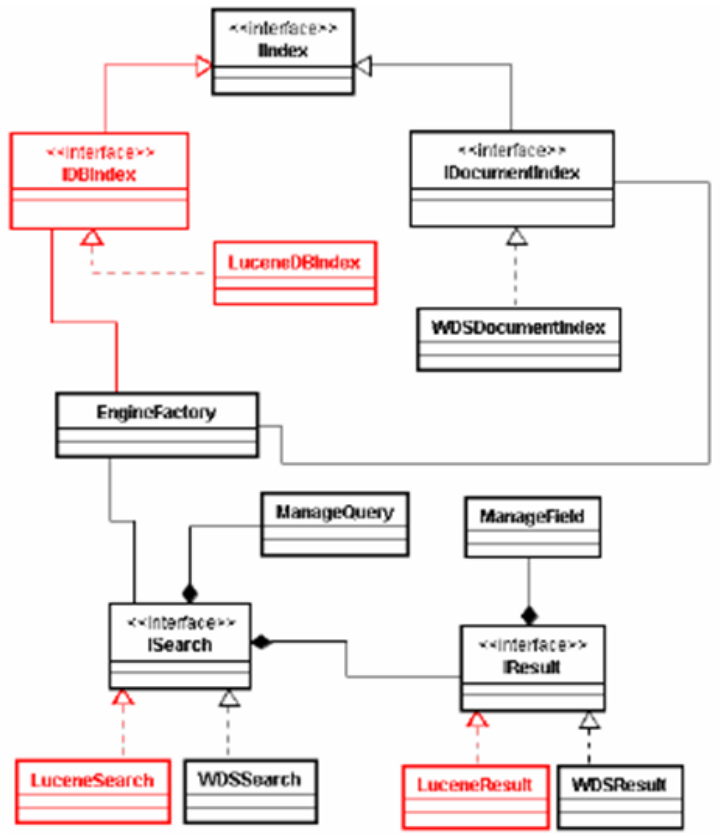

Figura: 5.14: Diagrama de classes da aplicação-exemplo com a inclusão do motor de indexação Lucene

Após esta extensão, foi indexada uma base de dados proveniente de um sistema de helpdesk armazenado em uma base de dados relacional.

Esta base de dados estava contida num servidor presente numa das seis máquinas que tiveram os seus arquivos indexados na etapa anterior do desenvolvimento.

Apesar da aplicação fazer uso dos dois motores de indexação após a extensão, a realização da busca é feita de forma integrada e transparente para o utilizador da aplicação, já que os resultados de quaisquer fontes de dados são padronizados e devolvidos numa só listagem, como pode ser observado na figura seguinte. Esta figura apresenta informações recuperadas do servidor de base de dados Harry em uma base de dados denominada DtHelp, e também dados provenientes de uma das máquinas utilizadas nas operações de indexação e busca. Estes resultados foram obtidos a partir de uma consulta executada através do mesmo campo de entrada de dados.

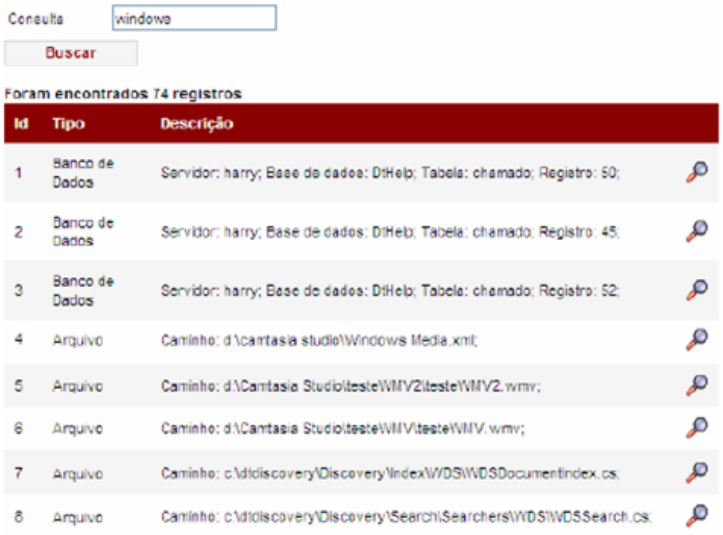

Figura: 5.15: Consulta realizada na aplicação-exemplo com a inclusão do motor de indexação Lucene

A listagem exibida pela Figura 5.15, assim como a da Figura 5.13, foi obtida através do trecho de código apresentado na figura seguinte. Percebe-se nesta figura que o uso da interface ISearch permite que sejam realizadas as pesquisas por todos os motores, o que deixa claro que o uso da estrutura proporciona um nível de abstração a ponto de não ser

necessário o conhecimento dos motores utilizados em cada pesquisa, nem quantos são eles.

Este facto fica claro na utilização do mesmo código para recuperação de dados ao se utilizar um ou mais motores.

A cada interacção, os resultados de cada motor são inseridos numa lista, que possui os resultados provenientes dos índices de todos os motores utilizados e, assim, a lista final de resultados é obtida.

```
1  //Cria a lista de resultados que será exibida
2  List<IResult> listResult = new List<IResult>();
3
4  //Cria uma nova instância da classe EngineFactory
5  EngineFactory engFact = new EngineFactory();
6
7   //Recupera a lista dos buscadores implementados na aplicação exemplo
8   List<ISearch> listSearch = engFact.CreateSearchEngines();
9
10 //Cria o objeto com os dados a serem buscados
11 ManageQuery mgQuery = new ManageQuery();
12 mgQuery.Sentence = TxtBoxQuery.Text;
13
14 //Recupera os resultados de cada buscador e os insere na lista
15 foreach (ISearch search in listSearch)
16 {
17   listResult.AddRange(search.Search(mgQuery));
18 }
```

Figura: 5.16: Trecho de código de realização de busca de informações

A partir do desenvolvimento e uso desta aplicação exemplo foi possível validar o modelo proposto, permitindo o uso integrado de dois ou mais motores de indexação, possibilitando também a independência entre a aplicação e os motores utilizados.

As buscas realizadas obtiveram retorno proveniente de diversas fontes de dados como base de dados relacional, arquivos estruturados em formato xml, arquivos de vídeo em formato wmv e arquivos de código em formato cs.

5.4 Indexação / Recuperação de Informação

Sabe-se que a geração de dados ocorre numa escala cada vez maior e a sua exploração é uma tarefa cada vez mais complexa devido à heterogeneidade das fontes de dados em que eles se encontram. Por isso, o aproveitamento das informações digitais acontece geralmente de maneira superficial. A solução desse problema encontra resposta parcial nos motores de indexação de dados existentes.

A indexação dos dados facilita o acesso aos mesmos, o que agiliza a recuperação de informações e reduz a quantidade de informação não aproveitada.

Contudo, apesar de possuírem características que contribuem para a solução do problema, esses motores não apresentam soluções completas, visto a actual procura dos dados provenientes de diversas fontes.

Neste contexto, o modelo de integração de motores de indexação apresentado permite o desenvolvimento de soluções que possibilitem a realização de recuperação de informações num ambiente de fontes de dados heterogéneas. Isto é possível devido à possibilidade de utilização de diversos motores responsáveis pela indexação e busca de dados em fontes específicas. Assim, por trabalharem em conjunto, estes motores permitem a indexação da quase totalidade dos tipos de dados gerados. O modelo ainda contempla a dissociação entre os motores de indexação e as aplicações desenvolvidas sobre ele. Isto torna possível a substituição de qualquer um desses motores, ou ainda, a incorporação de um novo, sem comprometer qualquer aplicação que já tenha sido desenvolvida sobre a estrutura anterior.

Após a avaliação da proposta com o desenvolvimento e uso da aplicação-exemplo, foi observado que o uso do modelo de integração cumpre o objectivo de possibilitar a utilização integrada de diferentes motores de indexação. Além disso, foi observada também a possibilidade de inclusão de novos motores de indexação às aplicações, sem a necessidade de alterações em partes destas aplicações. Assim, conclui-se que o modelo cumpre suas metas e possibilita o desenvolvimento de SRIs capazes de recuperar informações em ambientes de fontes de dados heterogéneas.

Vale a pena lembrar que a concepção desse modelo, não levou em consideração questões referentes ao desempenho na indexação e busca dos dados. Esta questão está fora do objectivo deste trabalho, uma vez

que a questão do desempenho está intrinsecamente relacionada com as características de cada um dos motores de indexação utilizados nos SRIs desenvolvidos sobre o modelo. No entanto, a utilização do modelo de integração não impede que sejam feitos testes de desempenho em SRIs desenvolvidos sobre ele.

Capítulo 6

Estudo de um Classificador TAN Incremental

6.1 TAN: Extensão ao Naive Bayes

O TAN foi introduzido por Friedman e Goldszmidt [Friedman & Goldszmidt, 1996] com o intuito de melhorar o naive Bayes [Rou, 2002], possibilitando representar dependências entre pares de atributos. Este classificador denomina-se por 'tree augmented naive' Bayes o que traduzido textualmente significa árvore aumentada do naive Bayes. No entanto, evitar-se-á a tradução, ficando apenas o esclarecimento.

Na figura 6.1 podemos observar o modelo para o naive Bayes e na figura 6.2 o modelo para o TAN. Nestas figuras a classe encontra-se representada por C e em ambas as figuras se encontram representados quatro atributos. Por exemplo, no caso da figura 6.1 podemos observar uma seta de C para X1, esta representa que o atributo X1 depende da classe.

O TAN é semelhante ao naive Bayes, permitindo no entanto dependências entre os atributos. As restrições são as seguintes: * cada atributo depende condicionalmente da classe (tal como ocorre no naive Bayes); * existem (n-1) atributos que dependem condicionalmente de um outro atributo.

Esta última condição implica que caso haja uma ligação de Xi para

Xj, estes dois atributos não são independentes dada a classe. Em vez disso a influência de Xj na probabilidade da classe C depende do valor de Xi. A figura 6.2 mostra o exemplo de uma rede Bayesiana com as condições descritas por Friedman para que seja considerada um TAN.

Comparando a rede da figura 6.1 (Naive Bayes) com a rede da figura 6.2 (TAN) verifica-se que foram acrescentadas algumas dependências entre os atributos.

De notar que a estrutura das dependências se encontra definida à partida para o naive Bayes. Para o TAN também se define que todos os atributos dependem da classe, mas é necessário encontrar as dependências entre os atributos seguindo as regras descritas. Para encontrar as dependências entre os atributos Friedman usou o algoritmo reportado por Chow e Lui [Hkl, 2002]. Uma vez que cada atributo pode ter no máximo um 'pai', é necessário encontrar aquele que possui maior probabilidade condicional dado o valor de C. De uma forma mais simples, é necessário encontrar os dois atributos que tenham maior correlação. Ao escolher para cada atributo o que tem maior dependência condicional dado C, obter-se-á, teoricamente, a melhor representação das dependências seguindo a regra do TAN.

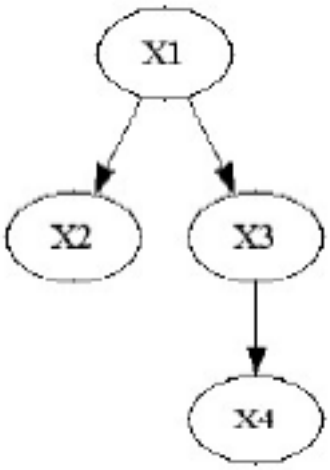

Figura: 6.1: Árvore de dependências encontrada pelo algoritmo de Chow and Liu.

6.1.1 Algoritmo de Chow and Liu

Se à figura 6.2, retirarmos a classe e todas as ligações entre os atributos e a classe, observa-se que as ligações entre os atributos formam uma árvore.

O algoritmo reportado em 1965 por Chow e Liu [Hkl, 2002] é utilizado para a construção de árvores de dependências que maximizam a informação mútua entre as variáveis. Assim, o algoritmo inicializa-se obtendo a informação que X fornece sobre Y , recorrendo à seguinte fórmula:

$$I_P(X;Y) = \sum_{x,y} P(x,y) \frac{P(x,y)}{P(x)P(y)}$$

Onde IP (X;Y) é a informação que X fornece sobre Y ou o inverso, informação esta que é calculada para todos os pares de atributos. Com a informação obtida entre todos os pares de atributos está-se em condições para construir um grafo completo.

A partir do grafo completo o objectivo é obter a árvore de dependências que maximize a informação mútua entre os atributos. Para achar essa árvore que maximiza o peso da informação mútua entre os atributos, existem dois algoritmos conhecidos, o algoritmo de Kruskal e o de Prim's.

Friedman ao analisar o caso específico da construção da árvore de dependências para o TAN verificou que, neste caso especifico, sabe-se à partida que todos os atributos são dependentes da classe. Assim, o necessário é conhecer a quantidade de informação que o atributo X fornece sobre o atributo Y dada a classe, utilizando uma fórmula que é uma adaptação da fórmula original (acima referida) para o caso específico do TAN:

$$I_P(X;Y|C) = \sum_{x,y,c} P(x,y,c) \frac{P(x,y|c)}{P(x|c)P(y|c)}$$

A diferença entre a primeira fórmula e a segunda é que esta última

assume que ambos os atributos são dependentes da classe, calculando deste modo a informação que X fornece sobre Y dado o valor da classe.

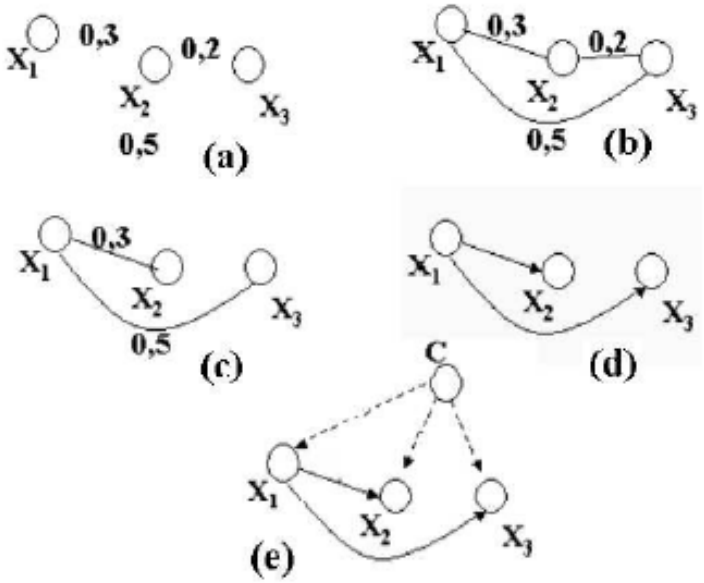

Figura: 6.2: Fases da construção da árvore que maximiza a informação mútua.

6.1.2 Construção da árvore de dependências

A construção da árvore de dependências é o que diferencia o TAN do naive Bayes. Esta vai determinar o desempenho do TAN. Teoricamente, é devido à dependência entre os atributos que o TAN melhora o desempenho em relação ao naive Bayes.

Para construir a árvore que maximiza a informação mútua entre os atributos, seguem-se os seguintes passos:

1. Obter a informação mútua, IP (Xi;Xj|C) entre cada par de atributos, i;j e construir um vector com essa informação mútua, figura 6.2, passo (a); 2. Desenhar um grafo completo não orientado, tendo como nós os atributos e como custo das ligações a informação mútua entre os atributos, figura 6.2, passo (b); 3. Calcular a árvore que maximiza a informação mútua, sem criar ciclos, figura 6.2, passo (c); 4. Transformar a árvore não orientada em orientada, escolhendo como raiz a informação

mútua mais alta, figura 6.2, passo (d); 5. Adicionar a classe como 'pai'de todos os atributos, figura 6.2, passo (e);

6.1.3 Variantes do TAN

Após a sua apresentação o TAN foi motivo de estudo por vários autores que apresentaram algumas ideias no sentido de tentar melhorar o desempenho do TAN de Friedman. As propostas mais relevantes são as seguintes:

- smoothed TAN foi apresentado por Friedman no mesmo artigo em que apresentou o TAN e este visa combater o problema de existirem poucos exemplos em algumas partições. Segundo Friedman [Friedman & Goldszmidt, 1996] este problema não afecta o Naive Bayes pois este dispõe de um menor número de partições e estas são elaboradas, apenas de acordo com o número de possíveis valores da classe que normalmente são bem representadas pelos dados de treino. Assim, para o cálculo do smoothed TAN são enviesados para a frequência marginal observada, formalmente:

$$\theta^s(x|\prod_x) = \alpha.\hat{P_D}(x|\prod_x) + (1-\alpha)\hat{P_D}(x)$$

- super parent (SP), apresentado por Keogh e Pazzani [Keogh & Pazzani, 1999], que propõem um algoritmo ambicioso que vai adicionando 'arcos' à estrutura do naive Bayes. Em cada passo o algoritmo adiciona mais um 'arco' mantendo a condição de que cada atributo não tenha mais de um pai. O algoritmo designa como órfãos os que não têm outro atributo como pai. No primeiro passo o SP inicializa a rede com o naive Bayes, em que todos os atributos são órfãos, assim a lista de órfãos é inicializada com todos os atributos.

