PEDRO ALEXANDRE DA COSTA SOUSA

UM ENQUADRAMENTO
PARA A CATALOGAÇÃO AUTOMÁTICA DE DADOS
UMA ABORDAGEM MULTIAGENTES

Dissertação apresentada para a obtenção do Grau de Doutor em Engenharia Electrotécnica, especialidade de Sistemas de Informação Industriais, pela Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia.

Lisboa

2004

Autor:	**Pedro Alexandre da Costa Sousa**
Título:	**Um enquadramento para a catalogação automática de dados - Uma abordagem multiagentes**
Orientador:	**Prof. Doutor Adolfo Sanchez Steiger Garção**
Co-orientador:	**Prof. Doutor Fernando Moura Pires**
Instituição:	**Universidade Nova de Lisboa** **Faculdade de Ciências e Tecnologia** **Departamento de Engenharia Electrotécnica**
Morada:	**Quinta da Torre** **2829-516 Caparica** **Portugal**
Copyright:	**Universidade Nova de Lisboa**
Ano:	**2004**
Nº de Arquivo:	

Em Memória

do Fajé

À Fátima, à Filipa e à Beatriz.

Agradecimentos

Gostaria de começar por expressar o meu sentido agradecimento a todas as pessoas que me têm acompanhado nos últimos anos e que presenciaram com enorme paciência a maturação deste trabalho.

À Universidade Nova de Lisboa e à Faculdade de Ciências e Tecnologia, nas pessoas, respectivamente, do Magnifico Reitor Prof. Doutor Leopoldo Guimarães, e do Prof. António Nunes dos Santos, agradeço as condições proporcionadas essenciais à realização deste trabalho.

À Comissão Europeia, agradeço o suporte financeiro que permitiu a execução do projecto DEEPSIA que serviu de estudo de caso a todo o trabalho realizado, e aos parceiros de consórcio o empenhamento que conduziu à execução do projecto.

Ao PRODEP agradeço a bolsa que me foi atribuída e que foi essencial para permitir a minha dispensa de serviço docente.

Aos meus colegas do Conselho Directivo da Faculdade de Ciências e Tecnologia da UNL, que acompanharam de forma indirecta o desenvolvimento deste trabalho ao longo dos últimos cinco anos, nas nossas reuniões matinais de quarta-feira, agradeço o estímulo e o voto de confiança constante, assim como a frontalidade das nossas discussões.

Ao UNINOVA – Instituto de Desenvolvimento de Novas Tecnologias, na pessoa do seu Presidente, Prof. Doutor Adolfo Steiger Sanchez Garção, agradeço o excelente ambiente de investigação essencial para a minha maturação pessoal e cientifica.

Aos meus colegas e amigos de profissão, a maioria já há mais de dez anos, em especial à Rita Ribeiro, à Rita Barros, ao Manuel Barata, ao Ricardo Gonçalves e ao Luís Filipe Gaspar, o meu obrigado pela enorme cumplicidade geradora de um excelente ambiente de trabalho.

Ao Prof. Adolfo Steiger Sanchez Garção agradeço a orientação, a confiança e a liberdade de acção (valores inestimáveis), e as nossas conversas de fim-de-tarde.

Ás diversas equipas que estiveram envolvidas na concretização deste desafio o meu profundo agradecimento; sem a sua colaboração os resultados obtidos não teriam sido possíveis. Em especial, um agradecimento muito sentido ao Bruno Rene Duarte Santos, por estes excelentes anos de trabalho conjunto, e ao Hugo Morganho, por todo o esforço, muitas vezes penoso, que desenvolveu, essencialmente, em horário pós-laboral.

Ao Hélder Silva, obrigado pelo apoio incondicional, que se revela pelas horas extra dedicadas à nossa aventura conjunta que já leva mais de dez anos.

Ao Pimentão, obrigado por tudo, pelo apoio pessoal, pelas imensas discussões que enriqueceram esta dissertação de forma determinante, e pelas imensas horas de substituição, que permitem a concretização de muitos projectos.

À Íris e ao Miguel, que assistiram de perto ao desenrolar deste projecto, o meu obrigado pelo suporte e pela presença constante, (essencial para muitos equilíbrios), o meu sincero desejo que concretizem todos os vossos sonhos.

À Sandra, minha irmã, ao Alexis, meu irmão e à minha querida Avó, agradeço a companhia e o apoio, constantes, no meu processo de crescimento pessoal, que termina inevitavelmente reflectido no que concretizo.

Para os meus pais, não tenho uma vez mais palavras que descrevam a minha admiração pessoal, e o meu sentido de agradecimento pelo apoio constante, pelo seu amor, por serem o meu referencial e por todo o trabalho de auxílio directo e indirecto nesta minha tarefa.

À Fátima e às minhas lindas filhas, Filipa e Beatriz, os meus últimos agradecimentos, sem o vosso apoio constante, e imenso carinho sempre que chego a casa, nunca teria sido possível e nada faria sentido.

Sumário

Nesta dissertação faz-se a apresentação dos trabalhos elaborados conducentes à realização de provas na Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia no ramo de Engenharia Electrotécnica, na especialidade de Sistemas de Informação Industriais, para obtenção do grau de Doutor.

A tese defendida consiste na proposta de um enquadramento global de suporte ao processo de recolha e catalogação dos dados disponibilizados na Web por forma a permitir uma maior eficácia e melhor desempenho na sua exploração.

O enquadramento global assenta nos seguintes pilares: *i)* uma metodologia geral; *ii)* uma arquitectura de referência; *iii)* uma metodologia específica de suporte à derivação de sistemas particulares e; *iv)* a operacionalização da arquitectura de referência.

A metodologia geral está centrada no utilizador tendo por objectivo simplificar a recolha e catalogação dos dados electrónicos e viabilizando a personalização da Web pela construção de catálogos dinâmicos.

A arquitectura de referência recorre à utilização de catálogos dinâmicos, sistemas de multiagentes inteligentes, ontologias e métodos de aprendizagem em texto, por contraste com os métodos habitualmente utilizados nos portais de recolha de dados.

A metodologia específica de suporte à derivação de sistemas particulares possibilita uma aproximação sistemática à instalação da arquitectura, propondo um conjunto de passos que permitem capturar e configurar as necessidades do utilizador.

Finalmente, a **operacionalização da arquitectura de referência** origina a construção de um protótipo composto por dois sistemas-base: o Sistema de Catalogação e o Sistema Interactivo de Apoio à Derivação de Sistemas Particulares.

O **Sistema de Catalogação** é o sistema que permite o armazenamento e a consulta dos dados recolhidos através das pesquisas previamente efectuadas. O **Sistema de Apoio à Derivação de Sistemas Particulares**, permite a personalização do Sistema de Catalogação, pela definição de regras e SAD específicos, dedicados a cada caso concreto.

Sumariamente, os obstáculos mais relevantes, abordados no decurso dos trabalhos, foram:

- a coexistência de diversos formatos de dados na Web;
- a capacidade de processamento dos dados, desde a filtragem de documentos tendo por base a sua relevância, passando pela identificação dos conceitos e sua posterior classificação;
- a formalização do conhecimento com vista à adopção de uma terminologia comum;
- a natureza do problema distribuído, complexo, descentralizado e com reduzida estruturação.

Este documento está organizado em diversos capítulos e cada capítulo está dividido em várias secções. O primeiro capítulo apresenta a inovação e os objectivos genéricos do enquadramento global. O segundo capítulo descreve o estado da arte de um conjunto de assuntos essenciais para o desenrolar dos trabalhos. O terceiro capítulo apresenta, em detalhe, o enquadramento global e a arquitectura proposta. O quarto capítulo descreve a metodologia de derivação de sistemas particulares. O quinto capítulo apresenta o estudo de caso e os resultados obtidos que visam validar a tese defendida. Finalmente, o último capítulo apresenta as conclusões e trabalhos futuros.