  Em seguida o algoritmo avalia para cada atributo o desempenho como 'pai'para cada uma das listas de órfãos, entendendo-se por desempenho a taxa de erro nos exemplos de teste. O que obtiver melhor desempenho é escolhido como 'super pai'. De seguida escolhe-se o 'filho favorito', que corresponde ao que apresentou melhor desempenho como 'filho' do 'super pai' escolhido. O 'filho'

favorito é retirado da lista de órfãos e volta-se a escolher o novo 'super pai'. Este ciclo repete-se até que a lista de órfãos contenha apenas um elemento ou até que se verifique que o desempenho não melhora. . De uma forma resumida são estes os passos a seguir: 1. Inicializar a estrutura com o naive Bayes 2. Avaliar o desempenho da estrutura 3. Considerar a hipótese de tornar cada atributo como 'super pai', em que ASP representa o atributo que apresenta o melhor desempenho como 'super pai'. 4. Considerar as estruturas que acrescentam à actual um arco de ASP para os órfãos.

Se a melhor das estruturas anteriores melhorar o desempenho, considera-se essa estrutura e volta-se ao passo 2. Caso contrário retorna-se à estrutura actual.

Os autores do algoritmo defendem que este método de procura da árvore melhora o desempenho do classificador, mas este método é mais moroso e dispendioso na procura da estrutura do TAN. Além disso, existe ainda o problema de este algoritmo encontrar a melhor estrutura para os dados de treino o que não significa que seja a melhor estrutura para o problema.

Uma outra causa para que o algoritmo seja mais moroso é o facto de este necessitar de avaliar a rede com os dados de treino em cada um dos passos. Para além de avaliar a rede obtida o algoritmo avalia ainda todas as possíveis redes, o que vai prejudicar o tempo de obtenção do algoritmo, que é directamente proporcional ao número de exemplos de treino. Assim pode-se concluir que o desempenho do algoritmo diminui apesar de a taxa de erro para os dados de treino poder melhorar.

Deve-se no entanto salientar que este algoritmo trouxe uma importante contribuição que consiste na ideia de não impor ao algoritmo que crie uma árvore com (n-1) ligações mas que pare de adicionar ligações quando atinge um critério.

- TANC, apresentado por Fagiuoli e Zanalon [Fagiuoli & Zanalon, 2000] tenta estender o TAN a uma 'credal network '. Os autores apresentam na realidade dois algoritmos, o TANC que baseado numa 'credal network ' classifica os exemplos e o TANCs que para os autores deve ser utilizado em domínios com grande complexidade. Estes algoritmos utilizam intervalos de probabilidades em vez de valores precisos. Os autores reclamam que o tempo de processamento é linear de acordo com o número de atributos.

- TBMATAN, apresentado por Cerquides e Lópes Mántaras em 2003 [Cerquides, 2003], no qual os autores demonstram como induzir distribuições decompostas sobre o modelo TAN. Este algoritmo parte do princípio que ao se assumirem as probabilidades à priori como decompostas podemos é possível obter as probabilidades à posteriori também decompostas, podendo estas ser completamente determinadas analiticamente em tempo polinomial. No mesmo relatório os autores determinam que para grandes séries de dados o TBMATAN tem grandes inconvenientes, por isso propõem uma aproximação ao TBMATAN que denominam por SSTBMATAN. O TBMATAN e o SSTBMATAN apesar de demonstrado empiricamente pelos autores que podem melhorar os resultados do TAN, acrescentam uma complexidade considerável ao algoritmo, o que lhes retira parte das vantagens.

6.1.4 Algoritmos incrementais para o TAN

A aproximação Bayesiana à tarefa de classificação é eficiente e foi aplicada de maneira natural e com sucesso na área da aprendizagem on-line. No entanto, a utilização de extensões para a sua aplicação de modo incremental, onde a informação sobre exemplos já analisados não são guardados, têm sido muito limitadas até ao momento [Saa, 1998]. Como anteriormente mencionado, a mais simples das redes Bayesianas é o naive Bayes, que é naturalmente incremental [Rou, 2004]. Uma vez que nos problemas do mundo real existem dependências, quando estas são representadas a rede perde a natureza incremental pois, em redes Bayesianas, um novo exemplo pode afectar toda a estrutura [Friedman & Goldszmidt, 1997], [Rou, 2004], [Hec, 1997a]. Devido a esta natureza, a abordagem incremental das redes Bayesianas é mais difícil do que em algoritmos de outras áreas [Rou, 2004].

6.1.5 TAN-ACO de Roure

Roure na sua tese [Rou, 2004] dedicou-se ao estudo de algoritmos incrementais na área das redes Bayesianas. Roure descreve na tese uma heurística que, segundo ele, aplica-se a qualquer algoritmo que siga o hill climbing. Este baseia-se no princípio de que quando um algoritmo utiliza a procura em hill climbing, este segue um caminho, que é definido

por operações. A heurística de Roure guarda a informação e a ordem das operações que o algoritmo seguiu. Deste modo quando o algoritmo processa novos dados, a heurística verifica se o caminho seguido continua válido e caso este não seja valido a heurística reconstrói o caminho a partir da operação onde o caminho deixou de ser válido.

Mas para a exemplificação da sua heurística Roure escolheu a aplicação ao TAN, a heurística aplicada ao TAN Roure chamou-se de Arches in Correct Order (TAN-ACO). O algoritmo guarda a ordem das operações para chegar a árvore de dependências. A operação é neste caso a adição de um novo ramo à árvore de dependências. Considerando que temos uma série de operações ordenadas para chegarmos a um modelo (M),

$$M = op_n(op_{n-1}...(op_1(M_0, A_1), A_2)..., A_{n-1}), A_n)$$

Para chegarmos ao nosso modelo M, que neste caso é a nossa árvore de dependências, efectuamos uma série de operações sobre os argumentos. O que Roure propõe é que, quando chegam novos dados, se actualizem as partições da distribuição de probabilidades de modo a actualizar os argumentos. De seguida percorre-se o mesmo caminho para verificar se as operações continuam correctamente ordenadas. Assim que o algoritmo detecta que houve uma alteração na ordem das operações (equação seguinte), activa o processo para recalcular o caminho a partir dessa operação, o que faz com que o algoritmo encontre uma nova árvore de dependências.

$$op_j(M_{j-1}, A_j) \ \vee \ op_i(M_{i-1}, A_i) \ \vee \ \ i > j \ \vee \ A_i < A_j$$

ou seja, o argumento i passou a ser maior que o argumento j mas no entanto a operação i só foi efectuada depois da operação j.

Simplificando, o algoritmo guarda a ordem pela qual os arcos foram escolhidos para a árvore de dependências e esta escolha foi efectuada ordenando de modo decrescente o valor da informação mútua. Quando novos dados ficam disponíveis, o algoritmo actualiza a informação mútua e verifica se a ordem ainda é a mesma. Caso o algoritmo encontre alguma alteração, então activa o processo para a actualização da árvore na operação onde a ordem dos argumentos se alterou.

6.2 Algoritmos Incrementais

Os algoritmos incrementais são igualmente denominados por algoritmos actualizáveis [Saa, 1998]. A aprendizagem de modo naturalmente incremental é um dos objectivos da área de aprendizagem automática. Será agora abordado um algoritmo que, seguindo a metodologia do TAN, se comporta de modo incremental. Este mantém diversas estatísticas de forma a manter as estimativas de probabilidades e efectuar uma actualização à árvore de dependências quando chegam novos dados.

Será descrita a forma de como obter o algoritmo e prova-se, empiricamente, que este apresenta um desempenho muito semelhante ao TAN de Friedman [Friedman & Goldszmidt, 1996].

Será organizado do seguinte modo: 1. descrição do TANi, algoritmo que implementa o TAN de forma incremental. 2. considerações sobre o TANi, suas vantagens e limitações. 3. resultados experimentais do TANi, ao qual são efectuados diversos testes: (a) comparação do desempenho em termos da taxa de acerto; (b) comparação das árvores de dependências encontradas pelo TAN e pelo TANi; (c) análise da evolução do desempenho ao longo do treino em termos da taxa de acerto; (d) análise da sensibilidade do algoritmo ao número de instâncias dos conjuntos de exemplos apresentados em cada passo.

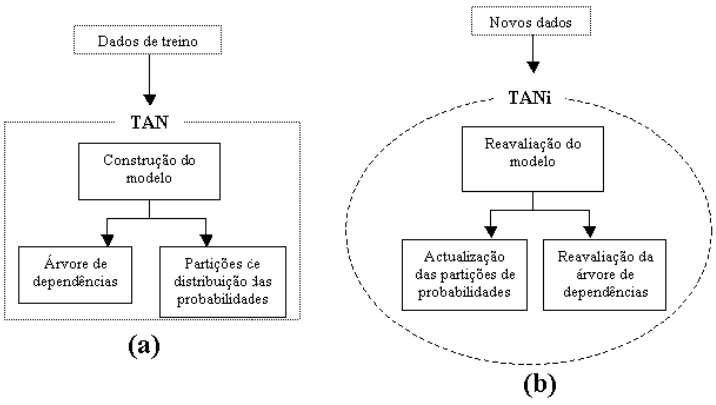

Figura: 6.3: Esquema do TAN (a) e do TANi (b)

6.2.1 Descrição de um TAN incremental (TANi)

O TANi tem por base o estudo do Roure [Rou, 2004]. Denominou-se o algoritmo por Incremental Version of Tree Augmented Naive Bayes (TANi).

Na primeira vez que constrói o modelo, o TANi processa-se como o TAN (figura 6.3 - (a)), no primeiro passo o TANi obtém a mesma estrutura que o TAN. No entanto o TANi pode receber mais conjuntos de dados após encontrar a estrutura que melhor se adapta ao conjunto de exemplos já analisado.

Para que o algoritmo possa receber mais conjuntos de dados, este contém duas estruturas fundamentais:

1. guarda a frequência dos valores por classe e por cada dois atributos, estas são chamadas na literatura de estatísticas de segunda ordem [Hec, 1997a];

2. A informação mútua entre todos os pares de atributos. Assim o algoritmo fica preparado para nos próximos passos actualizar o modelo com base nas estatísticas guardadas, ou seja, sempre que surgem novos dados o algoritmo actualiza as estatísticas de segunda ordem e a árvore de dependências que melhor que se adapta à nova distribuição de probabilidades.

A segunda fase do algoritmo pode ser separada em duas partes distintas:

1. Actualização das partições com as estimativas das probabilidades;

2. Obtenção da árvore de dependências que melhor se adapta a nova distribuição de probabilidades.

Na fase seguinte assume-se que quando o algoritmo analisa mais exemplos de treino, estes alteraram a distribuição de probabilidades. Este facto não implica que a árvore de dependências se altere. A análise da figura 6.3 permite uma melhor compreensão do algoritmo pois, esta esquematiza o seu processo.

Na figura 6.3 pode-se observar do lado esquerdo a metodologia do TAN, esta a partir de um conjunto de dados inicia a construção de um modelo. Modelo esse que gera uma árvore de dependências e um grupo de partições que contêm a distribuição de probabilidades do conjunto

de dados. Para o caso do TANi, este segue num primeiro passo o lado esquerdo da figura 6.3-(a), seguindo depois o esquema do lado direito da figura 6.3-(b). Quando chegam novos dados, o algoritmo dá início a reavaliação do modelo.

Esta reavaliação do modelo consiste na actualização do grupo de partições que guardam a distribuição de probabilidades e na geração de uma nova árvore de dependências entre os atributos, que melhor represente a nova distribuição de probabilidades.

Descreve-se de seguida cada um destes dois passos em pormenor, analisando primeiro a actualização das partições e de seguida a actualização da árvore de dependências.

6.2.2 Actualização das partições da estimativas das probabilidades

A manutenção de uma estimativa das probabilidades correcta implica manter contadores em cada uma das partições, ou seja, o algoritmo guarda a distribuição de valores de dois atributos por classe, entre todas as combinações de atributos. Estas estatísticas podem ser apelidadas de estatísticas de segunda ordem.

Como já descrito, o naive Bayes é um algoritmo incremental devido à sua natureza. Assim, a comparação dos custos da implementação da incrementalidade do TAN com um algoritmo da mesma área e naturalmente incremental permite uma melhor compreensão do custo que esta operação representa.

A análise da quantidade de informação que é necessário manter torna necessária a introdução de algumas notações.

Considerando Vi como o número de possíveis valores do atributo e VC o número de classes, se n for o número de atributos, para se obter o número de particões(np) que é necessário guardar temos:

naive Bayes: $V_c + \sum_{i=1}^{n} V_i \times V_C$

TANi: $V_C + \sum_{i=i}^{n} V_i \times V_C + \sum_{i=1}^{n} \sum_{j=1}^{n} V_i + V_j + V_C$

		Estatísticas	
V_i, V_j e V_C	n	1ª ordem	2ª ordem
3	3	30	273
3	6	57	1029
5	9	230	10355
5	10	255	12755
10	100	10010	10010010
10	200	20010	40020010
10	1000	100010	1000100010
10	5000	500010	25000500010

Figura: 6.4: Tabela de crescimento das estatísticas de primeira e segunda ordem

Na figura anterior podemos analisar o crescimento das estatísticas de primeira e segunda ordem à medida que o número de atributos e valores por atributo aumentam.

Se se comparar o número de contadores necessários para o naive Bayes com os necessários para o TAN verifica-se que o TAN necessita de guardar um maior número de partições. No entanto este número não torna proibitiva a sua utilização. Na literatura pesquisada existem soluções propostas que diminuem o tamanho ocupado por tais estatísticas e aceleram a pesquisa das mesmas. Para mais informação sobre o assunto consultar [Ml, 1998]. De notar que esta foi a mesma solução por que Roure optou no seu algoritmo.

6.2.3 Actualizar a árvore de dependências

A conclusão da actualização depende agora, apenas da obtenção da nova árvore de dependências entre os atributos. Uma vez que se trata de um algoritmo incremental, este é capaz de calcular a nova árvore de dependências sem recorrer aos exemplos já analisados pelo modelo. Por

essa razão quando se calcula pela primeira vez o TAN o procedimento é o seguinte: 1. Cálculo da informação mútua entre todos os pares Xi, Xj; 2. Construção da árvore de dependências.

Após a construção da árvore a informação mútua entre todos os pares de variáveis Xi, Xj é guardada num vector que se irá denominar por ID.

$$I''_D(X_i; X_j|C) = I_D(X_i; X_j|C) \times \frac{N}{N+N'} + I'_D(X_i; X_j|C) \times \frac{N'}{N+N'}, \forall_{i,j,i\neq j}$$

Quando novos dados chegam seguem-se os seguintes passos: 1. Cálculo da informação mútua, apenas para os novo conjunto de dados. 2. Guardar a informação mútua calculada no passo anterior num vector, denominado por I'D . 3. Somar o vector da informação mútua (IP) com o calculado no passo anterior (I'D) através da fórmula. 4. Calcular a nova árvore de dependências com o novo vector de informação mútua obtido no passo anterior.

De salientar que se irá denominar por N o número de exemplos e N' como o número de exemplos que se pretende adicionar à nossa base de conhecimento.

Uma vez que I"D é uma soma ponderada, a influência de cada um dos vectores de informação mútua é directamente proporcional ao número de casos que representa. O último passo é o recalcular da árvore de dependências com base no novo vector de dependências I'D.

De notar que o TANi segue a mesma metodologia que o TAN-ACO de Roure, ou seja, este é baseado na soma das informações mútuas entre cada par de atributos. As duas diferenças entre o TANi e o algoritmo descrito por Roure são:

1. O TANi efectua uma soma ponderada das informações mútuas, enquanto que o TAN-ACO apenas efectua a soma das informações mútuas. Em ambientes dinâmicos a soma simples tem vantagens sobre a soma ponderada pois com a soma simples da informação mútua o algoritmo reflecte mais rapidamente as alterações do ambiente, uma vez que os novos exemplos têm o mesmo peso na estrutura que os já processados. A soma das informações mútuas tem como vantagem uma aproximação a versão não incremental do TAN.

2. O TANi reconstrói sempre toda a árvore de dependências, enquanto que o TAN- ACO apenas reconstrói no ponto em que detecta uma alteração na ordem com que cada ramo da árvore de dependências foi introduzido. Reconstruir a árvore da raiz, apresenta a vantagem de esta teoricamente aproximar-se mais da árvore do algoritmo não incremental. A desvantagem é que este processo pode demorar mais tempo. Mas mais a frente vamos analisar este facto com maior detalhe.

6.2.4 Considerações sobre o algoritmo proposto

O algoritmo proposto segue a definição de incrementalidade proposta por Lanley [Lanley, 1995].

Num estudo sobre o custo associado à incorporação de novos exemplos na base de conhecimento, verifica-se que a segunda fase do TANi se divide em dois grandes cálculos (processamento): cálculo da informação mútua e construção da nova árvore de dependências.

- o cálculo da informação mútua tem um custo associado de O(n2.N) [Friedman & Goldszmidt, 1996], em que n é o número de atributos e N o número de instâncias, que é exponencial. No TANi o tempo de processamento gasto no cálculo do novo vector de informação mútua é apenas para os novos dados. Uma vez que este é directamente proporcional ao número de novos exemplos que se quer incorporar ao novo algoritmo, o tempo gasto para incorporar dois ou três exemplos é fixo e computacionalmente aceitável, tal como a definição de Langley sugere.

  Uma questão poderá surgir: Uma vez que se mantêm as estatísticas de segunda ordem, porque não se recalcula a informação mútua para todos os exemplos?

  No entanto esta pode não ser uma opção viável computacionalmente. Um computador dispõe de memória primária e secundária. Considera-se a memória primária a memória cache e RAM. Como memória secundária considera-se o disco rígido.

  As estatísticas de segunda ordem podem atingir valores na ordem dos megas ou mesmo gigas. Caso se calculasse a informação mútua para todos os casos ter-se-ia de recorrer às estatísticas de segunda

ordem, o que poderia tornar o processo de incorporação de poucos exemplos demasiado moroso, tornando-o incomportável. Contudo, com este algoritmo apenas é necessário manter em memória o resumo da informação mútua, o que é admissível para a memória primária, recorrendo pontualmente às estatísticas de segunda ordem, podendo estas estarem localizadas na memória secundaria. Estas apenas são necessárias quando a árvore de dependências se altera. E por cada ramo que se altera da árvore apenas é necessária a tabela de probabilidades entre os dois atributos e a classe. O que também é admissível computacionalmente.

- construção da nova árvore de dependências e o custo associado à construção da árvore é O(n2.logn) [Friedman & Goldszmidt, 1996] (onde n representa o número de atributos). Sendo o cálculo da árvore linear e independente do número de exemplos representados pelo modelo, o seu custo está relacionado com o número de atributos existentes, o que significa que é o mesmo quer se adicione um ou um milhão de exemplos ao modelo. Por esta razão a opção de recalcular a árvore é admissível quer em termos de tempo, quer em termos de peso computacional. De notar que o custo da construção da árvore depende apenas do número de atributos do conjunto de dados e o seu crescimento é linear.

Para a actualização da distribuição da probabilidade é necessário manter as estatísticas de segunda ordem e esta actualização pode-se revelar o maior requisito em termos de memória. No caso de conjuntos de dados com centenas ou milhares de atributos, este tipo de estatísticas pode necessitar de valores na ordem dos gigabytes, fazendo por isso sentido, apenas guarda-las em memória secundária.

O modelo TAN foi estendido para um modelo incremental (TANi), que reavalia o TAN quando chegam novos dados. No entanto, dada a sua natureza, o modelo apenas reavalia a árvore de dependências quando recebe um conjunto de dados com mais de um exemplo. Ou seja, uma vez que a informação mútua é calculada com base nos novos exemplos, se o novo conjunto de dados apenas apresentar um exemplo, cada atributo identifica o outro univocamente. Desta forma, a informação mútua entre os atributos é máxima, o que significa que não vai modificar o vector da informação mútua. Todavia, existem formas de contornar este problema. É possível estabelecer um número mínimo para proceder à actualização

da árvore de dependências.

6.3 Avaliação e Resultados

Esta secção tem por objectivo avaliar o desempenho do TANi. Para tal foram efectuados vários tipos de testes, utilizando conjuntos de dados do repositório de dados da Universidade da California, Irvine [Bm, 1998], que é um repositório de dados para a área de aprendizagem automática bastante usado na literatura. Os conjuntos de dados usados podem ser encontrados no site http://www.ics.uci.edu/ mlearn/MLRepository.html.

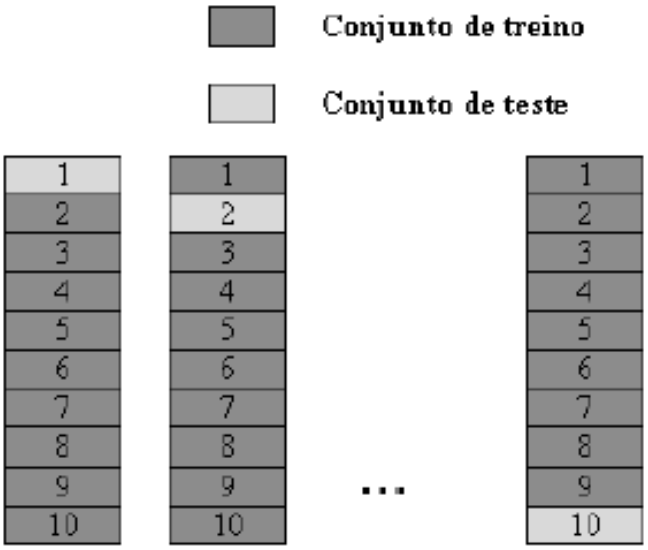

Figura: 6.5: Esquema da validação cruzada (com dez pastas).

Nas experiências foi utilizada a validação cruzada, de acordo com a descrita por Kohavi [Kohavi, 1995]. A validação cruzada pode ser estratificada ou não estratificada. A primeira cria pastas com um número de instâncias por pasta aproximado. A segunda escolhe aleatoriamente o número de instâncias por pasta.

Os dados (D) foram separados de forma aleatória em k pastas mutuamente exclusivas de tamanho semelhante, isto é, foi utilizada uma validação cruzada estratificada.

O classificador foi treinado e testado k vezes. De cada vez que se corre o modelo este é testado com Dt,t(1,...,k), e o treino foi o conjunto de dados D-Dt.

A estimação do erro foi calculada com base no total de instâncias mal classificadas em Dt a dividir pelo número de instâncias.

Repetir k vezes:

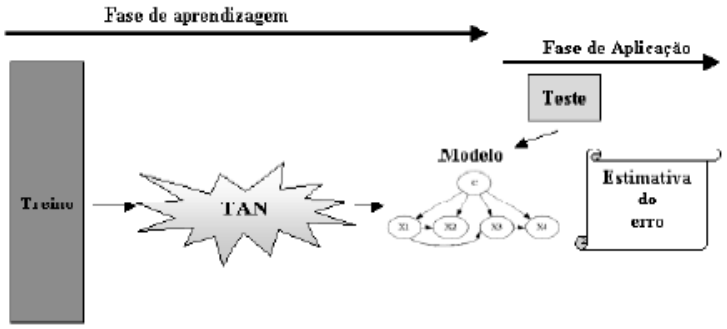

Figura: 6.6: Esquema da segunda fase da validação cruzada.

Uma vez que os algoritmos testados (TANi e TAN-ACO) são incrementais, estes são afectados pela ordem com que se apresentam os dados. Para contornar este problema, para os testes, os dados foram ordenados aleatoriamente.

Deverá ainda referir-se que as instâncias dos conjuntos de dados que continham valores em falta foram retiradas.

No esquema de validação cruzada pode-se analisar algumas características dos conjuntos de dados, como o número de atributos, número de possíveis valores da classe e número de instâncias do conjunto. Inicialmente comparam-se as estruturas das árvores gerada pelo TANi com as das árvores geradas pelo TAN. Posteriormente o desempenho do TANi é comparado com os algoritmos: naive Bayes, TAN, TAN-ACO, Bayes Network e uma implementação de uma árvore de classificação. De seguida procede-se a um estudo da evolução do desempenho do TANi à medida que este tem acesso a mais casos. As experiências terminam testando o impacto que tem na aprendizagem incremental a variação do número de exemplos por pasta.

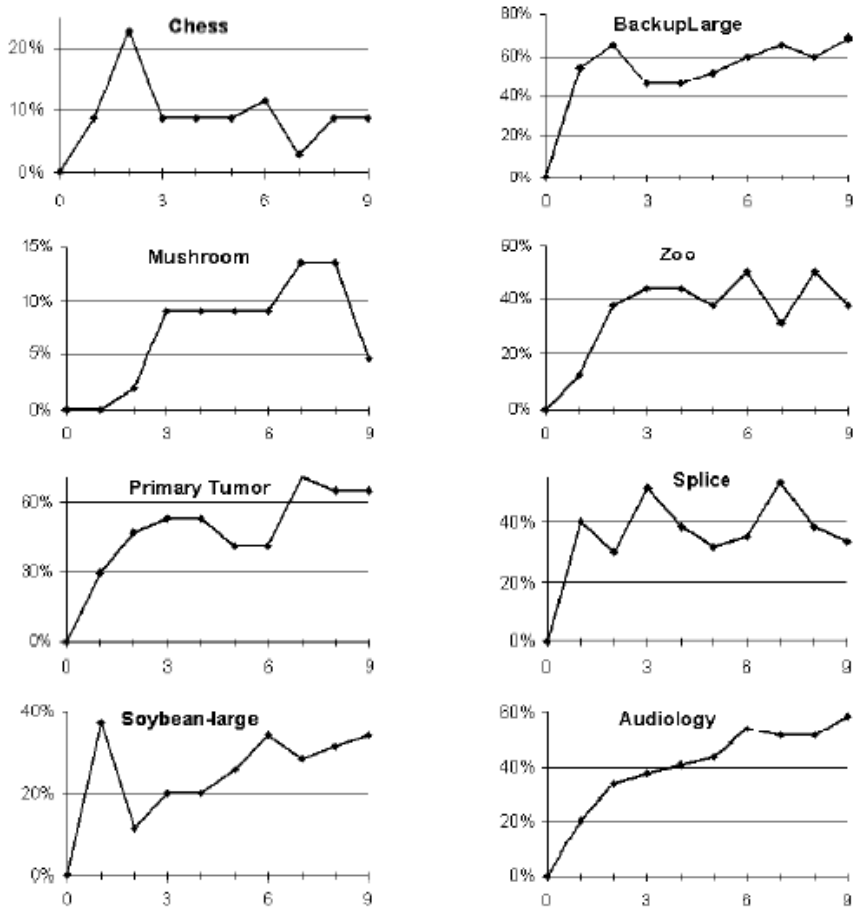

Figura: 6.7: Resultados da diferença ente as ligações na rede encontrada pelo TAN da encontrada pelo TANi. No eixo horizontal tem-se o número do passo (que dividindo o número de exemplos do conjunto de dados pelo número do passo obtemos o número de exemplos analisados) e no eixo vertical tem-se a diferença entre as árvores (em percentagem).

6.3.1 Estudo das árvores geradas pelo TANi

Como já foi referido, a diferença entre o naive Bayes e o TAN reside no facto de no naive Bayes partir do principio que os atributos são condicionalmente independentes entre si. Enquanto que o TAN encontra uma árvore de dependências em que (n-1) atributos dependem de outro, para

o naive Bayes todas os atributos são independentes entre si.

A construção do modelo TAN permite encontrar a árvore de dependências que representa as dependências entre atributos no conjunto dos dados, pois o TAN encontra de modo eficiente a árvore de dependências.

Nesta secção, compara-se a árvore de dependências encontrada pelo TANi, com a encontrada pelo TAN. Ambos os algoritmos tentam encontrar a árvore de dependências que melhor representa o conjunto de dados. O TANi efectua a procura num modo incremental e o TAN efectua o mesmo processo tendo acesso a todo o conjunto de dados. Este teste evidencia a diferença que existe entre a árvore encontrada incrementalmente e a encontrada tendo acesso a todo o conjunto de dados.

Este teste foi apenas efectuado para conjuntos de dados com mais de quinze atributos. Este é um espaço de procura suficientemente grande para que ambos os algoritmos encontrem árvores diferentes.

Para efectuar o teste os conjuntos de dados foram separados em dez partes (k1,k2,...,k10) e o processo separado igualmente em dez partes. Para cada um dos dez passos, foi usada a seguinte metodologia:

- TAN Em cada um dos passos o algoritmo era testado com o conjunto de dados kj , em que este conjunto é igual ao conjunto do passo anterior (kj¡1) mais o conjunto de dados desse passo (ki).

- TANi No caso do TANi, apenas se gerou uma vez o modelo, para o primeiro conjunto de dados. O modelo era em seguida reavaliado, adicionando ao modelo o conjunto de dados ki.

Em cada um dos passos a árvore gerada pelo modelo era guardada. De notar que no primeiro passo a árvore é igual para o TAN e para o TANi, logo a diferença entre as árvores é sempre zero no primeiro passo. A este passo seguem-se mais nove.

De salientar que na figura 6.7 observa-se o número de atributos que cada conjunto de dados contém. O eixo do x contém o número de passos e o eixo dos y a diferença em percentagem. Por exemplo, se se considerar a diferença de 50% e o conjunto de dados tiver 21 atributos significa que existem dez ramos iguais e dez ramos da árvore de dependências diferentes.

Tomando como exemplo o conjunto de dados do 'Chess', a diferença máxima encontrada entre os atributos foi de 23%, o que significa que existem oito ramos da árvore de dependências diferentes. A análise da figura que representa os resultados da diferença ente as ligações na rede encontrada pelo TAN da encontrada pelo TANi permite ainda verificar que na maior parte dos conjuntos de dados as diferenças entre as árvores se mantêm constantes ao longo das diversas etapas. Este facto apenas não se verifica para o conjunto de dados 'Audiology' em que a diferença vai sempre aumentando. Enquanto que, para o restante conjunto de dados a diferença mantém-se constante a partir de uma determinada etapa. Mesmo no caso do conjunto de dados 'Audiology' a curva parece querer estabilizar a partir da quinta etapa. De notar que o conjunto de dados 'Audiology' dispõe de 69 atributos e apenas 226 exemplos. O que significa que por pasta existiam aproximadamente 23 exemplos. Tendo em conta este facto, a diferença entre as árvores é atenuada. De notar que existem conjuntos de dados, nomeadamente o 'Chess' e o 'Mushroom' nos quais a diferença entre as árvores é muito baixa. Este facto pode indicar que as dependências entre os atributos são claras e que por essa razão ambos os algoritmos encontram praticamente a mesma árvore de dependências. Em conjuntos de dados como o 'BackupLarge' ou o 'Primary Tumor' as diferenças tomam valores mais altos, sugerindo que as dependências entre os atributos podem não ser tão claras. Analisando os gráficos como um todo, estes indicam que as diferenças não são substanciais. Assim, pode-se chegar à conclusão que no espaço de procura que existe, ambos os algoritmos têm um comportamento semelhante.

6.3.2 Análise do desempenho do TANi

Nesta secção pretende-se avaliar o desempenho do algoritmo proposto em termos da taxa de acerto. Para podermos ter alguma referência, o algoritmo foi comparado com o desempenho do naive Bayes (implementação do WEKA), rede Bayesiana (implementação do WEKA com as opções que vêm por defeito), árvore de classificação (implementação do algoritmo J48 do WEKA com as opções que vêm por defeito), TAN e TAN-ACO (implementação do autor). Infelizmente o programa do Roure (implementação do TAN-ACO) não permite correr um conjunto de dados de treino e de seguida testar o modelo com um conjunto de teste. O programa apenas permite dar um conjunto de dados e o próprio programa escolhe o conjunto de teste e o conjunto de treino, apresentando o resul-

tado final. Assim, os resultados apresentados representam o resultado obtido pelo programa do autor originando o total do conjunto de dados. De notar que a metodologia para a avaliação da taxa de acerto foi já descrita. Os conjuntos de dados seguidos com o sinal '*', são conjuntos de dados que contêm atributos contínuos. Como neste capítulo apenas se pretende testar o TANi, os atributos contínuos destes conjuntos de dados foram discretizados com a aplicação categorize do MLC++ [Ksd, 1996].

Conjunto de dados	Atributos	Instâncias	Classes	WEKA naive Bayes	WEKA J48	WEKA Bayes Net.	TAN-ACO	TAN	TANi
Hepatitis *	19	80	2	87.50 +/- 8.33	88.00 +/- 11.5	90.00 +/- 9.86	69.23	87.50 +/- 10.2	90.00 +/- 7.91
Hayes-rothdata	4	133	3	82.64 +/- 9.45	67.47 +/- 11.8	82.64 +/- 9.45	75.00	67.91 +/- 6.17	73.35 +/- 10.6
lymphography	19	148	4	46.67 +/- 11.2	50.67 +/- 14.8	47.33 +/- 11.8	36.73	43.81 +/- 11.1	42.33 +/- 14.7
Iris *	4	150	3	95.33 +/- 7.06	94.67 +/- 5.26	95.33 +/- 7.06	96.00	94.00 +/- 4.92	94.67 +/- 5.26
Wine *	13	178	3	98.89 +/- 2.34	93.30 +/- 5.10	97.71 +/- 4.05	79.66	96.63 +/- 4.70	97.19 +/- 3.96
Glass *	9	214	6	74.29 +/- 5.98	75.67 +/- 8.22	73.81 +/- 6.41	17.25	76.56 +/- 7.18	77.01 +/- 7.37
Audiology	69	226	24	70.43 +/- 6.47	76.52 +/- 6.40	75.71 +/- 6.20	80.00	64.78 +/- 13.7	66.52 +/- 12.9
Heart *	13	270	2	83.33 +/- 8.95	82.96 +/- 7.85	83.33 +/- 8.95	82.22	82.96 +/- 8.94	82.22 +/- 9.69
Cleve *	13	296	2	83.17 +/- 5.85	79.07 +/- 4.62	83.17 +/- 5.85	77.86	80.09 +/- 3.58	80.45 +/- 4.49
Solar	12	323	6	62.21 +/- 5.75	70.27 +/- 7.68	72.10 +/- 11.2	59.81	72.14 +/- 5.84	72.45 +/- 5.90
Liver-disorder *	6	345	2	63.39 +/- 6.07	59.60 +/- 10.1	63.67 +/- 6.30	55.31	62.97 +/- 6.70	62.69 +/- 6.20
Cars *	8	393	3	86.49 +/- 7.09	97.45 +/- 1.69	88.83 +/- 6.93	67.94	96.96 +/- 3.72	98.74 +/- 1.78
Soybean-large	35	562	15	91.64 +/- 4.10	91.47 +/- 3.72	92.00 +/- 4.28	82.35	91.65 +/- 4.24	90.76 +/- 4.40
BreastLoss *	32	569	2	97.51 +/- 2.08	95.76 +/- 2.32	97.51 +/- 2.08	99.11	96.63 +/- 2.60	96.78 +/- 2.67
Pima *	9	768	2	78.12 +/- 2.92	77.34 +/- 3.23	78.12 +/- 2.92	80.47	78.12 +/- 3.34	78.12 +/- 3.45
Diabetes *	8	768	2	77.99 +/- 3.93	77.21 +/- 5.59	77.99 +/- 3.93	79.69	79.16 +/- 3.12	77.73 +/- 4.00
Tic-tac-toe	9	958	2	69.94 +/- 4.13	86.01 +/- 3.13	69.94 +/- 4.13	61.88	76.31 +/- 4.33	76.72 +/- 3.85
Tokyo *	44	959	2	91.97 +/- 2.25	93.32 +/- 2.10	92.18 +/- 1.98	93.10	92.49 +/- 2.90	92.39 +/- 2.91
German *	20	1000	2	74.70 +/- 3.77	73.00 +/- 2.87	74.60 +/- 3.84	78.14	72.60 +/- 4.01	73.30 +/- 3.20
Car	6	1727	4	85.42 +/- 1.67	91.84 +/- 2.10	85.47 +/- 1.81	62.15	94.27 +/- 2.23	92.24 +/- 2.73
Splice	60	3190	3	95.36 +/- 0.72	94.55 +/- 1.33	95.42 +/- 0.74	93.51	95.33 +/- 1.07	95.30 +/- 1.10
Chess	36	3198	2	88.05 +/- 1.75	99.47 +/- 0.44	88.08 +/- 1.73	84.77	92.34 +/- 0.60	93.00 +/- 1.04
Waveform *	41	5000	3	80.72 +/- 1.20	76.28 +/- 1.18	80.72 +/- 1.21	82.95	80.28 +/- 1.54	81.98 +/- 1.36
Churn *	20	5000	2	87.92 +/- 1.13	93.54 +/- 0.60	87.94 +/- 1.08	76.43	90.98 +/- 1.65	87.54 +/- 1.29
Satdite Image *	36	6435	2	82.44 +/- 1.32	85.33 +/- 1.22	82.52 +/- 1.43	80.74	87.83 +/- 1.10	87.88 +/- 1.68
Mushroom	22	8124	7	64.44 +/- 1.21	62.23 +/- 1.81	64.76 +/- 1.21	30.69	65.66 +/- 1.68	66.15 +/- 1.45
Nursey	8	12961	5	90.24 +/- 0.68	97.19 +/- 0.27	90.25 +/- 0.66	62.64	93.46 +/- 0.69	92.45 +/- 0.88
Adult *	14	48222	2	83.72 +/- 0.45	86.29 +/- 0.38	83.76 +/- 0.45	88.23	85.85 +/- 0.53	85.68 +/- 0.48
Shuttle *	9	58000	7	99.45 +/- 0.15	99.91 +/- 0.04	99.46 +/- 0.14	99.91	99.94 +/- 0.04	99.93 +/- 0.04
Sleep	13	105908	5	68.22 +/- 0.30	72.35 +/- 0.56	68.22 +/- 0.30	67.68	73.17 +/- 0.37	73.25 +/- 0.32
Média da taxa de acerto				81.41	82.86	82.08	73.17	82.41	82.63
Ranking				5	1	4	6	3	2
Média do ranking				3.43	3.83	2.87	4.4	3.1	2.97