Summary

This dissertation presents the work to be submitted at Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia in the field of Electrothecnical Engineering, speciality of Industrial Information Systems for achieving a Phd degree.

The thesis defended proposes a global framework to support the process of information retrieval and extraction of data available in the Web in order to enable its efficient exploration.

The global framework foundations are: *i)* a general methodology; *ii)* the reference architecture; *iii)* a methodology for the support of the instantiation of particular systems; and *iv)* the prototype (instantiation of the reference architecture).

The **general methodology** is focused on the user. Its objective is the effortless retrieval and extraction of information enabling the Web customisation based on dynamic catalogues.

The **reference architecture** is based on dynamic catalogues, multi-agent systems, ontologies, and text learning techniques in contrast with the traditional methods used for the portals construction.

The **methodology for the support of the instantiation of particular systems** facilitates a systemic approach to architecture instantiation, imposing a set of steps that capture and configure the user needs.

Finally, the **instantiation of the reference architecture** generates the prototype composed by two base sub-system: The catalogue system and the interactive system to support the particular system instantiation. The **catalogue system** enables the storage and browsing of the retrieved data stored in previous searches. The **interactive system to support the particular system instantiation** enables the catalogue system customisation, by the definition of rules and Decision Support Systems specific to each case study.

The most significant obstacles studied in this work were:

- the need to process different data formats in the Web;
- the capability to process the data, from filtering the documents based on its relevance to the user, to the concepts identification and classification;
- knowledge formalization in order to achieve a common terminology;
- the nature of the problem: its complexity, the need for decision decentralization and its ill-structure.

This document is organized in several chapters divided in several sections. The first chapter presents the innovation and the generic objectives of the global framework. The second chapter describes the state of the art of the most relevant subjects related to the work. The third chapter presents, in detail, the global framework and the reference architecture. In the fourth chapter the methodology for the support of the instantiation of particular systems is discussed. The fifth chapter presents the study case and the achieved results that validate the presented thesis. Finally, the last chapter presents the conclusions and future work.

Simbologia e notações

Geral

$=$	igual a
$\neq$	diferente de
$\approx$	aproximadamente
$<$	Menor do que
$\leq$	Menor ou igual a
$>$	Maior do que
$\geq$	Maior ou igual a
$\subset$	está contido em
$\subseteq$	está contido ou igual a
$\supset$	contém a
$\supseteq$	contém ou é igual a
$\in$	pertence a
$\notin$	não pertence a
$\cup$	reunião com
$\cap$	Intersecção com
$\Rightarrow$	implica que
$\Leftrightarrow$	Equivalente a
$\varnothing$	conjunto vazio

Σ	somatório de
Π	produtório de
C_K^N	combinação de *N, K* a *K*
-	diferença de conjuntos

Probabilidades, estatística e teoria de informação

X	Uma variável.
x_i	Um valor específico da variável X .
x	Um valor genérico da variável X .
$\hat{x}$	Um valor estimado para a variável X .
$\|X\|$	A cardinalidade da variável X .
n_T	Uma contagem de existências que satisfazem T .
$\Delta = \{A_1,...,A_{\|\Delta\|}\}$	Um conjunto de atributos, conjunto de valores admissíveis para a característica $A_i = \{a_{i1},...,a_{i\|Ai\|}\}$.
A_i	Uma característica.
C	Um conjunto de classes.
$A = \{A_1 \times ... \times A_{\|A\|}\} = A^k$	Um vector de características, produto cartesiano das características $A = \{(a_{11},...,a_{n1}),..., (a_{1\|a1\|},...,a_{n\|an\|})\}$.
$S \subset A$	Vector de características seleccionadas do espaço de características. S é uma projecção de A .
E	Um classificador.
p()	probabilidade de
E()	esperança de
I()	informação
H()	entropia de

Funções matemáticas

$$\delta(x,y) = \begin{cases} 0 & se\ x \neq y \\ 1 & se\ x = y \end{cases}$$ Operador de comparação

$$\underset{j}{arg\,max}\ (F_j(x))$$ designa o valor j do argumento que maximiza a expressão indicada.

$$C(X,Y) = X.Y = \sum_{i=1}^{K} x_i y_i$$ Medida de dissemelhança (correlação)

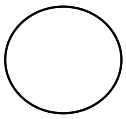
$$D_M(X,Y,\lambda) = \left(\sum_{i=1}^{K} |x_i - y_i|^{\lambda} \right)^{1/\lambda}$$ Distância de Mahalanobi

Notação gráfica

Módulo aplicacional

Base de dados, bloco de dados

Elemento de agregação



Agente

Fluxo de dados bidireccionais

Fluxo de dados unidireccional

Índice de matérias

Índice de figuras

Índice de tabelas

1 Introdução

Este capítulo expõe, resumidamente, o enquadramento, os problemas, as contribuições mais relevantes e a organização da dissertação.

A primeira secção apresenta a situação actual do ambiente Web em que se enquadram os trabalhos apresentados. A segunda secção apresenta os problemas identificados bem como as propostas efectuadas. A terceira secção apresenta as contribuições consideradas, na perspectiva do autor, como as mais relevantes. A quarta e última secção apresentam, respectivamente, a notação e a organização deste documento.

1.1 A situação

A mitológica capacidade de acesso ao saber universal por consulta ao Oráculo[1], surge ao longo de toda a existência do Homem. Estaremos nós, finalmente, perto de atingir esse objectivo com o recurso à utilização da Web? Será possível ter acesso a todo o conhecimento actual pela simples pesquisa na Web? Será a Web o primeiro Oráculo do novo milénio? Objectivamente, a promessa de informação está cada vez mais à distância de uma consulta.

Todavia, apesar de ser incontestável que a Web se transformou, em poucos anos, na maior e mais rica plataforma de informação, subsistem barreiras que nos afastam do sonho. Aceder de forma imediata e fiável à informação de que necessitamos, continua a ser uma tarefa árdua devido à disponibilização dos dados de forma não estruturada e não normalizada. Esta aproximação, apesar de ter sido fundamental para o sucesso obtido,

[1] Oráculo – Na antiga Grécia, o Oráculo era um local onde profecias divinas eram disponibilizadas aos mortais. Usualmente, as profecias eram respostas a perguntas, apesar de poderem fluir aleatoriamente através do intermediário, normalmente um sacerdote. O mais famoso Oráculo, «Apollo de Delphi», descoberto em forma de fissura em «Mt. Parnassus», liberta um gás que causava convulsões no gado. As convulsões e o comportamento selvagem dos animais afectados eram interpretados como inspiração divina.

dificulta, de forma determinante, a exploração do conhecimento, augurando que a procura de soluções mais eficazes estará sob investigação intensiva nos próximos anos.

Do estado inicial, em que estava confinada a uma pequena comunidade académica, a Internet tornou-se disponível à população, em geral, no início dos anos noventa com a generalização de computadores pessoais e das redes de dados. A Internet transformou-se, assim, numa plataforma acessível e compreensível, tornando possível a disseminação de enormes quantidades de informação. Vivemos, definitivamente, numa época marcada pela sua crescente «omnipresença» que alterou os processos de transmissão de conhecimento no espaço de um década, por contraste com uma lenta evolução durante séculos.