Figura: 6.8: Taxa de acertos dos algoritmos naive Bayes, rede Bayesiana, árvore de classificação, TANi e TAN-ACO para trinta conjuntos de dados.

Repetir 10 vezes:

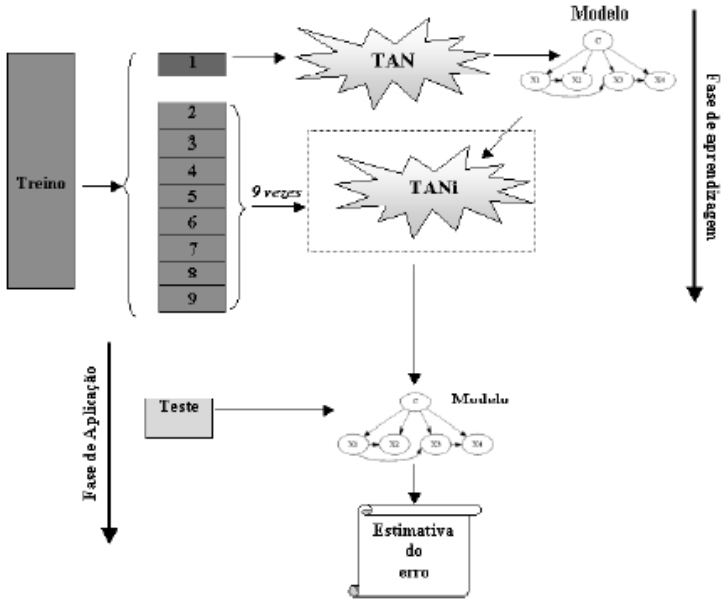

Figura: 6.9: Esquema da segunda fase das validação cruzada para os algoritmos incrementais.

Com excepção do TAN-ACO, todos os algoritmos correram as mesmas pastas estratificadas e em particular os conjuntos de teste foram os mesmos para todos os passos da validação cruzada.

Para o TANi, que reavalia o modelo, em cada passo da validação cruzada a metodologia utilizada foi a seguinte: 1. Os conjunto de treino foi divididos em dez pastas mutuamente exclusivas de tamanho aproximado (estratificadas); 2. Com a primeira pasta dos dados (escolhida aleatoriamente), foi construído o TAN com o algoritmo de Friedman; 3. As outras nove foram incrementadas, por ordem aleatória, ao primeiro modelo em nove passos;

Através da análise da figura anterior verifica-se que o naive Bayes consegue um melhor desempenho quando os conjuntos de dados contêm um menor número de exemplos. Este facto foi reportado por Friedman [Friedman & Goldszmidt, 1996]. Nomeadamente, o naive Bayes tem o melhor desempenho em quatro conjuntos de dados, todos eles fazem parte dos dez com menos instâncias.

A árvore de classificação do WEKA é a que melhor desempenho apresenta, podendo-se observar este facto na média da taxa de erro nos conjuntos de dados. Este algoritmo foi melhor em sete dos trinta conjuntos de dados.

O algoritmo que apresenta melhor média de ranking é a rede Bayesiana.

Conjunto de dados	Instâncias de treino	Instâncias de teste
Tic-tac-toe	480	468
Car	1150	577
Splice	2126	1064
Chess	2132	1066
Mushroom	5416	2708
Nursery	8640	4321

Figura: 6.10: Descrição da separação do conjunto de dados em teste e treino

Surpreendente é o facto de o TANi, apesar de ser incremental, melhorar os resultados do TAN em dezoito conjuntos de dados, sendo o melhor em sete destes. Além disso, é possível concluir que o TANi, em termos de desempenho, não se afasta muito do TAN, sendo este o objectivo principal.

Gostaria no entanto de realçar que para todos os testes de validação cruzada foi efectuado o teste de wilcox para verificar a diferença entre as médias da taxa de acerto. O teste Wilcoxon signed-ranks test / two-tailed test não encontrou diferenças significativas para nenhum conjunto de dados.

Este teste vem apenas comprovar que não existem diferenças significativas entre o TANi e a sua versão não incremental (TAN), não sendo por essa razão significativo o facto de o TANi ter obtido melhor taxa de desempenho. Mas o que realmente é um bom indicador é o facto de o teste de Wilcoxon não ter encontrado diferenças significativas.

6.3.3 Comparando a evolução do TANi, TAN

Nesta secção pretende-se avaliar a evolução do desempenho dos algoritmos à medida que estes têm acesso a mais instâncias de dados. Os algoritmos testados foram o TAN e o TANi.

Os conjuntos de dados seleccionados: Tic-tac-toe, Car, Slice, Chess, Mushroom, Nursery. Após a selecção do conjunto de dados, estes foram separados em dois conjuntos, treino e teste.

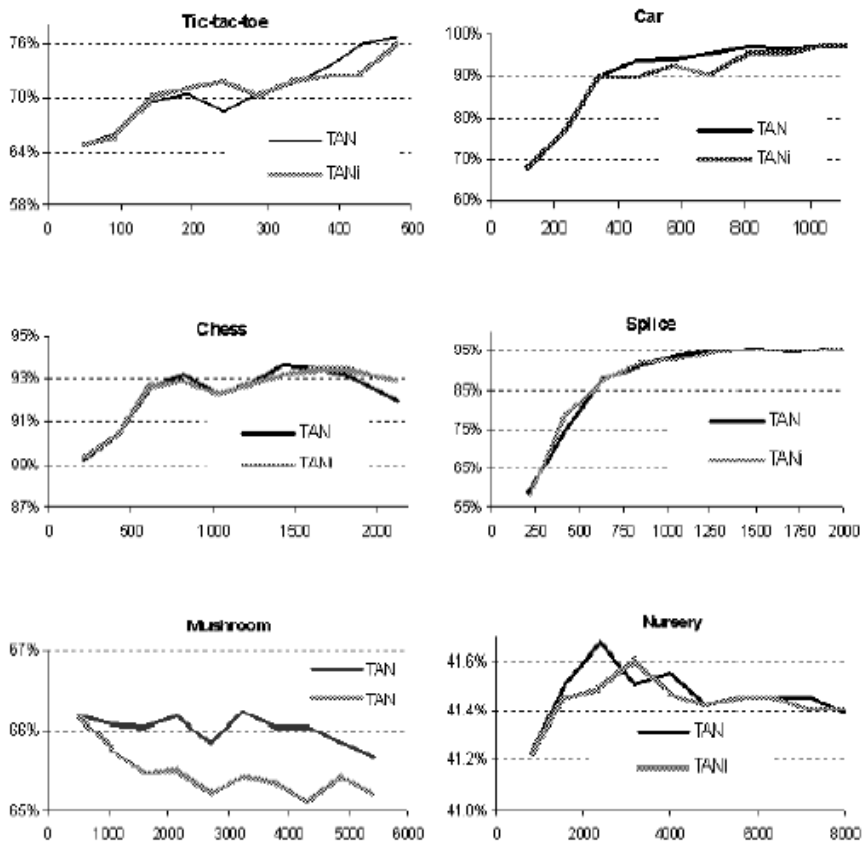

Figura: 6.11: Percentagem de acertos ao longo do treino do TAN e TANi com vários conjuntos de dados.

	exemplos por pasta	
Num pastas	Chess	Splice
1	2131	2127
10	$\simeq 213$	$\simeq 212$
100	$\simeq 21$	$\simeq 21$
500	$\simeq 4$	$\simeq 4$
1000	$\simeq 2$	$\simeq 2$

Figura: 6.12: Tabela com a descrição dos conjuntos de dados para o teste da sensibilidade incremental

Para o teste, as instâncias de treino foram separadas em dez pastas de tamanho aproximado e mutuamente exclusivas. Ao algoritmo incremental foram adicionadas as pastas. O algoritmo não incremental era reiniciado cada vez que se acrescentava uma pasta ao conjunto de pastas já observada pelo algoritmo.

Os resultados indicam que ambos os algoritmos têm curvas de aprendizagem bastante semelhantes, ou seja, tendem a subir a sua taxa de acerto a medida que têm acesso a mais exemplos.

De salientar que à medida que os algoritmos têm acesso a mais exemplos estes tendem a descrever curvas semelhantes, mesmo em conjuntos de dados em que o algoritmo não incremental baixa a taxa de acerto e o mesmo ocorre no algoritmo incremental. O que indica confirmar a hipótese de a árvore de dependências encontrada ser semelhante.

6.3.4 Análise da sensibilidade do algoritmo ao número de instâncias apresentadas por pasta

Esta secção tem por objectivo estudar o desempenho do algoritmo quando este analisa poucos exemplos de cada vez. Deste modo estuda-se a viabilidade de implementar o TANi como algoritmo online.

Conjunto de Dados	Treino	Teste
Splice	2131	1065
Chess	2127	1063

Figura: 6.13: descrição da separação do conjunto de dados em teste e treino

Os resultados do teste de sensibilidade incremental para os dois conjuntos de dados escolhidos, o 'Splice' e o 'Chess', têm diferentes comportamentos.

No caso do conjunto de dados 'Splice':

- este indica que não tem impacto no desempenho o facto de as pastas conterem poucos exemplos;
- todos convergem para a mesma taxa de erro;
- o algoritmo não incremental fica com uma taxa de erro superior aos modelos incrementais, mesmo quando este tem mil pastas.

No caso do conjunto da dados 'chess':

- o desempenho do algoritmo vai-se degradando à medida que o número de exemplos por pasta diminui;
- os diversos tamanhos de pastas obtêm taxas de acerto diferentes;
- A curva de aprendizagem acaba com uma taxa de acerto maior quando o número de instâncias por pasta é maior.

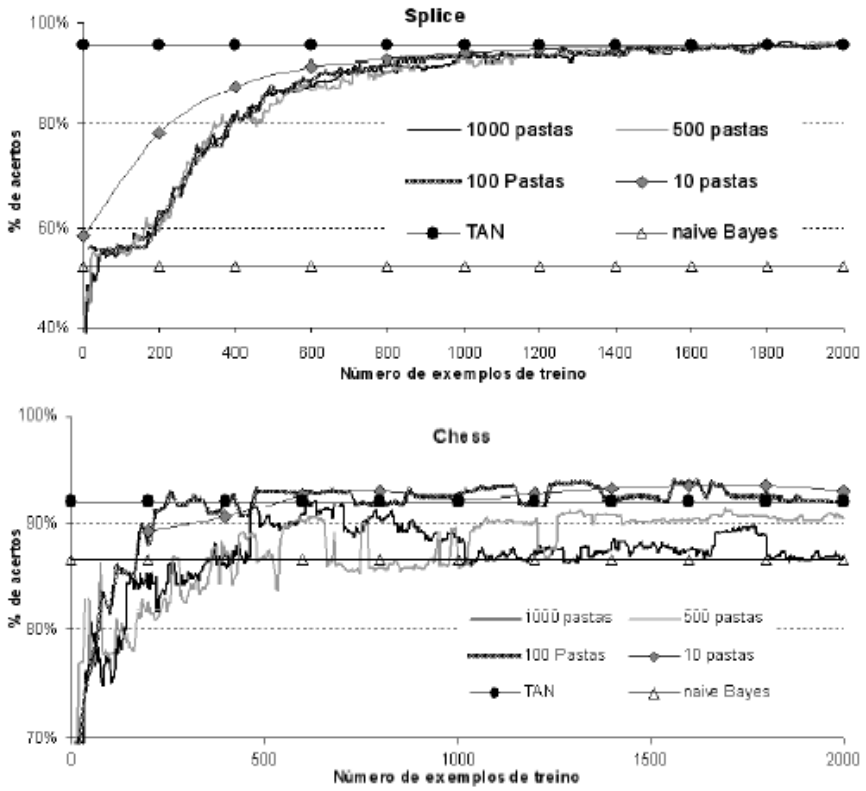

Figura: 6.14: Evolução da taxa de acerto do TANi quando treinado com diversos tamanhos de pastas para os conjuntos de dados 'Chess' e 'Splice'.

Denota-se que para o conjunto de dados 'Splice' o modelo final não apresentou diferenças significativas com diferentes tamanhos de pastas, o que não acontece no conjunto de dados 'chess', onde os resultados finais revelaram que existe uma diferença.

No conjunto de dados 'chess', o desempenho, apesar de se degradar por existirem poucos exemplos por pasta, nos casos em que o número de exemplos por pasta eram superiores a 20, foi superior ao algoritmo não incremental.

Este facto pode indicar alguma vantagem para o algoritmo incremental.

De notar, no entanto, que na experiência efectuada com mil pastas o modelo inicializou- se com dois exemplos.

Estas são condições extremamente difíceis para uma rede Bayesiana.

No entanto em qualquer um dos casos o algoritmo melhora o seu desempenho.

6.4 Abordagem Incremental à Discretização

A discretização fornece uma alternativa à estimativa da densidade da probabilidade. No cálculo desta estimativa é assumido um tipo de densidade, que se sabe à partida que não é a real mas sim aproximada [Dks, 1995], o que pode tender a degradar o desempenho dos algoritmos.

No entanto, alguns autores [Dks, 1995] [Yw, 2003] sugerem que a discretização em classificadores, tal como o naive Bayes, obtêm taxas de erro inferiores à estimativa da função de probabilidades.

Por outro lado, Catlett efectuou um estudo que lhe permitiu concluir que, para grandes conjuntos de dados, a discretização reduz significativamente o tempo de aprendizagem do modelo [Catlett, 1991].

Neste capítulo é apresentada uma nova abordagem ao método de discretização. Esta nova abordagem sugere uma metodologia que aborda a discretização de modo incremental, desta forma esta é reformulada à medida que mais conjuntos de exemplos são disponibilizados.

Este novo tipo de abordagem à discretização salienta duas das questões centrais para a área de aprendizagem automática: a aprendizagem incremental e a discretização como método de pré-processamento para alguns algoritmos incrementais.

O segundo ponto não é muito focado na literatura analisada. Assim, a partir deste ponto aponta-se a motivação para o desenvolvimento de uma discretização incremental. De seguida introduz-se uma nova abordagem à discretização, o Pré-processamento incremental de Discretização (PiD), que consiste numa metodologia que pode ser utilizada quer como discretização supervisionada, quer como discretização não supervisionada. O trabalho prossegue com a avaliação experimental do método, terminando com algumas considerações aos resultados e ao método proposto.

6.5 Motivação

Os humanos procedem a uma discretização que é facilmente comprovável pois, quando uma pessoa descreve uma outra, tende a discretizar certos atributos da pessoa, nomeadamente a altura, a idade e o peso. Se se disser, por exemplo, que o João é alto, novo e magro está-se a proceder à discretização dos três atributos mencionados. No entanto, é possível chegar mais longe e afirmar que esta discretização se altera ao longo do tempo de vida da pessoa. Assim é possível afirmar que os humanos dispõem de uma discretização incremental. Se se pedir a um rapaz de quinze anos, por exemplo, para definir o que é para ele um jovem, o rapaz certamente responderá que considera um jovem uma pessoa com menos de vinte e cinco e um velho uma pessoa com mais de cinquenta. No entanto, verifica-se que a sua percepção de jovem ou velho se alterará com a sua experiência e idade. Caso se pergunte à mesma pessoa dez anos mais tarde o seu conceito da idade ter-se-á provavelmente alterado, uma vez que o seu limite de juventude e velhice se alteraram. Quer-se com este exemplo ilustrar que a sua discretização da idade se altera ao longo da vida, na medida em que as pessoas ajustam a discretização às suas vivências e experiência de vida. Outros atributos também sofrem alteração pois, se perguntarmos a um jovem se uma botija de gás é pesada, este responderá provavelmente 'muito'. No entanto, se este na sua fase adulta se tornar num culturista e lhe for colocada a mesma questão ele responderá provavelmente 'nem por isso'.

Desta forma, uma das motivações é novamente a observação dos humanos e a incorporação das suas capacidades em algoritmos da aprendizagem automática. A literatura conhecida refere-se aos algoritmos incrementais como uma evolução natural na área da aprendizagem automática. No entanto, na maioria dos trabalhos quando se procede à avaliação de um algoritmo incremental, que não trabalhe com atributos contínuos, os autores optam por discretizar previamente o conjunto de dados [Rou, 2004].

O processo de discretização utilizado na literatura com algoritmos incrementais, divide-se em quatro fases: 1. Discretização do conjunto de dados com o método escolhido; 2. Obtenção dos dados discretizados; 3. Separação do conjunto de dados em k pastas para o algoritmo incremental; 4. Teste ao algoritmo incremental em k passos.

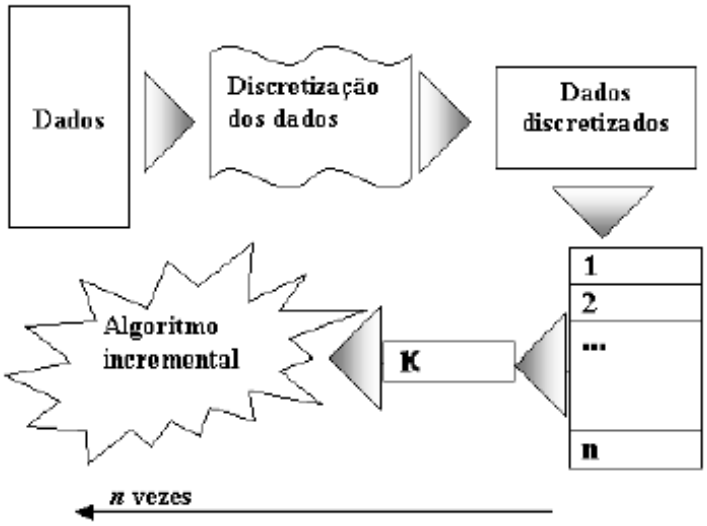

Figura: 6.15: Processo de discretização em modo não incremental

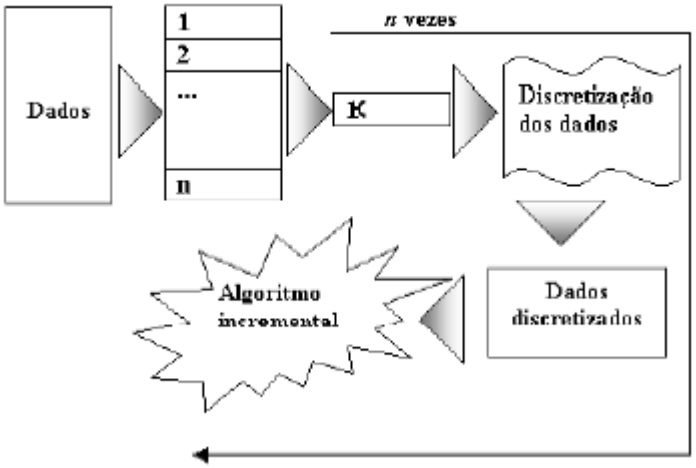

Figura: 6.16: Processo de discretização em modo incremental

Deve-se salientar que com o teste dos algoritmos se pretende analisar o seu desempenho, para que seja possível deduzi-lo no futuro e numa situação real. No entanto, numa situação real o algoritmo não terá acesso prévio a todos os dados, o que indica que os resultados obtidos podem

ser enviesados. Assim, se a discretização for efectuada à medida que o algoritmo tenha acesso aos dados, a discretização resultante seria diferente logo, os seus resultados também seriam diferentes. Contudo, os respectivos autores apenas pretendem testar o algoritmo, ignorando este aspecto.

Actualmente a discretização é encarada como um pré-processamento mas a verdade é que esta tem um papel fundamental para o bom desempenho do algoritmo. Da sua escolha depende a obtenção dos resultados, quer seja incremental ou não. Assim pretende-se obter um algoritmo que proporcione uma discretização incremental, ou seja, que caso seja necessário acompanhe a alteração dos dados.

Existem diversas motivações para o uso de algoritmos incrementais, sendo uma delas a existência de ambientes dinâmicos. Nestes ambientes sobressaem as vantagens da utilização de uma discretização incremental, pois esta pode evidenciar as alterações do ambiente, facilitando assim a interpretação por parte do algoritmo para a reformulação do modelo.

Preço	1990	1997	2004
500 €	barato	barato	barato
1000 €	barato	normal	normal
1500 €	barato	normal	caro
2000 €	normal	caro	caro
2500 €	caro	caro	caro

Figura: 6.17: Discretização do preço de um computador ao longo do tempo

Actualmente a discretização incremental em ambientes dinâmicos possui uma elevada importância. Imagine-se, por exemplo, as aplicações que presentemente existem para a extracção de conhecimento a partir de grandes bases de dados. Estas bases de dados podem conter as preferências dos consumidores de supermercados possuindo assim milhões de registos. A título exemplificativo pode-se mencionar empresas tais como a Sonae ou a Jerónimo Martins que detêm bases de dados deste tipo. No

entanto, os preços dos produtos são um atributo contínuo, devendo-se proceder à discretização do mesmo.

Se se assumir que as ferramentas para a extracção do conhecimento procedem a uma discretização do atributo, quando esta é actualizada, é necessário proceder a um novo processamento dos dados. Esta situação implica um novo gasto de recursos e tempo. Com a utilização de um método de discretização incremental, este novo gasto de recursos e tempo não existe.

Com a discretização incremental espera-se ainda obter uma melhor interpretação dos dados para o algoritmo pois, como já mencionado um dos argumentos para a utilização da discretização é a simplificação dos dados e a obtenção de melhores modelos de classificação. Com uma discretização incremental é possível obter uma correcta interpretação dos dados que variam ao longo do tempo, o que pode implicar que os valores do atributo, em alturas diferentes, possam pertencer a diferentes intervalos. Se se tomar por exemplo uma aplicação que pretenda extrair o hábito de compra de computadores por extractos da população e supondo a discretização do preço dos computadores da figura 6.17, apesar de para o mesmo valor obtermos diferentes discretizações, a análise assentaria em bases correctas. Caso se aplicasse a mesma discretização ao longo do tempo estar-se-ia a induzir o algoritmo a resultados errados. No entanto, no caso de se ter implementado uma discretização incremental em 1990, esta poderia ter obtido os resultados descritos na tabela.

Logo, uma das grandes vantagens da discretização como método incremental é o facto de esta reflectir a mudança no ambiente, enquanto que uma discretização não incremental pode induzir a resultados errados. Deve-se no entanto realçar que no caso de não ocorrerem mudanças no ambiente, a discretização incremental continua a obter resultados correctos. A discretização incremental adapta-se assim as duas situações, enquanto que a discretização não incremental se restringe à primeira.

6.6 Pré-processamento incremental de Discretização (PiD)

A discretização é um pré-processamento utilizada como preparação dos dados, existindo vantagens numa abordagem simples, rápida e 'leve'

computacionalmente. Estas vantagens acentuam-se se se considerar uma discretização incremental, pois o método de discretização é chamado sempre que o algoritmo tiver acesso a mais dados, enquanto que uma discretização não incremental apenas é efectuada uma vez.

O PiD é composto por duas camadas, em que cada uma delas tem uma função diferente. O método pode ser aplicado de modo supervisionado e não supervisionado. O ultilizador deve definir o modo como deseja correr o método antes de inicializar a primeira camada.

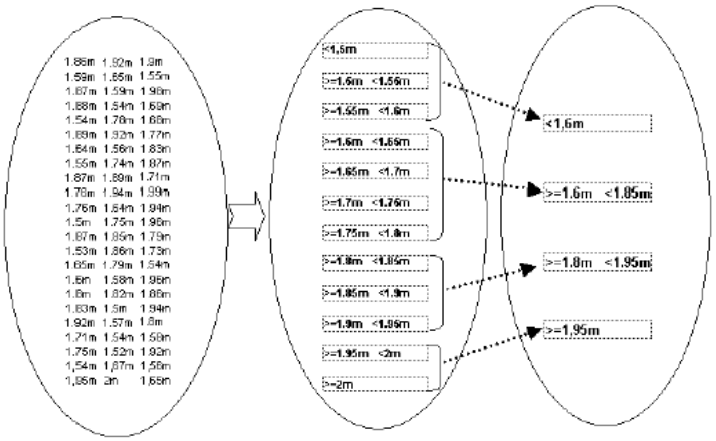

Figura: 6.18: Processo de discretização não supervisionado

Cada uma das camadas é composta por:

1. A primeira contém uma discretização dos dados muito fina. Esta camada é composta pelos limites dos intervalos achados para a discretização. Caso se aplique o método como não supervisionado esta camada guarda a frequência de valores por intervalo, caso se tenha aplicado como supervisionado então a camada guarda a frequência de valores por classe.

2. A segunda também contém uma discretizacao, esta é calculada com base nas estatisticas guardadas na primeira camada. Esta discretização é a discretização final que fornece ao algoritmo o exemplos do conjunto de dados discretizados.

Inicialização de cada uma das camadas:

1. A primeira camada tem duas possibilidades de ser inicializada,

esta pode ser inicializada sem nunca ter visto os dados ou esta pode ser inicializada com o primeiro conjunto de dados. No primeiro caso, deve ser dado o limite de cada um dos atributos e o número de intervalos desejado, é então aplicado o EWB a cada um dos atributos. Deste modo efectua-se a discretização desta camada. No segundo caso, deve ser dado qual o método de discretização a utilizar e o primeiro conjunto de dados. No caso do método ser não paramétrico (chi-merge, ENT-MDL, ...), deve-se parametrizá-lo de modo a criar mais intervalos que os necessários numa discretização final. Deve-se ter em atenção que esta camada deve criar mais intervalos que a discretização final. Segue o conselho de se criar no mínimo 100 intervalos.

2. A segunda camada é inicializada a primeira vez que se corre o método. Deve ser definido pelo utilizador qual o método a utilizar para a discretização dos dados. Caso este seja paramétrico o utilizador deve ainda fornecer o número de intervalos a criar, caso o método seja não paramétrico o utilizador deve fornecer o número mínimo de intervalos a criar. O método é inicializado a primeira vez que recebe os dados da primeira camada.

Actualização de cada uma das camadas:

1. Quando a primeira camada tem acesso a novos dados esta actualiza a frequência dos intervalos, ou seja, caso o método tenha sido aplicado de modo não supervisionado apenas necessita de actualizar os valores do atributo por intervalo. Caso o método tenha sido aplicado de modo supervisionado actualiza os valores dos intervalos por classe.

2. A segunda camada é processada depois da primeira camada. Como a primeira camada actualizou a frequência de valores foi gerada uma nova distribuição de valores, podendo deste modo originar uma alteração dos limites dos intervalos nesta segunda camada. Para actualizar os limites dos intervalos pode-se proceder de dois modos: (a) Reinicia a segunda camada, correndo novamente o método de discretização, com a condição de originar o mesmo número de intervalos. (b) Aplicar uma heurística para a reformulação dos limites dos intervalos com base na nova distribuição de valores.

Simplificando, o método divide-se em duas camadas, cada uma das camadas contendo uma discretização. A primeira camada tem como função simplificar os dados e guardar as estatísticas da distribuição de frequência de valores. A segunda camada tem como função aplicar a

discretização aos dados.

O número de intervalos para cada uma das camadas é definido na sua inicialização. Como é descrito, caso se opte por um método de discretizaçao não supervisionado na segunda camada, então a primeira guarda a frequência do atributo por intervalo. No caso de o método de discretização da segunda camada ser supervisionado, a primeira guarda a frequência de valores do atributo por classe em cada intervalo. No entanto, existem métodos de discretização que necessitam de mais do que a frequências dos valores. Neste estudo vamo-nos restringir a métodos não supervisionados e métodos supervisionados univariados (que apenas necessitam da frequência do atributo por classe).

De notar que o método reavalia os intervalos gerados para a discretização. Para efectuar esta reavaliação o método precisa de guardar a frequência de valores por intervalo (no caso não supervisionado) ou a frequência de valores em cada intervalo por classe (no caso supervisionado). No primeiro caso, as estatísticas a guardar são irrelevantes, no segundo caso as estatísticas a guardar crescem linearmente, por essa razão cumpre as definições de Langley. O método de discretização encontra-se representado na Figura 6.18 onde é possível observar uma discretização não supervisionada efectuada em dois passos: uma primeira discretização muito 'fina', com bastantes intervalos, e uma segunda discretização com os intervalos necessários para o algoritmo.

O passo intermédio da primeira camada serve como plataforma para que seja possível ao método reformular a discretização quando surgem novos dados e o método actualiza a frequência de valores dessa mesma camada, gerando, deste modo, uma nova distribuição de frequências. Tendo como base esta nova distribuição de frequências o PiD actualiza os limites dos intervalos da segunda camada, reformulando a discretização gerada anteriormente ou gerando uma nova discretização. O novo conjunto de dados é então discretizado e disponibilizado ao algoritmo pela nova discretização gerada para a segunda camada.

De salientar que o PiD converge para qualquer método de discretização não incremental. A verificação desta convergência pode ser realizada utilizando o Equal-with Binning na primeira camada e o método desejado na segunda e corre-se o algoritmo não incremental. Para obter o mesmo resultado com o PiD basta ir aumentando o número de intervalos da primeira camada, o que provoca uma diminuição do número de valo-

res diferentes por intervalo. No pior cenário a convergência dá-se quando houver um número de intervalos tal que permita que cada intervalo da primeira camada contenha o máximo de um valor diferente do atributo. Deste modo, é equivalente efectuar a discretização não incremental sobre os dados ou correr essa mesma discretização na segunda camada. No entanto, o mais natural é que a convergência se dê antes. Isto implica que o PiD pode convergir para a discretização não incremental à medida que a frequência de valores por intervalo da primeira camada diminui. O grande ganho do PiD reside no facto de encontrar o número de intervalos para a primeira camada que permite simplificar os dados, sem prejudicar a discretização da segunda camada. Pode-se ainda concluir que o PiD simplifica os dados com a primeira camada, encontrando em seguida a discretização que melhor se adapta à estatística dos dados. Uma discretização como o ENT-MDL ou o Chi-Merge apresentam algum peso de processamento se estivermos a falar de milhões de registos. Com o PiD a discretização dos dados pode ser acelerada em relação a qualquer um destes métodos, pois a primeira camada simplifica os dados com um método simples tal como o Equal-with Binning e o Equal-frequency Binning, acelerando deste modo a discretização da segunda camada (ENT-MDL ou o Chi-Merge). Por conseguinte o PiD possui quatro principais vantagens:

1. simples - não necessita de grandes requisitos de memória pois, mesmo quando aplicado a grandes bases de dados os seus requisitos de memória são bastante modestos.

2. leve - não requer grandes custos a nível computacional, dependendo no entanto dos métodos escolhidos para cada uma das camadas.

3. incremental - esta é a principal novidade no método de discretização. O método divide-se em dois passos precisamente para alcançar este requisito.

4. adaptável - o método adapta-se a qualquer algoritmo, pois este serve apenas como pré-processamento dos dados.

Com um exemplo ilustra-se os resultados obtidos utilizando o PiD com uma discretização não supervisionada e com uma discretização supervisionada. Escolheu-se o conjunto de dados Iris por este ser bastante utilizado na literatura como exemplo para a discretização. Escolheu-se o conjunto de dados Íris, que contém aproximadamente 150 instâncias. Aplicou-se o PiD de modo incremental com 5 pastas de 30 instâncias

cada. Para a primeira camada aplicou-se o EFW com treze pontos de corte, quer no método supervisionado, quer no método não supervisionado. De seguida, para o método não supervisionado, aplicou-se novamente o EFW na segunda camada apenas com um ponto de corte, o que tornou bastante leve a aplicação do PiD. Para o método supervisionado aplicou-se o ENT-MDL na segunda camada.