A aceleração da alteração dos processos de transmissão de conhecimento inicia-se no século passado, com a gradual introdução de novos meios de comunicação, (em especial a rádio e a televisão) que permitem assegurar a disseminação de informação de forma massiva. Todavia, com o surgimento da Web, as alterações impostas nos últimos dez anos ultrapassam as expectativas, sendo visível a sua influência nos restantes meios. Muitos livros são complementados por conteúdos dinâmicos na Internet, já para não referir os livros totalmente digitais. São cada vez mais raras as edições periódicas (revistas, jornais) que não são publicadas na Internet. A televisão e a rádio, para além de terem iniciado os primeiros passos na era digital interactiva, fazem já intensiva utilização da Internet na preparação e realização dos seus programas. Este cenário deve-se, principalmente, ao aumento vertiginoso de informação disponível, que durante o primeiro semestre de 2000 permitiu ultrapassar a mítica barreira de um milhar de milhão de páginas, a um ritmo de crescimento, à data, de três milhões de documentos por cada dia [1]. Porém, passados somente dois anos, em 2002, um estudo da Global Reach [2], indica que a Internet já era composta por um número total de 310 mil milhões de documentos, o que representa um aumento superior a 300 por cento.

O «Internet Systems Consortium» regista, desde 1981, uma métrica indirecta (o número de servidores Internet), que é utilizada para caracterizar o crescimento da Web [3]. Os registos remontam a 1981, com um valor total de 213 servidores, apresentando um crescimento exponencial que só sofre um abrandamento a partir de 2002. O último valor apresentado é já de 171 638 297 servidores em Janeiro de 2003.

Neste meio, a língua Inglesa tem assegurado uma esmagadora prevalência no número de documentos, observável na Figura 1 que resume os dados apresentados em [2]. É, todavia, muito significativo o número de documentos noutras línguas, que em valores absolutos ultrapassavam já, à data do estudo, os 100 mil milhões, representando um conhecimento que não deve, nem pode, ser ignorado.

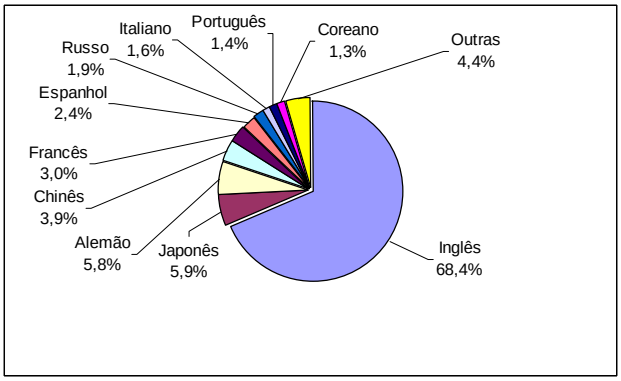


Figura 1 – Caracterização dos conteúdos na Web
por língua para um total de 313 mil milhões de documentos

Igualmente relevantes, são as estimativas de [4], quanto ao número de pessoas por comunidade linguística de utilizadores, onde a relevância do Inglês é inferior ao esperado, tendo em consideração os conteúdos disponíveis. Este estudo, (apresentado na Figura 2 e Tabela 1) foi organizado por comunidades virtuais, ignorando o local de residência, e abrangeu uma população de 3,515 mil milhões de pessoas, tendo revelado uma prevalência das línguas maternas, e um aumento da influência das línguas asiáticas.

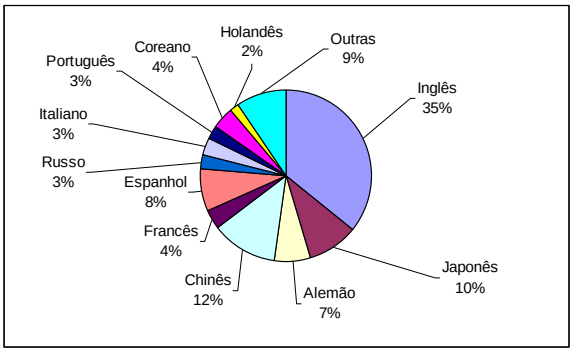


Figura 2 – População mundial *on-line* por comunidade linguística

A Tabela 1 corrobora a importância das comunidades de língua não inglesa, não só pela supremacia do número de utilizadores e do número de acessos mas, também, pelo seu enorme potencial de expansão, uma vez que a sua abrangência é ainda inferior a 10 por cento do total da comunidade, por contraste com a comunidade inglesa que já ultrapassa os 50 por cento. Nesta babilónia linguística em crescimento exponencial, o cenário inicial, em que o utilizador sabia da existência de um conjunto de endereços Internet onde residiam as informações necessárias foi, rapidamente, ultrapassado. Surgiram, então, os motores de pesquisa de informação em paralelo com os directórios de endereços, ideais para a realização da pesquisa de informação. Estes portais, que de uma forma generalista ou temática disponibilizam informação, passaram a ser os sítios Web mais visitados, assegurando a porta de entrada na Internet para uma grande parte de utilizadores.

	Língua inglesa	Línguas europeias excluindo o inglês	Línguas asiáticas	Total
Acesso à Internet	262,3	257,4	216,9	679,7
Estimativa para 2004	280	328	263	940
Total da população	508	1,218	1789	3515

Tabela 1 – Apresenta o número de pessoas por comunidade linguística, com valores apresentados em milhões de pessoas

Este cenário, de elevada quantidade e heterogeneidade linguística na informação disponível em formato electrónico na Web, lança novos desafios à comunidade de investigadores, devido não só à magnitude do problema mas, também, devido à sua natureza distribuída, à combinação arbitrária de formatos (texto, imagens, sons, vídeo, etc.), bem como pela constante alteração dos conteúdos.

Actualmente, estão em curso iniciativas com vista a possibilitar o processamento automático da informação existente na Internet. Esta nova fase permitirá estender a Internet, (ainda muito focada à utilização directa por humanos), à utilização por máquinas, o que auxiliará a sua exploração. O processo, que se iniciou com a inclusão de novas normas de dados, focadas na utilização de marcas nas páginas Internet, fará a sua evolução, até atingirmos a Web Semântica, altura em que, não só se disponibilizarão os dados, mas igualmente os meta-dados e as regras armazenadas nas bases de dados. A Internet será, então, uma «super base de dados» com capacidades de disponibilização verdadeiramente «assombrosas».

Em paralelo, surgem sistemas baseados, principalmente, na análise de conteúdos na vertente texto, que têm como objectivo auxiliar o utilizador. Estes desenvolvimentos, suportados em técnicas de Recuperação de Informação, Extracção de Informação, Aprendizagem Automática, Agentes Inteligentes permitem antever veículos privilegiados de mediação entre a Web e o utilizador.

1.2 O problema

Ironicamente, a melhor característica da Web é, ao mesmo tempo, o seu maior «calcanhar de Aquiles». A facilidade de utilização, não só pela forma simples e democrática de criar conteúdos, como também pela sua utilização intuitiva, disponibilizou um conjunto de informação de tal forma gigantesco que muitas vezes funciona como um obstáculo à pesquisa de dados relevantes para os utilizadores, devido à natureza caótica da organização e apresentação da informação. A riqueza disponível é incalculável, porém está oculta numa rede de elos e numa heterogeneidade que dificulta, ou impossibilita a sua descoberta. Esta situação conduz ao desnorte do utilizador devido à dimensão,

complexidade e ausência de estruturação, que dificulta a pesquisa de informações específicas, tendo passado a ser virtualmente impossível obter resultados por navegação nos elos disponíveis no hipertexto. Esta dificuldade, conduziu à criação de portais de pesquisa que disponibilizam mecanismos de auxílio ao utilizador (motores de pesquisa).

A utilização dos motores de pesquisa passou a ser a forma mais usual de procurar informações contornando, assim, a necessidade de navegar de forma «aleatória» na busca das informações necessárias. Esta abordagem, apesar de obrigar o utilizador a conhecer os servidores que disponibilizam mecanismos de pesquisa, e a conhecerem as suas particularidades, oferece vantagens de eficiência evidentes.

Todavia, mesmo através dos portais, a quantidade esmagadora de informação actualmente residente na Internet levanta o problema de encontrar, de forma fiável, a informação necessária. A pesquisa de documentos torna-se, com a disponibilidade de dados, numa verdadeira arte mágica na qual um misto de sensibilidade, sorte e saber dos utilizadores, passaram a ser determinantes para o sucesso das pesquisas efectuadas. Os obstáculos a ultrapassar são diversos. Em primeiro lugar, existe a necessidade de identificar correctamente o objecto de pesquisa, o que é agravado pelo facto de, na maioria dos casos, o utilizador só fazer uma vaga ideia do que procura. O segundo obstáculo, relaciona-se com a necessidade de recuperar a informação existente na gigantesca base de dados, que é actualmente a Web, com a agravante da informação, na maioria dos casos, ser de carácter não estruturado.

A disponibilização destes serviços obrigou, no caso dos motores de pesquisa, à criação de um conjunto de aplicações que naveguem e cataloguem as páginas encontradas e, no caso dos directórios, à criação de mecanismos de actualização, (usualmente realizados de forma manual) de toda a informação.

Para além do problema da descoberta diária de elevadas quantidades de novos documentos é, também, necessário validar os documentos já referenciados restringindo a uma cobertura limitada mesmo os grandes motores de pesquisa, tais como o Google[2], Northern Light[3], ou AltaVista[4]. Outra grande barreira, reside na dificuldade de identificar de forma inequívoca, os conteúdos presentes em cada documento. A identificação da palavra «portaria», tanto pode significar a presença de um documento legal, como a descrição de um edifício. Esta ambiguidade, forçada pela não estruturação dos dados, leva o utilizador a iniciar a sua pesquisa a partir de uma lista de endereços Internet que podem, ou não, ser relevantes.

[2] http://www.google.com

[3] http://www.northernlight.com

[4] http://www.altavista.com

Este facto conduz a que a pesquisa da informação na Internet seja uma tarefa cada vez mais exigente e muitas vezes improdutiva.

A solução deste problema seria trivial se os documentos contivessem informação semântica sobre os seus conteúdos. Neste caso, estando o sistema na posse do significado semântico, a recolha da informação seria simples. Todavia, a realidade está longe de se aproximar deste cenário, por diversas razões, entre outras, pela:

- esmagadora maioria da informação disponibilizada não contém a semântica dos dados, o que impossibilita a abordagem descrita;
- existência da informação semântica não assegura, de imediato, a capacidade de reconhecimento por parte do sistema, pois é necessário que exista uma normalização de terminologias, ou seja, que exista a partilha do significado associado à informação;
- inexistência de um conjunto de normas alargadas de representação de conhecimento em diversos domínios inviabiliza a curto prazo a implementação da Web semântica.

Neste cenário, novas metodologias e ferramentas são essenciais para abordar os problemas identificados, com vista à criação de soluções inovadoras que permitam aos utilizadores explorar, de forma intuitiva e eficaz, as potencialidades da informação armazenada na Internet.

1.3 As contribuições

A tese apresentada nesta dissertação consiste na proposta de criação de um enquadramento global que permite identificar e classificar dados disponíveis num subconjunto de sítios Internet, previamente seleccionados, e realizar o seu armazenamento, num catálogo dinâmico, que faculte consultas de forma intuitiva.

O enquadramento global proposto visa permitir a procura, identificação e processamento de documentos relevantes para o utilizador, assim como a extracção e armazenamento dos dados constantes nesses documentos num catálogo previamente personalizado, com vista a simplificar a pesquisa posterior.

Este objectivo encerra em si diversos problemas que serão atempadamente aprofundados no decorrer desta dissertação; todavia, com o propósito de conduzir e contextualizar o leitor são, de imediato, sumariamente apresentados.

O primeiro grande obstáculo a ultrapassar é a capacidade de interpretação dos documentos em análise, que surgem em diversos formatos e meios, pelo que foi necessário criar uma representação interna que permitisse a abstracção do formato em que os dados são armazenados. Desta forma, é possível realizar a abstracção do documento, e desenvolver

técnicas e metodologias de análise exclusivamente dependentes da representação adoptada.

O problema seguinte assenta na identificação de documentos relevantes para o utilizador, isto é, no desenvolvimento da capacidade de aprendizagem das suas necessidades, permitindo ao sistema o reconhecimento de documentos relevantes. As técnicas adoptadas recorrem à aprendizagem supervisionada, tendo sido testados diversos métodos convencionais, propostas alterações e criados novos métodos. Esta área foi desenvolvida seguindo a abordagem tradicional de: *i)* criação de base de dados de exemplos (o *corpus*), que se espera represente o universo em estudo; *ii)* pré-processamento dos dados; *iii)* utilização de algoritmos de indução de classificação e; finalmente *iv)* adopção de um sistema de apoio à decisão, baseado nos classificadores e na experiência obtida através da análise dos dados.

Ultrapassada a barreira da identificação dos documentos relevantes para o utilizador, segue-se a necessidade de identificar os assuntos apresentados nos documentos e qual a sua classificação dentro do catálogo previamente personalizado pelo utilizador. Esta tarefa, devido à variedade de assuntos e classificações existentes, foi realizada sem o recurso à aprendizagem automática, utilizando regras de inferência «se-então» que permitem descrever e extrair os conceitos presentes nos documentos seleccionados, com vista à sua posterior classificação.

Finalmente, a catalogação dos assuntos é realizada com o recurso a uma ontologia e à indexação por palavra-chave. A escolha da ontologia baseou-se na necessidade de criar uma sistema com uma interface intuitiva, acompanhando, ao mesmo tempo, a evolução para a Web semântica.

De uma forma lata, a metodologia geral defende uma abordagem invertida, em que o foco é transferido para o sistema do utilizador, por substituição do esforço de personalização dos sítios Internet.

A arquitectura de referência proposta baseia-se numa abordagem de sistemas de multiagentes, isolando cada um dos assuntos, sumariamente apresentados, num agente especializado (encarregue de efectuar o seu processamento) e posterior encaminhamento para o agente seguinte. Os multiagentes são utilizados como paradigma de análise e desenvolvimento, não existindo, ao longo da dissertação, contribuições para a melhoria evolutiva deste paradigma, para além da demonstração da sua utilidade.

A metodologia específica de suporte à derivação de sistemas particulares assenta na adaptação da metodologia e na arquitectura, aos requisitos concretos do utilizador, sendo realizada, essencialmente, pelo recurso à aprendizagem supervisionada.

Finalmente, a operacionalização da arquitectura de referência origina a construção de um protótipo composto por dois sistemas-base: **o Sistema de Catalogação** e o **Sistema Interactivo de Apoio à Derivação de Sistemas Particulares**.

O **Sistema de Catalogação** é o sistema que permite o armazenamento e a consulta dos dados recolhidos através das pesquisas previamente efectuadas. O **Sistema de Apoio à Derivação de Sistemas Particulares** permite a personalização do Sistema de Catalogação, pela definição de regras e SAD específicos, dedicados a cada caso concreto.

O protótipo foi implementado recorrendo, sempre que possível, à utilização de ferramentas e ambientes de desenvolvimento disponíveis em código livre com vista a permitir a sua fácil adopção.