Primeira camada com *equal frequency discretization*

Atributo	Pontos de corte	Num. de intervalos
At1	4.95; 5.15; 5.35; 5.45; 5.55; 5.65; 5.75; 5.86; 5.95; 6.25; 6.40; 6.80	13
At2	2.35; 2.45; 2.60; 2.75; 2.95; 3.05; 3.15; 3.25; 3.35; 3.45; 3.60; 3.75	13
At3	1.35; 1.45; 1.55; 2.60; 3.80; 4.10; 4.30; 4.55; 4.80; 5.05; 5.45; 5.65	13
At4	0.25; 0.70; 1.05; 1.15; 1.20; 1.25; 1.35; 1.45; 1.60; 1.75; 1.95; 2.15	13

Segunda camada com *equal frequency discretization* (Discretização não supervisada)

Atributo	Pontos de corte após ...					Num. de int.
	30 instânc.	60 instânc.	90 instânc.	120 instânc.	150 instânc.	
At1	5.55	5.65	5.65	5.75	5.75	2
At2	2.95	2.95	2.95	2.95	2.95	2
At3	4.30	4.10	4.10	4.10	4.30	2
At4	1.25	1.25	1.25	1.25	1.25	2

Segunda camada com *recursive entropy discretization* (Discretização supervisada)

Atributo	Pontos de corte após ...					Num. de int.
	30 instânc.	60 instânc.	90 instânc.	120 instânc.	150 instânc.	
At1	5.45	5.55	5.85	5.45	5.55	2
At2	2.75	3.25	3.25	3.25	3.05	2
At3	2.60; 5.05	2.60; 5.05	2.60; 5.05	2.60; 4.80	2.60; 4.80	3
At4	0.75; 1.65	0.75; 1.65	0.75; 1.65	0.75; 1.75	0.75; 1.75	3

Figura: 6.19: Descrição da discretização alcançada com o PiD (supervisionado e não supervisionado) para o conjunto de dados Iris

Na tabela anterior podemos observar que os limites dos intervalos são flexíveis, podendo alterar-se a medida que surgem novos exemplos. Observa-se igualmente que o método mantém o número de intervalos

definido na primeira vez que se observa os dados.

6.7 Resultados experimentais

Segundo Kohavi, os métodos mais vulgarmente utilizados para avaliação de um algoritmo são: a validação cruzada e o bootstrap [Dks, 1995]. Ainda segundo Kohavi a validação cruzada é importante não só para prever o futuro desempenho do algoritmo mas também para a escolha do algoritmo [Dks, 1995], existindo vários estudos em que se utiliza esta validação para prever o futuro desempenho de algoritmos incrementais. Por essa razão, neste capítulo irá ser focada a avaliação do algoritmo em termos da taxa de erro. Kohavi aconselha ainda a utilização da validação cruzada com dez pastas para a avaliação do algoritmo. O TANi apresenta um desempenho muito próximo do TAN, que sendo um algoritmo incremental é deste modo um bom algoritmo para o teste ao PiD. Para os testes do TAN e TANi com o PiD seguiu-se a seguinte metodologia:

- primeira fase da discretização:

  Para a discretização inicial aplicaram-se os métodos mais simples: o Equal-with Binning e o Equal-frequency Binning. Para que o método não necessitasse da intervenção do utilizador, na escolha do número de intervalos, recorreu-se a seguinte fórmula:

  $$t = \frac{N}{\sqrt{N}}$$

  Esta é uma fórmula totalmente empírica, mas que demonstrou obter bons resultados.

- segunda fase da discretização:

  Na segunda fase da discretização testaram-se dois tipos de discretização, uma não supervisionada e uma supervisionada. Para os testes da discretização não supervisionada, foi seleccionado o método Equal-frequency Binning. Dada a simplicidade do método, esta segunda parte da discretização é facilmente obtida a partir da primeira fase da discretização. Assim o método, é reiniciado sempre que chegam novos dados. Para que o método não necessitasse

da intervenção do utilizador, para definir o número de intervalos criados, usa-se a seguinte formula:

$$t = \sqrt[4]{N}$$

N é o número de exemplos observados e t o número de intervalos.

O método supervisionado escolhido foi o ENT-MDL [Fi, 1993], onde o critério de paragem é o minimum description length. Para a actualização da segunda camada usa-se a heurística de manter a árvore de discretização encontrada inicialmente.

Nos diversos quadros são apresentadas as taxas de acerto para os algoritmos: - NB , o naive Bayes do WEKA. - NB Cont. , o naive Bayes que lida com atributos contínuos. - Bayes Net. , uma versão de rede Bayesiana do WEKA. - J48 , uma versão de árvore de classificação do WEKA. - TAN , o TAN de Friedman. - TANi , o algoritmo incremental.

Em todos os conjuntos de dados foi usada a validação cruzada com dez pastas. O quadro mostra a taxa de erro e o desvio padrão apresentados para cada um dos testes.

					numero de exemplos por pasta		
Nome dos dados	Instancias	Atributos	Atributos continuos	Classes	10Pastas(10%)	20Pastas(5%)	50Pastas(2%)	100Pastas(1%)
Hepatitis	80	19	6	2	8			
Iris	150	4	4	3	15			
Wine	178	13	13	3	17			
CRX	194	33	33	2	19			
Glass	214	9	9	6	21			
Heart	270	13	6	2	27			
Cleve	296	13	4	2	29			
Liver-disorder	345	6	6	2	29			
Ionosphere	351	35	34	2	34			
Cars	393	8	7	3	39			
Breast-Loss	569	32	31	2	56			
Australian	690	14	8	2	69	34	13	6
Pima	768	9	8	2	76	38	15	7
Diabetes	768	8	8	2	76	38	15	7
Tokyo	959	44	44	2	95	47	19	9
German	1000	20	13	2	100	50	20	10
Segmentation	2310	20	19	7	231	115	46	23
Waveform	5000	41	40	3	500	250	100	50
Churn	5000	20	16	2	500	250	100	50
Satelite Image	6435	36	36	2	643	321	128	64
Adult	45222	14	6	2	4522	2261	904	452
Shuttle	58000	9	9	7	5800	2900	1160	580

Figura: 6.20: Descrição dos dados usados no teste

Discretização	nenhuma	supervisada	supervisada	supervisada	não supervisada	supervisada	supervisada	não supervisada	supervisada	supervisada
	não increm.	não increm.	não increm.	não increm.	não increm.	não increm.	não increm.	Incremental	Incremental	Incremental
	NB(Cont.)	NB	Bayes Net	J48	TAN(EPW)	TAN(ENT)	TANi(ENT)	TAN(PID-EPW)	TAN(PID-ENT)	TANi(PID-ENT)
Conj. dados								pastas de 10%	pastas de 10%	pastas de 10%
Iris	95.33+/-5.49	95.33+/-5.49	93.33+/-5.44	94.67+/-6.89	92.67+/-6.63	92.00+/-6.13	93.33+/-4.44	78.00+/-7.73	83.33+/-11.0	84.00+/-11.0
Wine	97.22+/-1.40	97.22+/-1.40	97.78+/-1.88	91.89+/-7.61	91.05+/-8.29	92.75+/-7.87	94.41+/-7.41	89.35+/-7.25	90.98+/-6.01	92.68+/-5.22
CRX	77.35+/-4.87	77.04+/-4.48	86.07+/-3.67	85.15+/-2.45	83.62+/-4.37	82.69+/-2.99	83.00+/-2.63	83.17+/-3.83	82.08+/-4.41	82.09+/-3.93
Glass	48.57+/-5.12	68.18+/-7.46	69.11+/-7.93	65.87+/-10.84	64.44+/-9.44	64.37+/-9.31	64.37+/-9.58	64.39+/-7.82	55.58+/-8.69	52.84+/-10.5
Heart	84.07+/-8.56	84.07+/-8.38	83.70+/-7.45	74.44+/-7.03	82.22+/-6.49	78.89+/-6.54	77.04+/-7.77	79.26+/-7.45	80.00+/-9.27	79.63+/-9.11
Clevo	82.82+/-1.85	83.15+/-1.72	82.47+/-4.58	78.07+/-6.72	78.41+/-6.03	78.18+/-4.83	78.06+/-4.19	82.09+/-4.26	80.44+/-5.11	80.07+/-5.60
Liver-disorder	54.76+/-6.05	55.64+/-5.34	57.97+/-5.00	67.24+/-7.30	66.10+/-4.63	67.19+/-5.75	70.09+/-5.34	67.81+/-8.23	67.25+/-4.82	66.06+/-5.90
Ionosphere	80.65+/-4.85	90.03+/-2.76	90.03+/-2.76	91.17+/-3.89	90.60+/-3.83	90.89+/-3.74	88.90+/-5.09	85.20+/-5.30	90.87+/-4.23	90.30+/-3.09
Cars	80.60+/-4.89	80.60+/-4.89	88.78+/-3.84	96.42+/-3.46	96.16+/-2.18	95.65+/-3.21	97.18+/-3.30	98.72+/-1.81	97.95+/-2.02	91.56+/-6.18
BreastLoss	96.13+/-2.43	96.18+/-2.52	97.37+/-2.17	95.90+/-2.18	95.90+/-2.05	96.61+/-2.41	96.49+/-2.10	88.88+/-4.06	97.22+/-1.76	97.51+/-2.30
Australian	77.54+/-2.92	77.54+/-3.00	85.80+/-2.80	84.64+/-4.05	86.96+/-2.56	87.10+/-4.12	86.23+/-4.11	85.65+/-4.40	86.52+/-3.62	85.80+/-4.47
Pima	75.39+/-4.17	74.87+/-4.24	75.65+/-3.72	73.69+/-5.25	75.78+/-1.78	76.44+/-4.25	75.26+/-4.09	72.26+/-4.71	74.59+/-5.79	74.98+/-6.38
Diabetes	75.52+/-3.20	75.78+/-2.84	74.87+/-3.21	75.13+/-4.75	76.05+/-4.33	75.01+/-2.65	75.13+/-3.23	72.00+/-3.48	73.70+/-4.81	73.83+/-3.86
Tokyo	90.30+/-2.15	90.40+/-2.60	90.40+/-2.60	91.66+/-2.77	91.35+/-2.25	91.55+/-2.41	92.28+/-1.50	91.13+/-2.62	91.45+/-1.84	91.34+/-2.16
German	74.10+/-4.86	74.20+/-4.78	73.60+/-3.78	70.90+/-3.67	72.40+/-4.01	70.70+/-3.59	71.50+/-3.81	73.10+/-5.26	72.27+/-5.56	71.88+/-4.98
Segmentation	80.74+/-1.54	91.26+/-1.32	91.69+/-1.20	96.97+/-1.08	93.68+/-2.35	94.98+/-1.45	94.68+/-1.37	90.35+/-2.56	94.20+/-1.86	93.90+/-1.56
Waveform	79.90+/-1.02	79.88+/-1.06	80.12+/-1.51	75.74+/-1.37	80.28+/-1.54	80.68+/-1.41	80.80+/-2.02	81.30+/-1.15	80.58+/-1.56	80.04+/-1.34
Churn	88.60+/-1.85	88.62+/-1.78	87.44+/-1.56	94.38+/-0.61	84.78+/-1.44	89.02+/-1.92	87.04+/-1.44	84.04+/-1.45	87.54+/-1.44	86.84+/-1.73
Sat. Image	79.56+/-1.23	82.02+/-1.20	81.99+/-1.30	86.57+/-1.13	86.99+/-1.28	87.65+/-1.30	87.89+/-1.24	86.11+/-1.35	86.08+/-1.46	85.86+/-1.67
Adult	82.70+/-0.60	82.59+/-0.65	85.57+/-0.50	83.51+/-0.62	83.37+/-0.56	85.66+/-0.54	83.37+/-0.56	84.84+/-0.47	84.72+/-0.42	
Shuttle	93.01+/-0.58	99.44+/-0.14	99.44+/-0.14	99.97+/-0.03	99.68+/-0.06	99.93+/-0.04	99.86+/-0.08	98.79+/-0.06	99.80+/-0.07	99.78+/-0.08
% de acerto	80.72	83.05	84.35	84.67	84.41	84.53	84.68	82.62	83.68	83.13
rank	10	8	5	2	4	3	1	9	6	7
média do rank	6.62	5.71	4.9	4.71	5	4.57	4.48	6.71	5.33	6.24

Figura: 6.21: Resultados experimentais do TAN e TANi com o PiD, na sua versão supervisionada e não supervisionada comparada com alguns algoritmos do WEKA.

Discretização	nenhuma	supervisado	supervisado	supervisado	supervisado	supervisado	supervisado	supervisado	supervisado	supervisado
	não Incremental	não Incremental	não Incremental	não Incremental	não Incremental	inicial	Incremental	Incremental	Incremental	Incremental
Conj. dados	NB(Cont.)	NB	J48	Bayes Net	TANi	TANi	TANi	TANi	TANi	TANi
						pastas de 10%	pastas de 10%	pastas de 5%	pastas de 2%	pastas de 1%
Australian	77.54+/-2.92	77.54+/-3.00	84.64+/-4.05	85.80+/-2.80	86.23+/-4.11	85.94+/-3.81	85.80+/-4.47	85.65+/-2.69	85.36+/-4.85	82.17+/-6.14
Pima	75.39+/-4.17	74.87+/-4.24	73.69+/-5.25	75.65+/-3.72	75.26+/-4.09	71.99+/-4.17	74.98+/-6.38	72.52+/-3.45	74.08+/-4.56	73.18+/-5.84
Diabetes	75.52+/-1.30	75.78+/-2.84	75.13+/-4.75	74.87+/-1.21	75.13+/-1.23	73.05+/-5.05	73.83+/-1.86	73.7+/-4.58	73.44+/-4.87	73.57+/-3.57
Tokyo	90.50+/-2.15	90.40+/-2.60	91.66+/-2.77	90.40+/-2.60	92.28+/-1.50	90.09+/-2.62	91.34+/-2.16	92.39+/-2.31	88.11+/-4.09	89.05+/-3.87
German	74.10+/-4.86	74.20+/-4.78	70.9+/-3.67	73.60+/-3.78	71.5+/-3.81	72.20+/-4.14	71.88+/-4.98	72.10+/-3.75	71.4+/-3.6	72.90+/-4.86
Segmentation	80.74+/-1.54	91.26+/-1.32	96.97+/-1.08	91.69+/-1.20	94.68+/-1.37	77.49+/-1.80	93.9+/-1.56	92.38+/-2.69	91.39+/-2.01	91.73+/-1.36
Waveform	79.90+/-1.02	79.88+/-1.06	75.74+/-1.37	80.12+/-1.51	80.8+/-2.02	79.68+/-1.91	80.04+/-1.34	77.92+/-1.99	77.4+/-1.51	77.42+/-1.42
Churn	88.60+/-1.85	88.62+/-1.78	94.38+/-0.61	87.44+/-1.56	87.04+/-1.44	86.60+/-1.35	86.84+/-1.73	86.02+/-1.21	86.36+/-1.49	86.28+/-1.23
Sat. Image	79.56+/-1.23	82.02+/-1.2	86.57+/-1.13	81.99+/-1.30	87.89+/-1.24	86.73+/-1.60	85.86+/-1.67	86.22+/-0.96	85.77+/-1.43	84.04+/-1.53
Adult	82.70+/-0.60	82.59+/-0.65	85.57+/-0.5	83.78+/-0.48	85.66+/-0.54	84.74+/-0.43	84.72+/-0.42	84.29+/-0.49	84.31+/-0.52	84.23+/-0.57
Shuttle	93.01+/-0.58	99.44+/-0.14	99.97+/-0.06	99.44+/-0.14	99.86+/-0.07	99.80+/-0.05	99.78+/-0.08	99.77+/-0.07	99.69+/-0.07	99.61+/-0.12
% acerto	81.58	83.33	85.02	84.07	85.05	82.58	84.45	83.91	83.39	83.11
rank	10	7	2	4	1	9	3	5	6	8
média do rank	6.69	5.64	4.36	4.91	2.64	6	4.45	5.82	7.36	7.27

Figura: 6.22: Resultados experimentais para o PiD com diversos tamanhos de pastas comparada com a sua discretização inicial e com alguns algoritmos do WEKA.

De notar que todos os algoritmos correram as mesmas pastas mutuamente estratificadas e em particular os conjuntos de teste foram os mesmos para todos os passos da validação cruzada.

Deixa-se apenas a nota de que foram testados como método de discretização inicial o Equal Frequency Binning e o Equal With Binning. Por questões de espaço e objectividade, são apenas apresentados os resultados para os que foram considerados mais representativos. Gostaria no entanto de salientar que os resultados com o Equal With Binning e o Equal Frequency Binning apresentam desempenhos muito semelhantes.

No fim de cada um dos quadros encontra-se a média do desempenho para todos os conjuntos de dados, na linha seguinte encontra-se a posição do resultado anterior. Na linha seguinte pode-se visualizar a média das posições alcançadas em cada um dos conjuntos de dados.

Através da análise do quadro 6.21 verifica-se que:

* Os resultados dos algoritmos incrementais se aproximam dos algoritmos não incrementais;

* Uma análise mais exaustiva permite concluir que os resultados da discretização incremental obtêm um melhor desempenho nos conjuntos de dados com mais exemplos;

* Na discretização supervisionada o TANi com discretização incremental obtém uma taxa de acerto mais alta que o TAN com discretização não incremental em cinco conjuntos de dados;

* A discretização supervisionada indica obter melhores resultados que a discretização não supervisionada incremental;

* O pior resultado é obtido pelo naive Bayes sem discretização;

* A discretização não supervisionada incremental obtém um média da taxa de acerto mais baixa que o naive Bayes Com discretização supervisionada. De salientar no entanto que o naive bayes consegue essa vantagens nos conjuntos de dados com menos exemplos.

Nos resultados do quadro 6.21 verifica-se que a discretização incremental obtém melhores resultados com conjuntos de dados com maior número de exemplos. Separa-se então os onze conjuntos de dados com mais exemplos para relizar uma análise mais exaustiva dos resultados.

Analisando o quadro 6.22 observa-se que:

* A melhor média é obtida pelo TANi com discretização não incremental;

* O TAN com a discretização inicial, obtida com a primeira pasta dos dados, obtém a penúltima melhor taxa de acerto;

* Todas as discretizações incrementais obtêm melhor média da taxa de acerto que a discretização inicial;

* A discretização incremental obtém taxas de acerto próximas da discretização não incremental;

* A diferença entre a melhor média de acerto obtida e a discretização incremental, com pastas de 1%, é de 1.94, o que é um bom indicador. Ao analisar com mais detalhe podemos verificar que, por exemplo, no caso do conjunto de dados Australian a discretização inicial foi efectuada com cerca de 7 instâncias e foi reformulada 99 vezes. Tendo em consideração este facto, a troca no desempenho entre taxa de acerto e condições tão duras de incrementalidade foi satisfatória.

Um dos objectivos dos quadros 6.21 e 6.22 é verificar se a discretização tem um papel preponderante no desempenho do algoritmo e pode-se verificar que tal acontece, pois o mesmo algoritmo com o mesmo método de discretização, mas obtido quer inicialmente, quer incrementalmente torna-se decisivo no desempenho do algoritmo.

Analisando os dois quadros verifica-se que a discretização incremental melhora os seus resultados para conjuntos de dados com mais exemplos. No quadro 6.22 a diferença da média da taxa de acerto entre o TANi-PiD e o TANi com discretização não incremental, diminuiu. Deste modo pode-se concluir que o PiD obtém melhor desempenho em conjuntos com maior número de exemplos, podendo por essa razão indicar o PiD como um bom método para áreas como o Datamining.

Verifica-se ainda que a discretização incremental indica obter um desempenho melhor que o naive Bayes com discretização não incremental.

Capítulo 7

Conclusões e trabalho futuro

7.1 Conclusões

Esta tese tinha como objectivo o estudo de uma rede Bayesiana (TAN) incremental. Durante o decorrer desta verificou-se a lacuna na área de uma discretização incremental para a avaliação de um algoritmo incremental. Assim procurou-se dar como contribuição para a área não só um classificador Bayesiano incremental mas também um modo de avaliação correcto do classificador.

Os Sistemas de Recuperação de Informação têm como objectivo a realização das tarefas de indexação, busca e classificação de documentos (expressos na forma textual), a fim de satisfazer a necessidade de informação do indivíduo, geralmente expressa através de consultas. A necessidade de informação pode ser entendida como a busca de respostas para determinadas questões a serem resolvidas, a recuperação de documentos que tratam de determinado assunto ou ainda o relacionamento entre assuntos.

Hoje em dia, a localização de documentos através de engenhos de busca, geralmente, é feita com a utilização de buscas por palavras-chave ou expressões contidas nos documentos. O sucesso em encontrar documentos relevantes depende do casamento dos termos fornecidos pelo utilizador numa consulta, com os utilizados como índices na indexação da base de dados de documentos.

Com o crescimento das colecções de documentos digitais, os siste-

mas de recuperação de informação que localizam documentos utilizando buscas por palavras-chave e expressões simples têm-se tornado cada vez menos eficientes. Este insucesso está relacionado com os seguintes motivos: a dificuldade do utilizador em expressar o que realmente procura através de uma consulta; a forma desorganizada como os documentos resultantes da busca são mostrados; o número excessivo de documentos devolvidos.

Com a vasta quantidade e variedade de documentos disponíveis, formular uma consulta efectiva para uma busca é uma tarefa difícil, e examinar uma lista resultante de uma pesquisa onde os itens são muitos e estão ordenados de forma claramente não significativa pode ser tediosa. Assim, tornam-se necessários métodos que sejam capazes de realizar uma organização automática dos documentos em conjuntos, evidenciando o relacionamento entre os conteúdos desses documentos, e as relações de proximidade entre os conjuntos de documentos de forma visual.

Existem, na Web, classes de páginas com conteúdo e estrutura similar (por exemplo, páginas de chamadas de trabalhos, referências bibliográficas, etc); algumas delas têm sido tratadas por agentes extractores. Porém, estes sistemas negligenciam o facto de que algumas destas classes se relacionam entre si, formando grupos (por exemplo, o meio científico).

Torna-se necessária uma organização ou arquitectura de sistemas multiagentes cognitivos para a recuperação, classificação e extracção integradas de informação, a partir destes grupos. Para a realização destas tarefas, uma visão da Web que incorpora estas classes (visão por conteúdo) e também a funcionalidade de apresentação das informações mantidas nas páginas torna-se necessária. Cada agente processa uma classe, empregando ontologias do domínio e ontologias estratégicas para reconhecer páginas e extrair delas as possíveis informações úteis, comunicando entre si e cooperando com os outros agentes.

As indicações sobre páginas e links, trocadas entre os agentes, normalmente contêm menos lixo do que os resultados das consultas dos mecanismos de buscas tradicionais (por exemplo, Google, AltaVista e Excite). A arquitectura do agente apresenta várias formas de reutilização: código, esquema da base de dados, conhecimento e serviços dos mecanismos de busca. Resultados promissores da recuperação e classificação funcional e de conteúdo foram obtidos para agentes que processam eventos e artigos científicos, empregando uma ontologia do domínio cien-

tífico criada especificamente para este fim, sugerindo que a arquitectura é realizável.

A problemática referente à manipulação de informação em grandes redes como a Internet colocou questões de difícil solução para tornar fácil o acesso à grande fonte de informação disponibilizada pelos utilizadores. Áreas como recuperação de informação, agentes inteligentes, ontologias, classificação e extracção, e modelagens da Web subitamente incluíram-se entre os temas de maior pesquisa no campo da informática. A pesquisa nestas áreas tem, continuamente, tentado fornecer soluções adequadas, mas ainda se encontram em fase de maturação, e longe de soluções gerais, de alta performance.

A Web tem participado efectivamente nas actividades rotineiras de milhões de utilizadores dos sistemas computacionais e sua aplicação aos locais de trabalho tem sido indispensável em muitos casos, principalmente nos meios educacionais. Estas actividades podem ser prejudicadas quando se desperdiçam horas em buscas ineficientes, portanto análises devem ser empreendidas para se compreender melhor os processos de pesquisa implementados pelos utilizadores.

O entendimento dos processos de busca são primordiais para a melhoria da efectividade das pesquisas, pois o tempo consumido com estas actividades chega a 70% do total de acesso à Internet. Uma forma de melhorar o entendimento sobre o processo de busca é estudar o comportamento do pesquisador, analisando as habilidades e condições necessárias para uma busca de sucesso.

As formas de busca indicam estratégias empregues pelos utilizadores, sendo definidas aqui como um plano contemplando uma série de acções visando encontrar uma informação. Como exemplo, uma simples estratégia de busca seria a utilização de um site de buscas (Google, Yahoo!, ...) onde se digita determinado termo e se recebe uma listagem das páginas referenciadas e que contém algum relacionamento com aquele termo. Continua a seguir-se para alguma página recebida e assim por diante, até encontrar o que procura ou desistir.

Quanto à tendência da globalização do mercado da produção intelectual, pode-se argumentar que, dentro do quadro de mudanças estruturais porque vem passando o mundo, a disseminação de padrões culturais globalizados vem assumindo proporções sem limite. Tal situação tem-se acentuado principalmente porque o modo de produção industrial capita-

lista tornou-se vantajoso na produção e distribuição de produtos intelectuais, e através de seus mecanismos de distribuição - os média em geral - interfere poderosamente nos processos económicos, políticos e culturais das sociedades. Enquanto processo de desenvolvimento de complexas interligações entre sociedades, culturas, instituições e indivíduos, a globalização estimula e favorece a remoção dos nossos relacionamentos e de nossas referências de vida de contextos locais para contextos transnacionais.

A convergência tecnológica tem vindo a eliminar os limites entre os meios, tornando-os solidários em termos operacionais, e desgastando as tradicionais relações que mantinham entre si e com seus utilizadores. Na verdade, com a tecnologia digital torna-se possível o uso de uma linguagem comum: um filme, uma chamada telefónica, uma carta, um artigo de revista, qualquer deles pode ser transformado em dígitos e distribuído por fios telefónicos, microondas, satélites ou ainda por via de um meio físico de gravação, como uma fita magnética ou um disco. Além disso, com a digitalização o conteúdo torna-se totalmente plástico, isto é, qualquer mensagem, som, ou imagem pode ser editada, mudando de qualquer coisa para qualquer coisa.

A convergência tecnológica parece tender a cancelar a validade de fronteiras entre diferentes tipos de produtos intelectuais, serviços informativos e culturais, e a suprimir as linhas divisórias entre comunicação privada e de massa, entre meios baseados em som e em vídeo, entre texto e vídeo, entre as imagens baseadas em emulsão e as electrónicas, e mesmo a fronteira entre o livro e a tela. Uma das maiores consequências disso é a observável tendência de integração de diversos aspectos das políticas públicas para informática, electrónica e telecomunicações, com alguns aspectos das políticas relativas aos média e à cultura. A Internet, a imprensa, a indústria gráfica, o rádio, a televisão, as bibliotecas, os livros e as revistas científicas, as telecomunicações e a informática estão a ficar mais interconectadas e interdependentes, de tal forma que uma política de governo para uma delas pode ter significativas implicações para as outras.

A sociedade actualmente pode ser considerada, de modo geral, sociedade da informação, devido ao seu grande envolvimento com o meio da informatização. Independentemente dos caminhos que adoptemos, caberia levar em consideração os seguintes conceitos na abordagem do tema:

a) a imprevisibilidade da inteligência humana irá dar continuidade a estas estruturas. O vertiginoso desenvolvimento das tecnologias de informação e comunicações tem sido um poderoso instrumento para a rotina, reorganização e automatização do trabalho intelectual. O fenómeno tecnológico tem operado como libertador de energia cognitiva, que será necessariamente aplicada na área de conhecimento de cada ser humano, não importa seu nível de educação. E dado que além de libertar energia o fenómeno tecnológico também disponibiliza um fantástico arsenal de ferramentas de concepção e desenvolvimento de produtos e processos, torna-se impossível prever os conteúdos em si mesmos e, mais que isto, as formas que tais conteúdos tomarão, e a maneira como os elementos estruturais se organizarão e se relacionarão entre si e com os utilizadores.

b) a incontrolabilidade dos conteúdos que circulam, sob várias formas, através dos serviços de informações e comunicações. É da própria natureza dos elementos estruturais, sobretudo pelo avanço extraordinário da convergência tecnológica entre informática, comunicações e electrónica, a incontrolabilidade da produção e circulação de conhecimento.

O desenvolvimento tecnológico equilibra a equação social inventando dispositivos de relativo controlo de consumo, pelo menos enquanto se necessita de máquinas lógicas para aceder ao conhecimento circulante. Mais importante que isto, contudo, é o facto incontestável da incontrolabilidade da produção e circulação do conhecimento ser parte constitutiva, estruturante mesmo, da cultura contemporânea. Ela, através das tecnologias de informação e comunicações, realiza e radicaliza o sonho humano libertário (sem restrição às liberdades individuais).

c) a inevitabilidade de acção no sector, seja regulando - ou desregulando - a organização, a gestão e a produção, na intenção de garantir o atendimento do interesse público, a ordem democrática, os valores morais e éticos, a livre competição e a busca contínua da universalização do consumo dos serviços de informação e comunicações.

Um dos principais indicadores do desenvolvimento da sociedade da informação é a penetrabilidade das tecnologias de informação na vida diária das pessoas e no funcionamento e transformação da sociedade como um todo. Em âmbito geográfico, a penetrabilidade é medida principalmente pelo número de utilizadores da Internet numa determinada população.

Outro indicador fundamental da sociedade da informação, que com-

plementa a penetrabilidade das tecnologias de informação, constitui o nível de operação ubíqua, num determinado contexto, de recursos, produtos e serviços de informação na Internet por parte dos seus utilizadores, representando indivíduos, governos e as mais diferentes organizações sociais de carácter público ou privado. Esta operação ubíqua representa a consecução de inovações muitas vezes radicais no funcionamento da sociedade actual, especialmente nas actividades e processos que requerem o acesso à informação.

Os recursos, produtos e serviços de informação são identificados na Internet com o nome genérico de conteúdos. Em resumo, conteúdo é tudo o que é operado na Internet. Uma das contribuições mais extraordinárias da Internet é permitir que qualquer utilizador, em carácter individual ou institucional, possa vir a ser produtor, intermediário e utilizador de conteúdos. E o alcance dos conteúdos é universal, resguardadas as barreiras linguísticas e tecnológicas do processo de difusão. É através da operação de redes de conteúdos de forma generalizada que a sociedade actual vai mover-se para a sociedade da informação. A força motora para a formação e disseminação destas redes reside na eficiência das decisões colectivas e individuais.

Os conteúdos são, portanto o meio e o fim da gestão da informação, do conhecimento e da aprendizagem na sociedade da informação. Resumindo, a sociedade da informação desenvolve-se através da operação de conteúdos sobre a infra-estrutura de conectividade.

Pretende-se enfatizar a importância do processo de selecção de conteúdos e dos níveis de qualidade relativa - no sentido interpessoal - que deverá ser objecto de análise e controlo por parte dos sistemas intermediários de informação, mediante instrumentos adequados nas etapas de formação de stocks, processamento técnico e disseminação. Da acção normalizadora e do tratamento parametrizado dos conteúdos vai depender a sua melhor difusão e uso pela sociedade.

Podemos ir mais além e prever que será o volume de conteúdos operados por um país que determinará o seu desenvolvimento económico e social e a qualidade de vida dos seus habitantes. Num contexto globalizado, o volume de conteúdos operados por um país passa também a medir a sua capacidade de influenciar e de posicionar a sua população no futuro da sociedade humana.

7.2 Trabalho Futuro

Na área do desenvolvimento de algoritmos ou métodos incrementais existe ainda bastante a estudar, o trabalho de juntar um algoritmo incremental com um método de discretização incremental foi apenas um início. Assim, deixa-se aqui algumas das propostas:

1. avaliar o TANi como algoritmo online;

2. Um dos testes que se pretende efectuar ao TANi é programar um TAN, para que este gere exemplos. Esses exemplos têm como objectivo gerar um conjunto de dados para treinar o TANi. Com este teste pretende-se atingir dois objectivos: (a) verificar se o TANi adopta a mesma rede que o algoritmo que gerou os exemplos; (b) verificar a que velocidade o TANi adapta a estrutura correcta.

3. Igualmente importante seria o estudo do impacto da ordenação dos dados na construção do TANi, já que este é afectado pela maneira como os dados são ordenados;

4. Um bom desafio será ainda o desenvolvimento do K-means incremental para poder comparar o seu desempenho em termos de custo computacional, velocidade e taxa de acerto com o do PiD;

5. Outro desafio que poderá demonstrar a emergência do uso da discretização incremental é o teste desta em ambientes dinâmicos.

Capítulo 8

Bibliografia

[Akman & Surav, 1996] Akman, Varol & Surav, Mehmet. Steps toward Formalizing Context. AI Magazine 17(3): 55-72, 1996.

[Alvares & Sichman, 1997] Alvares, Luis Otávio; Sichman, Jaime S.. Introdução aos sistemas multiagentes. In: Jornada de atualização em Informática., Brasília, 1997.

[Apache, 2007] Apache - The Apache Software Foundation. Apache - Overview. Disponível em: <http://lucene.apache.org/java/docs/>.

[Appel, 2002] Andrew Appel. Modern Compiler Implementation in Java, Second Edition, 2002

[Araújo, 2006] Araújo, Anderson Viçoso de. Árvore de Decisão Fuzzy na mineração de imagens do sistema Footscanage. Curitiba, PR: 2006. Dissertação (Mestrado) - Programa de Pós- Graduação em Informática, Universidade Federal do Paraná, 2006.

[Atzeni, 1997] P. Atzeni, G. Mecca, P. Merialdo. Design and Maintenance of Data-Intensive Web Sites. Technical Report n. 25-1997

[Baeza-Yates & Ribeiro-Neto, 1999] R. Baeza-Yates, B. Ribeiro-Neto, 'Modern Information Retrieval'. New York: ACM Press Series/Addison Wesley, 1999.

[Bm, 1998] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[Borges, 1998] J.BORGES & M.LEVENE, 'Mining association rules

in hypertext databases'. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). New York City, New York, USA, 1998.

[Borges, 2006] BORGES, HELYANE BRONOSKI. Redução de Dimensionalidade de Atributos em Bases de Dados de Expressão Gênica. Curitiba, PR: 2006. 123 p. Dissertação (Mestrado) - Programa de Pós Graduação em Informática. Pontifícia Universidade Católica do Paraná, 2006.

[Braga, 2000] Braga, Antônio de Pádua; Laudermir, Teresa Bernarda; Carvalho, André Carlos Ponce de Leon Ferreira. Redes neurais artificiais: teoria e aplicações. Livros Técnicos e Científicos Editora S.A., 2000.

[Brin, 1998] S. Brin & L. Page, 'The anatomy of a large scale Web Search Engine'. In Seventh International World Wide Web Conference, Brisbane, Australia, 1998

[Brusilovsky, 1998] P.Brusilovsky.. Adaptative Educacional Systems on the World-Wide-Web: A Review of Available Technologies, In: Proceedings of Workshop 'WWW-Based Tutoring', 4th International Cofe-rence on Intelligent Tutoring Systems (ITS' 98), San Antonio, TX, agosto 1998.