Neste contexto, as contribuições de carácter geral mais significativas desta dissertação, são:

- a criação de um enquadramento global para a recolha e catalogação de informação baseado em catálogos dinâmicos *versus* a perspectiva tradicional de personalização de pesquisas suportadas em portais;
- a proposta de uma arquitectura de recolha de dados, baseada num sistema de multiagentes inteligentes, que assegure a capacidade de identificar documentos contendo assuntos relevantes para o utilizador, e que obtenha o reconhecimento de conceitos e sua catalogação;
- a aplicação da tese a um estudo de caso com Pequenas e Médias Empresas (PME) encaradas como entidades de aquisição de produtos e serviços.

As contribuições mais significativas, de carácter específico para a recuperação e extracção de informação, foram:

- a proposta de uma metodologia sugerida para assegurar a representatividade de uma base de dados de exemplos (*corpus*);
- a proposta para a representação de documentos adequada à aprendizagem em texto;
- a utilização da determinação da Informação Mútua Condicional (IMC) para a optimização da selecção de características;
- o algoritmo de indução de estimadores C4.5 iterado, baseado no C4.5, que permite optimizar a compactação das árvores induzidas;
- a criação de Sistemas de Apoio à Decisão (SAD) baseados em vários estimadores, tirando partido da diversidade de desempenho, dependendo da localização da observação;

- a solução apresentada para reconhecimento de conteúdos (representação e regras) e classificação de conteúdos (regras «se-então» e referência inversa de palavras-chave).

As metodologias e a arquitectura propostas, nesta dissertação, foram adoptadas e validadas no âmbito do projecto DEEPSIA «Dynamic on-linE IntErnet Purchasing System based on Intelligent Agents», submetido ao 5º Programa quadro IST e financiado pela Comissão Europeia. O projecto visava a criação automática de catálogos de produtos anunciados na Internet para PMEs. O autor foi o responsável técnico-científico do projecto, com supervisão conjunta do Prof. Doutor Adolfo Steiger Garção e do Prof. Fernando Moura Pires.

1.4 A notação e organização da dissertação

Ao longo de todo o texto e com vista a simplificar a leitura, foi adoptada uma notação única que pode ser consultada no capítulo Simbologia e notações. A simbologia geral não merece comentários. Todavia, o carácter mais específico da segunda secção é, em seguida, sumariamente apresentada.

Uma variável é representada por uma letra maiúscula, genericamente X , um valor genérico admissível é representado pela letra minúscula, genericamente x , sendo os valores específicos, que a variável admite, representados pela letra minúscula com um índice *i*, genericamente x_i . A cardinalidade dos valores admissíveis pela variável são representados pelo módulo da variável, genericamente $|X|$, e o seu valor estimado pela letra minúscula sob o acento circunflexo, $\hat{x}$. As restantes simbologias são específicas de determinados capítulos, pelo que serão introduzidas sempre que oportuno.

Foram, igualmente, adoptadas como definições: ***uma característica,*** por vezes referida como ***atributo*** ou ***variável***, que toma valores predefinidos de um conjunto dependente do problema em estudo; uma ***observação***, por vezes referida como ***exemplo***, sendo um conjunto de ***variáveis ordenadas***; a ***característica objectivo***, que é a característica do exemplo que descreve o fenómeno em estudo sobre o qual se pretendem realizar as previsões. A ***característica objectivo*** é, usualmente, referida como ***variável dependente*** e as restantes como ***independentes***. Um ***exemplo*** para o qual o valor da ***característica objectivo*** é conhecido é um ***exemplo classificado***. Um ***classificador,*** por vezes referido como ***modelo*** ou ***estimador***, descreve uma relação entre ***características*** e a ***característica objectivo***, sendo utilizado para realizar as estimativas para observações não classificadas.

Esta dissertação está organizada em capítulos e em anexos que se explicitam a seguir:

1) Este, primeiro capítulo, que apresenta, resumidamente, o enquadramento, os problemas actuais, as contribuições consideradas mais relevantes, na perspectiva do autor, e a notação e organização adoptadas neste documento;
2) O segundo capítulo, expõe o estado da arte dos assuntos mais relevantes abordados ao longo desta dissertação. Conceitos e problemas são discutidos com o objectivo de permitir ao leitor uma melhor interpretação do trabalho efectuado;
3) O terceiro capítulo foca, em detalhe, a metodologia geral e a arquitectura proposta, com um enfoque especial no sistema de multiagentes;
4) O quarto capítulo, explica a metodologia específica de suporte à derivação de sistemas específicos, assim como o sistema de apoio implementado;
5) O quinto capítulo, descreve o estudo de caso que serviu de base à validação das propostas efectuadas e apresenta e analisa os resultados experimentais obtidos;
6) Finalmente, o último capítulo resume as propostas e trabalhos apresentados, discute as conclusões obtidas e apresenta as perspectivas de trabalho futuro.

A) Os anexos apresentam demonstrações e listagens que, devido à sua relevância, são adicionados à dissertação por forma a permitirem uma consulta acessível.

A leitura desta dissertação deve proporcionar uma visão global da problemática da catalogação de dados armazenados na Internet, das soluções existentes e das propostas do autor, que têm como objectivo dar resposta aos problemas identificados.

2 Estado da Arte

O início da dissertação é dedicado à apresentação do Estado da Arte das áreas mais relevantes para a realização dos trabalhos efectuados. A próxima secção apresenta um breve historial da Web e dos actuais sistemas de pesquisa, vulgarmente utilizados pelos cibernautas, sendo referida a necessidade de adopção de técnicas de exploração de conhecimento para a Web, por forma a permitir a sua evolução. As secções seguintes, aprofundam assuntos determinantes para a obtenção de uma nova geração de ferramentas de suporte ao utilizador, especificamente: *i)* a aprendizagem em texto; *ii)* a representação de conhecimento; e *ii)* os sistemas baseados em multiagentes.

2.1 A Web

A Web é a plataforma-base responsável pelo desafio que serve de mote aos trabalhos apresentados nesta dissertação. Neste sentido, esta secção é dedicada à caracterização da Web e dos sistemas de pesquisa utilizados pelos cibernautas, identificando possíveis progressos e os métodos de exploração de conhecimento passíveis de serem aplicadas para a sua evolução.

O conceito World Wide Web (WWW) foi desenvolvido no CERN – *Center for European Nuclear Research* em 1989 por Tim Berners-Lee and Robert Cailliau [5, 6]. Apesar de ter começado como uma ferramenta de partilha de dados para investigadores, evoluiu para a maior e mais rica rede de conhecimento partilhado, conduzindo a Humanidade a uma era de conhecimento global. Os documentos são disponibilizados em formato HTML – Hypertext Markup Language, i. e., em hipertexto, criando com o recurso aos elos[5], uma rede de informação baseada em referências cruzadas, que permite a interligação de documentos, e assegura a inclusão de conteúdos de imagem, som, e vídeo, muito para além do texto.

[5] Elos – Tradução de «Hyperlink».

Foi com a Web que a Internet deixou de ser uma gigantesca rede de computadores que interligavam elitistamente laboratórios de investigação e empresas internacionais, permitindo a partilha de dados, para iniciar o processo de democratização. Desta forma, foi possível cativar milhões de utilizadores dispersos por todo o mundo, criando o maior repositório de conhecimento alguma vez disponível para a Humanidade. As últimas estimativas, indiciam um número total de 10 mil milhões de documentos, o que representa um acervo documental avassalador [7]. Na realidade, até ao momento, só este meio foi capaz de disseminar e disponibilizar conhecimento de forma tão democrática e expedita que, em poucos anos, ultrapassou «tudo e todos».