[Catlett, 1991] J. Catlett. On changing continuous attributes into ordered discrete attributes. In European Working Session on Learning, pages 164-178, 1991.

[Cerquides, 2003] Jesus Cerquides. Tan classifiers based on decomposable distributions. Technical report, Institut d'Investigación en Inteligência Artificial, January 2003.

[Chakrabarti, 1999] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan, & A. Tomkins, 'Mining the link structure of the World Wide Web', 1999.

[Chakrabarti, 2000] S. Chakrabarti, 'Data mining for hypertext'. ACM SIGKDD Explorations, 2000.

[Chen, 1994] Hsinchun Chen. A textual database/knowledge-base coupling approach to creating computer-supported organizational memory. MIS Department, University of Arizona, 5 de Julho de 1994.

[Coe, 1995] Helder Coelho. Inteligência Artificial em 25 lições. Fun-

dação Calouste Gulbenkian, 1995.

[Cole, 2005] Cole, Bernard. Search Engines Tackle the Desktop. IEEE Computer, Los Alamitos, EUA, Vol. 38, p. 14-17, mar. 2005.

[Cooley, 1997] R. Cooley, B. Mobasher & J. Srivastava, 'Web mining: information and pattern Discovery on the World Wide Web'. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, 1997.

[Cooley, 1999] R. Cooley, B. Mobasher & J. Srivastava, 'Data preparation for mining world wide web browsing patterns'. Knowledge and Information Systems, 1999.

[Cormen, 2002] Cormen, Thomas H. et al. Algoritmos: teoria e prática. Rio de Janeiro, Brasil, Editora Campus, 2002. 916p.

[Cutting, 1992] D.D. Cutting, J. Karger, J. Pederson & J. Scatter. A cluster based approach to browsing large document collections. Proceedings of the Fifteenth International Conference on Research and Development in Information Retrieaval, 1992.

[Decker, 2000] S. Decker, S. Melnick F. V. Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, I.Horrocks, 'The Semantic Web:The Roles of XML and RDF'. IEEE Internet Computing, 2000.

[Dhs, 2001] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. John Wiley and Sons, New York, 2nd edition, 2001.

[Dks, 1995] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In International Conference on Machine Learning, pages 194-202, 1995.

[Etizione, 1996] O. Etizione, 'The World Wide Web Quagmire or gold mine' Communications of the ACM, vol.39, no.11, pp. 65-68, 1996.

[Fagiuoli & Zanalon, 2000] E. Fagiuoli and M. Zanalon. Tree-augmented naive credal classiffers. In Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, pages 1320-1327, 2000.

[Fi, 1993] U. M. Fayyad and K. B Irani. Multi-interval discretization of continuous valued attributes for classification learning. In 13th International Joint Conference on Artificial Intelligence, pages 1022-1027. Morgan Kaufmann, 1993.

[Frawley, 1992] W.J. Frawley, G. Piatetsky-Shapiro, AND C.J. Matheus. Knowledge discovery in databases: na overview. In G. Piatetsky-Shapiro & W.J. Frawley, editors, 'Knowledge Discovery in Databases'. AAAI / MIT Press, 1992.

[Freeman, 2004] Freeman, Eric et al. Use a cabeça - Padrões de Projecto. Rio de Janeiro, Brasil: Alta Books, 2004. 496p.

[Friedman & Goldszmidt, 1996] Nir Friedman and Moises Goldszmidt. Building classifiers using bayesian networks. In AAAI/IAAI, Vol. 2, pages 1277-1284, 1996.

[Friedman & Goldszmidt, 1997] Nir Friedman and Moises Goldszmidt. Sequential update of Bayesian network structure. In Proc. 13th Conf. on Uncertainty in Artificial Intelligence, pages 165-174, 1997.

[Freitag, 1998] D. Freitag, 'Information Extraction from HTML: Application of a General Machine Learning Approach'. American association for Artificial Intelligence, 1998.

[Gamma, 1994] Gamma, Erich et al. Design Patterns: Elements of Reusable Object-Oriented Software. First Edition. Upper Saddle River, EUA: Addison-Wesley Professional, 1994. 416p.

[Gantz, 2007] Gantz, John F. The Expanding Digital Universe, A Forecast of Worldwide Information Growth Through 2010. IDC White Paper, Framingham, EUA, mar. 2007.

[Ghani, 2000] R. Ghani, R. Jones, D. Mladenic, K. Nigam, & S. Slattery. Data mining on symbolic knowledge extracted from the Web. In Proceedings of the Sixth Internaional Conference on knowledge Discovery and Data mining (KDD-2000), 2000.

[Girardi, 1995] R. Girardi,. 'Classification and Retrieval of Software through their Descriptions in Natural Language', Ph.D. dissertation, No. 2782, University of Geneva, December 1995.

[Girardi, 1998] R. Girardi. 'Main Approaches to Software Classification and Retrieval'. Em: Ingeniería Del Software y reutilización: Aspectos Dinámicos y Generación Automática. Editores J. L. Barros y A. Domínguez. (Universidad de Vigo - Ourense, del 6 al 10 de julio de 1998). Julio, 1998.

[Goldschmidt, 2005] Goldschmidt, Ronaldo; Passos, Emmanuel. Data

Mining: Um Guia Prático: Conceitos, Técnicas, Ferramentas, Orientações e Aplicações. Rio de Janeiro, RJ: Elsevier, 2005.

[Gonchoroski, 2007] Gonchoroski, Sidnei Pereira. Utilização de Técnicas de KDD em um Call Center Ativo. Novo Hamburgo, RS: 2007. 119 p. Monografia (Bacharelado em Ciência da Computação) - Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, 2007.

[Google, 2007a] GOOGLE. Google Desktop Features. Disponível em: <http://desktop.google.com/en/features.html>.

[Google, 2007b] GOOGLE. Google Desktop SDK. Disponível em: <http://desktop.google.com/dev/>.

[Gospodnetic, 2005] Gospodnetic, Otis; Hatcher, Erik. Lucene in Action. Greenwich, Reino Unido: Manning Publications, 2005. 421p.

[Han, 2000] J. Han. OLAP Mining: An integration of OLAP with Data Mining. School of Computing Science, Simon Fraser University, British Columbia, Canada, 2000.

[Haykin, 2001] Haykin, Simon S. Redes neurais: princípios e prática. 2. ed. Porto Alegre, RS: Bookman, 2001. 900 p.

[Hec, 1997a] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft research, Advanced Technology Dision, February 1997.

[Hec, 1997b] David Heckerman. Bayesian networks for data mining. Data Mining and Knowledge Discovery, 1(1):79-119, 1997.

[Hkl, 2002] K. Huang, I. King, and M. Lyu. Constructing a large node chow-liu tree based on frequent itemsets. In Proceedings of the International Conference on Neural Information Processing - ICONIP2002., 2002.

[Jeronimo, 2001] Jeronimo, Paulo Marcelo. Estudo sobre: Data Mining : Data Warehouse : Cases - Data Warehouse. Novo Hamburgo, RS: 2001. 73 p. Monografia (Bacharelado em Ciência da Computação) - Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, 2001.

[Junior & Perez, 2006] Junior, Adelir José Schuler; Perez, Anderson Luiz Fernandes. Análise do perfil do utilizador de serviços de telefonia utilizando técnicas de mineração de dados. Revista Eletrônica de

Sistemas de Informação, Florianópolis, p. 1 - 8, 01 jun. 2006.

[Keogh & Pazzani, 1999] E. Keogh and M. Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches, 1999.

[Kleinberg, 1998] J.M. Kleinberg, 'Authoritative Sources in a Hyperlinked Enviroment'. In Proc. ACM-SIAM Symposyum on Discrete Alghoritms, 1998.

[Khoshgoftaar, 2001] K. Gouda AND M. Zaki: 'Efficiently Mining Maximal Frequent Itemsets'. In proceedings of the IEEE International Conference on Data Mining, San Jose, USA, pages 163-170, November 2001.

[Kosala, 2000] R. Kosala & H. Blockeel, 'Web mining research: a survey'. SIG KDD Explorations, vol.2, pp. 1-15, 2000.

[Kohavi, 1995] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI, pages 1137-1145, 1995.

[Kranz, 2004] Kranz, Paulo Henrique. Business Intelligence: Estudo Aplicado em Cooperativa Médica. Novo Hamburgo, RS: 2004. 103 p. Monografia (Bacharelado em Ciência da Computação) - Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, 2004.

[Ksd, 1996] Ron Kohavi, Dan Sommerfield, and James Dougherty. Data mining using MLC++: A machine learning library in C++. In Tools with Artificial Intelligence, page To Appear. IEEE Computer Society Press, 1996. http://www.sgi.com/tech/mlc.

[Kumar, 1999] S.R. Kumar, P. Raghavan, S. Rajagopalan & A. Tomkins, 'Trawling the web for emerging cybercommunities'. In Proceedings of the Eighth WWW Conference, 1999.

[Kumar, 2002] P-N. Tan and V. Kumar: 'Discovery of Web Robot Sessions Based on their Navigational Patterns'. Data Mining and Knowledge Discovery, 6, 9-35, 2002.

[Kushmerick, 1997] N. Kushmerick, 'Wrapper Induction for Information Extraction'. Doctoral thesis. University of Washington, Department of Computer Science and Engineering, 1997.

[Landwehr, 2003] Landwehr, Niels; Hall, Mark; Frank, Eibe. Logistic

model trees. Proceedings of the 14th European Conference on Machine Learning, p. 241-252, 2003.

[Lanley, 1995] P. Lanley. Learning in humans and machines: Towards an inter-disciplinary learning science, chapter Order Effects in Incremental Learning. Oxford, 1995.

[Lau, 2003] Lau, Lawrence J. Economic Growth in the Digital Era. Symposium on 'Welcoming the Challenge of the Digital Era' 2003 Kwoh-Ting Li Forum, Taipei, China, nov. 2003.

[Lee, 2001] Berners - Lee, Tim, Hendler, James, Lassila, Ora. The Semantic Web. Scientifc American, May 2001.

[Leong, 1996] C. Wang and T.-Y. Leong. Knowledge-based formulation of dynamic decision models. In Topics in Artificial Intelligence: Proceedings of the 5th PacificRim Conference on Artificial Intelligence, 1996

[Levy, 2000] A. Levy & D. Weld, 'Intelligent Internet Systems'. Artificial Intelligence, vol.118, no.1-2, 2000.

[Lima, 2003] Lima, Gercina Ângela Borém. Interfaces between information science and cognitive science. Ci. Inf., Brasília, Brasil, v. 32, n. 1, 2003 . Disponível em: <http://www.scielo.br/ scielo.php?script= sci_arttext&pid=S0100-19652003000100008&lng=en&nrm=iso>.

[Lingras, 2002] P. Lingras. Rough Set Clustering for Web Mining. Saint Mary´s University, 2002.

[Lls, 2000] Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 40(3):203-228, 2000.

[Martinhago, 2005] Martinhago, Sergio. Descoberta de Conhecimento sobre o processo seletivo da UFPR. Curitiba, PR: 2005. 114 p. Dissertação (Mestrado) - Departamento de Matemática, Universidade Federal do Paraná, 2005.

[Mesquita Mota, 2003] Mesquita Mota, M. Tecnologias de Gestão do Conhecimento e sua Relação com a Inovação nas Organizações: O Caso de uma Multinacional de Consultoria. Dissertação (Grau de Mestre em Administração de Empresas), Universidade Federal da Bahia, Salvador,

Bahia, Brasil, 2003.

[Meyer, 2000] Meyer, Bertrand. Object-Oriented Software Construction. Santa Barbara, EUA: Prentice Hall, 2000. 1296p.

[Microsoft, 2007a] MICROSOFT. Windows Search. Disponível em: <http:// www.microsoft.com/ windows/ products/winfamily/ desktopsearch/default.mspx>.

[Microsoft, 2007b] MICROSOFT. Development Plataform. Disponível em: <http:// msdn2.microsoft.com /en-us/ library/ bb331575.aspx>.

[Minsky, 1986] Minsky, Marvin. The Society of Mind. Simon and Schuster, New York, 1986.

[Mit, 1997] Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.

[Ml, 1998] A. Moore and M.S. Lee. Cached sufficient statistics for efficient machine learning with large datasets. Journal of Artificial Inteligence Research, 8:67-91, 1998.

[Mladenic, 1998] M. Mladenic & M. Globelnik, 'Efficient text categorization'. In Proceedings of Text Mining Workshop on the 10th European Conference on Machine Learning, 1998.

[Mobasher, 1997] B. Mobasher, N. Jain, E.H. Han & J. Srivastava, 'Web Mining: Patterns from WWW transactions'. Tech. Rep. TR96-050, Dept. of Computer Science, University of Minesota, 1997.

[Mst, 1994] Donald Michie, D. J. Spiegelhalter, and C. C. Taylor. Machine Learning, Neural and Statistical Classification. Ellis Horwood, 1994.

[Múg, 2002] Pedro María Larrañaga Múgia. Classificación supervisada via modelos gráficos probabilísticos. prova de habilitação para catedrático, November 2002.

[Nagappan, 2005] A Software Testing and Reliability Early Warning (STREW) Metric Suite, Nagappan, N., Ph.D. Dissertation, North Carolina State University, February 2005.

[Nayak, 2001] Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in Quantum Communication and the Complexity of Set Disjointness. In Proceedings of the Thirty-Third Annual ACM Symposium on the Theory of Computing, pages 124–133,

2001.

[Oliveira, 2001] Oliveira, Ivana Corrêa de. Aplicação de Data Mining na Busca de um Modelo de Prevenção da Mortalidade Infantil. Florianópolis, SC: 2001. Dissertação (Mestrado) - Engenharia e Sistemas, Universidade Federal de Santa Catarina, 2001.

[Pal, 2000] Sankar K. Pal, Varum Talwar, Pabitra Mitra, 'Web Mining in Soft Computing Framework: Relevant, State of the Art and Future Directions', 2000.

[Palm, 2002] R. Baraglia and P. Palmerini: 'Suggest: A web usage mining system'. In Proceedings of the IEEE International Conference on Information Technology: Coding and Computing, 2002.

[Peng, 2007] Peng, Fuchun et al. Context sensitive stemming for web search. Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, p. 639-646. ACM, 2007.

[Qui, 1993] R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. San Mateo, CA, 1993.

[Rou, 2002] Josep Roure. An incremental algorithm for tree-shaped bayesian network learning. In F. Van Harmelen, editor, Proceedings of the fteenth European Conference of Articial Intelligence (ECAI 2002), page 350. IOS Press, July 2002.

[Rou, 2004] Josep Roure. Incremental Methods for Bayesian Network Structure Learning. PhD thesis, Universidad Politécnica de Cataluña, 2004.

[Russel & Norvig, 1995] Stewart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall. 1995.

[Saa, 1998] David Saad. On-line Learning In Neural Networks. Cambridge University Press, 1998.

[Salton, 1983] G. Salton, 'An Introduction to Modern Information Retrieval'. New York: McGraw-Hill, 1983.

[Santos, 2008] Santos, Daiana Pereira dos. Mineração em Notas Fiscais de entrada de uma empresa calçadista. Novo Hamburgo, RS: 2008. 93 p. Monografia (Bacharelado em Ciência da Computação) - Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, 2008.

[Sousa, 1998] Sousa, Mauro Sérgio Ribeiro de. Mineração de Dados: Uma Implementação Fortemente Acoplada a um Sistema Gerenciador de Base de Dados Paralelo. Rio de Janeiro, RJ: 1998. 75 p. Dissertação (Mestrado) - Programa de Pós Graduação de Engenharia. Universidade Federal do Rio de Janeiro, 1998.

[Souza, 2006] Souza, Renato Rocha. Sistemas de Recuperação de Informações e Mecanismos de Busca na web: panorama actual e tendências. Perspect. ciênc. inf., Belo Horizonte, Brasil, v.11, n.2, p.161 -173, ago. 2006.

[Steele, 2001] Steele, R. Techniques for Specialized Search Engines, In Proceedings of Internet Computing '01, Las Vegas, June 25-28, 2001.

[Soderland, 1999] Stephen Soderland, Learning Information Extraction Rules for Semi-Structured and Free Text, Machine Learning, v.34 n.1-3, p.233-272, Feb. 1999.

[Sycara, 1996] K. Sycara, K. Decker, A. Pannu, M. Williamson & D. Zeng, 'Distributed Intelligent Agents'. The robotics institute, Carnegie Mellon University, 1996.

[Tan, 2005] Zhiyi Tan, Yong He, Leah Epstein: Optimal on-line algorithms for the uniform machine scheduling problem with ordinal data. Inf. Comput. 196(1): 57-70, 2005.

[Villain, 1999] Franck Petit, Vincent Villain: Time and Space Optimality of Distributed Depth-First Token Circulation Algorithms. WDAS 1999.

[Yw, 2003] Ying Yang and Geoff I. Webb. Weighted proportional k-interval discretization for naive-bayes classifiers. In 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Morgan Kaufman, 2003.

[Zaiane, 1998] O. Zaiane: 'Web Usage Mining for a Better Web-Based Learning Environment'. In Proceedings of Conference on Advanced Technology for Education, pages 60-64, Banff, Alberta, 1998.

[Zhong, 2002] N. Zhong, J. Liu, Y. Yao, 'In Search of the Wisdom Web'. IEEE Computer, vol.35, no.11, 2002, pp.27-31.

[Zobel, 2006] Zobel, Justin; Moffat, Alistair. Inverted Files for Text Search Engines. New York, EUA: ACM Pres, 2006.