Todavia, a estrutura dos documentos em rede, através dos elos, apesar de permitir uma fácil navegação, não se adequa à pesquisa de informação. Identificar a informação relevante tornou-se uma tarefa complexa e quase impossível de realizar sem o recurso a sistemas de pesquisa. Em pouco tempo, os sistemas de pesquisa deixaram de ser meros protótipos obtidos no decurso de projectos de investigação, passando a oferecer a utilizadores de diversos graus de experiência, a porta de entrada na Web.

A monitorização de 125 mil sítios, através da plataforma analítica HitBox da WebSideStory's, permitiu ao StatMarket a representação dos hábitos de navegação de 12 milhões de utilizadores da Internet [8]. A Tabela 2 apresenta um resumo dos valores obtidos que confirmam a crescente influência dos motores de pesquisa, e a redução drástica da utilização dos elos como forma de pesquisa de informação.

Tipo de referência	2002 (%)	2003 (%)
Navegação directa	50,12	65,48
Elos Web	42,60	21,04
Motores de pesquisa	7,18	13,46

Tabela 2 – Dados resultantes do comportamento de navegação de 12 milhões de utilizadores da Internet

Todavia, o acréscimo em mais de 15 por cento da navegação directa, para um valor superior a dois terços do total dos acessos, confirma não só a dificuldade de utilização dos elos como método de pesquisa, como ainda uma insatisfação dos utilizadores em relação aos sistemas actuais.

2.1.1 Actuais sistemas de pesquisa para a Web

Os actuais sistemas de pesquisa podem ser vistos como verdadeiros sistemas globais, extravasando, em muito, o domínio físico de uma biblioteca ou de uma rede de bibliotecas e atingindo um universo esmagadoramente superior de documentos.

Existe um conjunto alargado de iniciativas com o objectivo de contornar as dificuldades encontradas na pesquisa de informação. No âmbito das soluções apresentadas na óptica da criação de sistemas de suporte à pesquisa realçam-se, entre os mais bem sucedidos, os baseados em palavras-chave. A simplicidade de utilização aliada a resultados aceitáveis transformou estes sítios em portas de acesso. De uma forma genérica, após a introdução de um conjunto de palavras-chave, o sistema encarrega-se de devolver um conjunto de documentos que, potencialmente, contêm a informação solicitada. O sucesso desta operação está intimamente relacionado com a sua eficácia, mas, igualmente, com a experiência do utilizador na selecção acertada de palavras que identifiquem, inequivocamente, os assuntos que deseja. Apesar de existirem casos de especialização, a maioria dos sistemas são multidomínio, encarregando-se de analisar toda a Web.

Os primeiros portais de pesquisa a surgir foram os directórios Web que implementaram um mecanismo semelhante ao das páginas amarelas. Para além de um processo de busca por palavra-chave é possível procurar informação por navegação em árvores de categorias. A utilização das categorias, baseia-se num processo incremental, iniciado pelas mais genéricas e conduzindo, progressivamente, às mais específicas. Esta funcionalidade está fortemente relacionada com a catalogação manual das páginas em categorias, exigindo uma forte intervenção humana directa.

O Yahoo! é um dos mais antigos serviços de directório na Web, tendo iniciado o seu funcionamento em Agosto de 1994. Os documentos foram integralmente classificados por um processo manual, numa taxinomia de termos, até Outubro de 2002, estimando-se que, somente 4 por cento dos sítios submetidos foram indexados. Nessa data, o Yahoo! substituiu as listas de directoria compiladas manualmente por pesquisas no Google, o que, apesar de lhe permitir manter a supremacia, ofereceu ao Google uma visibilidade que se veio a tornar prejudicial. Desde então, procurou endogeneizar tecnologia de pesquisa e indexação tendo, inclusive, adquirido o motor de pesquisa Inktomi. Surpreendentemente, em Fevereiro de 2004 o Yahoo! anunciou a substituição do Google, não pelo Inktomi, mas sim por um novo motor de pesquisa desenvolvido internamente. Actualmente, o Yahoo! continua a disponibilizar o directório actualizado de forma híbrida, permitindo pesquisas de documentos através do seu novo motor de pesquisa, de imagens através do Google e de notícias através do seu directório de categorias [9].

O LookSmart é outro dos sobreviventes aos conturbados anos de aquisições em cadeia. Uma vez mais, o que distingue este portal é o seu sistema de directorias que facilita uma pesquisa por assuntos. A sua colecção de documentos manualmente classificados permite assegurar uma elevada qualidade de resultados [10].

Contudo, com a dinâmica da Web e a dimensão dos documentos disponibilizados, os sistemas de maior sucesso são os motores de busca. Os portais baseados nesta tecnologia

possuem programas de busca e classificação de páginas de forma automática, chamados navegadores[6]. Nestes sistemas existe pouca intervenção humana na indexação das páginas, o que viabiliza o cadastro de um número gigantesco de páginas. Em consequência da abrangência do tipo de informação armazenada, podem ser considerados **especialistas** ou **generalistas**. Os **motores generalistas** mais conhecidos são o AltaVista (www.altavista.com), o Google (www.google.com), o Lycos (www.lycos.com), o AlltheWeb (www.alltheweb.com), o AOL (search.aol.com/aolcom/index.jsp), e o Gigablast (www.gigablast.com) devido às suas características de indexação de páginas multidomínio. Todavia, apesar dos sítios manterem uma identidade própria, partilham a tecnologia de motor de pesquisa que é propriedade do Google e do Yahoo!. Em [11], na lista dos 10 melhores motores de pesquisa, somente o Gigablast possui tecnologia própria, os restantes portais utilizam uma das duas soluções.

O Altavista continua a marcar a diferença, visto funcionar como um sistema de páginas amarelas da Web e realizar a indexação de páginas completas. Para além de ser um dos primeiros motores de pesquisa baseado em palavra-chave, criou o primeiro motor de pesquisa multilingue e, posteriormente, inovou uma vez mais, com a introdução de capacidades de pesquisa em documentos de língua chinesa, japonesa e coreana, através do seu tradutor Babel Fish.

Porém, nos últimos anos, o Google afirmou-se como o motor mais poderoso. Distinguiu-se pela quantidade de documentos catalogados, bem como no cálculo da sua relevância para cada pesquisa específica. Os documentos são seleccionados por grau de semelhança entre os termos da pergunta e a representação do documento. Todavia, a ordem de apresentação dos documentos é condicionada por um sistema de cotação que privilegia os documentos mais referenciados em detrimento dos documentos isolados. A «referenciação» é efectuada em função do conjunto de elos que apontam para o documento em causa. Tendo em conta que cada pesquisa pode identificar milhares de documentos, o Google investiu num mecanismo de ordenação com vista a privilegiar os mais relevantes para o utilizador. Em Fevereiro de 2004 o Google anunciou a indexação de 4,3 mil milhões páginas Web [7].

Contudo, o domínio do Google está constantemente a ser posto à prova com o surgimento de novas iniciativas resultantes da introdução de novas técnicas e métodos de indexação. A mais recente, em Março de 2004, foi lançada pelo Yahoo!, de novo em busca da posição de topo.

Os **motores de pesquisa especialistas** são menos famosos, em consequência da sua especificidade. Na área da investigação, o CiteSeer (http://citeseer.nj.nec.com/) é um dos mais conhecidos pela sua característica indexação de páginas de artigos científicos

6 Navegadores – Tradução de «Crawler»

publicados. Todavia, as grandes universidades, os melhores laboratórios de investigação e multinacionais também possuem motores de pesquisa especialistas de extrema utilidade nas sua áreas de actuação.

Apesar do aparecimento desta multiplicidade ao longo dos tempos, existe uma matriz comum característica, ilustrada na Figura 3, composta pelos sistemas de navegadores, de armazenamento de dados, de filtragem, de indexação e de interpretação de pesquisas.

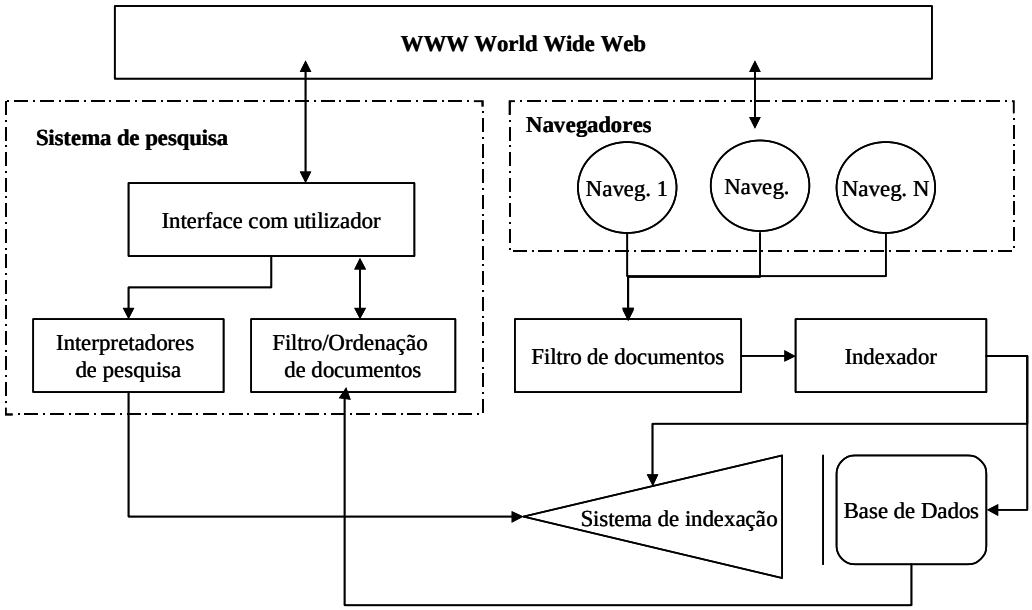


Figura 3 – Arquitectura genérica dos sistemas de pesquisa

O sistema de navegadores tem por objectivo navegar na Web e fazer o carregamento local da informação, efectuando em tempo diferido a localização, recolha e actualização de documentos. Têm a constante tarefa de manter páginas actualizadas o que os obriga a percorrer, em contínuo, toda a Web.

O sistema Indexador de documentos é responsável por extrair as palavras que passarão a representar o documento no momento da pesquisa. A solução típica passa por seleccionar o conjunto de palavras que melhor definam o documento. A selecção das palavras é crítica, tendo em conta que, quanto menor for o seu número menor será o sistema de indexação e menor será a relação com o documento. Existem algumas técnicas de compactação que são utilizadas, tais como a passagem de todas as letras a minúsculas, redução das palavras à sua raiz morfológica, e a utilização de listas de palavras frequentes, i. e., listas paragem.

O sistema de armazenamento de dados é composto pelo subsistema de indexação e por uma base de dados para páginas carregadas. Usualmente, o sistema de indexação está baseado numa filosofia de indexação de ficheiros invertida, i. e., a localização do ficheiro é realizada através das palavras que o compõem. O índex é previamente construído no momento da adição de novos documentos e optimiza a pesquisa, mas torna complexo o

processo de manutenção e actualização da base de dados. Tipicamente, o sistema de índex de ficheiro é uma estrutura em árvore de palavras que representam o documento; nas folhas estão armazenadas as palavras, a sua posição e a localização do documento. Este é o módulo crítico do sistema de pesquisa não sendo divulgado em detalhe. Todavia, a sua implementação é baseada na combinação de algoritmos de indexação, entre eles, tabelas de hash e árvores.

O sistema interpretador de pesquisas tem a seu cargo a responsabilidade de receber os dados fornecidos pelo utilizador e procurar os documentos que mais se assemelham, por comparação com a representação adoptada. Os documentos mais semelhantes são, então, apresentados e ordenados por grau de semelhança. As pesquisas mais comuns são feitas através:

- Palavras : Esta é a pesquisa-base e procura localizar documentos que contenham as palavras introduzidas. As palavras são utilizadas na pesquisa directa no índice. Esta pesquisa, por vezes, é enriquecida pela associação de semântica a cada palavra, i. e., pela descrição da utilização da palavra no documento: texto, título, nome de imagem, nome de documento, etc.;
- Combinação lógica de palavras: Uma sofisticação da pesquisa anterior passa por permitir a utilização de operadores lógicos entre as palavras, tipicamente a conjunção, a disjunção e a negação com o significado da não existência do termo. Passa a ser possível pesquisar documentos que contenham «todas as palavras», «uma das palavras», «nenhuma das palavras», etc.;
- Frases: Nesta pesquisa, as palavras introduzidas são interpretadas como frases, sendo a pesquisa realizada na procura de documentos que contenham a sequência de palavras introduzidas. A forma tradicional de indicar a utilização de uma frase é através do recurso às aspas, forçando a identificação da sequência de palavras. Uma variante menos comum, mas muito eficaz, apesar de pouco intuitiva, é a descrição da frase pelo recurso a métricas de proximidade. Neste caso, para além de se indicar as palavras, é necessário descrever a distância máxima admitida, por indicação do número máximo de caracteres ou palavras possíveis entre as mesmas;

As pesquisas anteriores, apesar de intuitivas para utilizadores habituados à utilização das tecnologias de informação, provaram ser uma barreira para utilizadores pouco experientes, por ser extremamente crítico para a obtenção de sucesso na pesquisa, a correcta selecção de palavras e de operadores. Os motores de pesquisa tecnicamente mais sofisticados procuram oferecer novas interfaces enriquecidas em que é permitido ao utilizador a realização de perguntas em Linguagem Natural. Sistemas como AskJeeves (http://www.ask.com/) ou ElectricMonk (http://www.electricmonk.com/) permitem a utilização

de frases na forma de pergunta, e. g., «Qual o endereço do Museu da Cidade?». A maior parte das implementação deste tipo de interfaces continua, todavia, a basear-se no armazenamento de uma quantidade esmagadora de perguntas-tipo, permitindo simular a capacidade de interpretação das mesmas, quando na realidade o motor de pesquisa continua a utilizar somente as variações acima descritas.

Finalmente, **o sistema de filtragem** (o filtro de documentos) é responsável por seleccionar os documentos relevantes, quer da lista de documentos potenciais a serem devolvidos ao utilizador, quer, numa fase anterior, dos documentos identificados pelos navegadores.

Complementares aos motores de pesquisa surgiram os meta-motores de pesquisa. Procurando explorar as vantagem parciais oferecidas por cada um dos motores disponíveis surgiu, assim, uma nova classe de portais. Estes portais não possuem, actualmente, uma relevância determinante, devido à forte diminuição do espectro de oferta devido ao processo de aquisições sucessivas. Na prática, estes sistemas funcionavam como interfaces entre o utilizador e um conjunto de motores de pesquisa, recebendo a pergunta do utilizador, seleccionando o conjunto de motores que, potencialmente, podem responder, com mais acuidade ao utilizador, adaptando a pergunta aos formatos dos motores de pesquisa específicos e, finalmente, filtrando as respostas e apresentando-as ao utilizador. Entre os mais conhecidos destacam-se o Inquirus (inspector.nj.nec.com) desenvolvido pelo NEC Research Institute [12]; o Metacrawler (www.metaCrawler.com); o sherlockHound, SavvySearch, Inference Find, Fusion, ProFusion, Highway 61, Mamma, Quarterdeck WebCompass, Metabot, Symantec Internet FastFind, and WebSeeker. (Para uma visão mais extensa sobre meta-motores de pesquisa consultar [13].)

Simplesmente a título de exemplo, o MetaCrawler transferia as perguntas para o Google, Yahoo!, AltaVista, AskJeeves, About, LookSmart Overture e o Findwhat, filtrando as respostas para posterior apresentação ao utilizador. A diversidade de soluções inicial ficou seriamente comprometida pela concentração de actores disponíveis no mercado dos motores de pesquisa, tendo a maioria desaparecido.

Em Maio de 2003 o Yankee Group defendeu que, após o processo de consolidação do mercado, existem, somente, três grandes competidores o – Yahoo!, Overture e o Google, em consequência do recente desaparecimento do Inktomi, e da aquisição do Altavista e FATS Web Search pelo Overture [14]. Esta afirmação é prematura, tendo em conta que ignora por exemplo o MSN que, pela sua poderosa capacidade de pesquisa, deve ser incluído entre os quatro maiores. O argumento, de que o MSN não possui tecnologia de pesquisa própria é, unicamente, mais um factor indicativo de que um novo processo de consolidação, por aquisição de actores de menor dimensão, tais como o LookSmart ou o Ask Jeeves, pode estar ainda por acontecer. Todavia, o número de motores de grande dimensão diminuiu sensivelmente, estando reduzido a um conjunto restrito. Em Novembro

de 2003 o Yahoo! anunciou a aquisição do Overture o que reduz, ainda mais, o já de si diminuto clube de motores de pesquisa de grande dimensão.

Em Fevereiro de 2004, a comScore Media Metrix relatava que, em Dezembro de 2003, o Google era responsável por efectuar 35 por cento das pesquisas da Web, comparado com 27 por cento dos portais do Yahoo! e 15 por cento da Microsoft. O AOL e outros portais da Time Warner representavam 16 por cento das pesquisas, maioritariamente asseguradas através da utilização do Google.

Todavia, esta área está longe de se encontrar esgotada e são constantes as iniciativas para afirmarem novos portais que, apesar de estarem ainda em fase experimental, visam encontrar soluções alternativas aos tradicionais motores de pesquisa baseados em palavra-chave.

A criação de interfaces mais intuitivas, através do processamento de linguagem natural é uma área em franca evolução. Nestes sistemas, o utilizador é convidado a inserir frases livres em linguagem natural, estando o sistema encarregue de realizar a sua interpretação, traduzi-las em perguntas e fornecer as respostas que melhor se adequem.

Outra área promissora é a das pesquisas multimédia, baseadas em processamento de imagem ou catalogação prévia de conteúdos. As soluções, até agora apresentadas, estão limitadas a pesquisas em texto, o que é extremamente redutor, tendo em conta que a Web é composta por componentes multimédia onde, apesar do texto continuar a ser determinante, é simplesmente um dos componentes possíveis. Para além do texto, por exemplo, o som, as imagens ou vídeo são componentes que contêm informação relevante e que são ignorados nas pesquisas de texto. Os sistemas multimédia procuram identificar soluções que permitam ao utilizador pesquisar imagens que contenham determinados elementos gráficos, e. g., árvores, barcos, pessoas, casas. Os trabalhos em curso estão ainda numa fase inicial, todavia a possibilidade de existência de sistemas que pudessem fazer o reconhecimento automático de elementos gráficos, abriria novas possibilidades à exploração dos dados e à recuperação de informação.

Finalmente, um último exemplo são os sistemas de pesquisa baseados em perguntas, que procuram criar linguagens formais de alto nível, viabilizando, à semelhança das linguagens de interface com as bases de dados, e. g., SQL (Structural Query Language), aumentar o nível de abstracção das pesquisas. O maior obstáculo a esta abordagem é a inerente falta de estruturação da Internet que não se adequa a pesquisas semelhantes às possíveis nas bases de dados. A consolidação dos esforços de normalização da informação disponibilizada na Internet (permitindo juntar aos documentos a meta-informação) conduzirá à segunda geração da Web, viabilizando as aproximações descritas, e contribuindo para o

surgimento de novas ferramentas. Entretanto, são os sistemas baseados em palavras-chave que ocupam a primazia.

2.1.2 Exploração do conhecimento na Web

A evolução dos sistemas de pesquisa passa, inevitavelmente, pela utilização de técnicas de exploração de conhecimento[7] e representação de informação; (sobretudo dedicadas à Web), que englobem análise de dados, orientada para a descoberta de informação. A sua utilização permite examinar, de forma automática e exaustiva, gigantescos bancos de dados com o objectivo de identificar relações não evidentes entre registos, encobertas pela magnitude dos problemas. Esta tecnologia tem sido aplicada com sucesso, não só em projectos de investigação como, igualmente, em diversos campos de aplicação comercial, principalmente ligados aos seguros, medicina, finanças, publicidade, e em combate à fraude, permitindo fortalecer posições de mercado e revelar informações insuspeitas.

As técnicas utilizadas estão fortemente focadas na manipulação e transformação dos dados em análise, tais como classificação, agregação e associação ou reconhecimento de padrões, tendo sido, na maior parte dos casos, recuperadas dos campos da matemática, cibernética e genética, podendo ser utilizadas em conjunto ou em separado. O objectivo último é permitir identificar «respostas para perguntas que não sabemos fazer» pela extracção de informação de qualidade com vista à identificação de factos, conduzindo a conclusões baseadas em padrões ou relações dos dados. Exemplo paradigmático da sua utilização é realizado pelas seguradoras, com vista a identificar grupos de risco reduzido, possibilitando a criação de produtos que permitam atrair ou fidelizar clientes potencialmente lucrativos. Esta tecnologia, mais do que disponibilizar a capacidade de realizar pesquisas sobre relações conhecidas, visa descobrir relações ocultas, garantindo uma aprendizagem do próprio negócio.

As técnicas aplicadas podem ser agrupadas nas seguintes categorias [15]:

- Agregação/Classificação: Esta técnica visa a reunião de itens semelhantes a fim de permitir identificar as características e as relações relevantes, identificadoras das agregações realizadas, o que seria difícil de evidenciar na totalidade dos dados. Exemplos deste processo são a identificação de características marcantes de páginas de venda ou de grupos de utilizadores, que acedem a páginas específicas;
- Regras de Associação: Esta técnica visa a determinação de regras que permitam identificar sequências de acções, evidenciando as correlações entre itens, assegurando com um grau de certeza, que a presença de um item antecipa a presença de um outro. Exemplos da sua utilização são a antecipação das acções

[7] Exploração de conhecimento – Tradução de «Data Mining»

dos utilizadores, i. e., a consulta de uma página leva a prever a encomenda *on-line* de determinado produto;

- Análise de Padrões e Sequências: Esta técnica visa a análise de registos com vista a identificar sequências, modelo e padrões. Um exemplo da sua utilização é a identificação de caminhos críticos; i. e., quais os caminhos-tipo percorridos até à chegada a um determinado URL[8], ou a constatação de que uma percentagem de utilizadores que consultou um URL realizou um conjunto de acções, num determinado período.

A aplicação das técnicas de exploração de conhecimento aos dados disponíveis na Web é conhecida por exploração de conhecimento na Web[9], e tem um papel essencial na criação de sistemas que permitem ao utilizador encontrar a informação de que necessita de forma expedita e eficiente. Esta abordagem é, muitas vezes, apontada como uma segunda geração de tecnologias derivadas da inteligência artificial, uma vez que é baseada, de uma forma geral, em aprendizagem automática, com vista à identificação de padrões nos dados e à revelação de informações insuspeitas.

A Exploração de Conhecimento na Web permite a pesquisa de informação relevante, assim como a monitoração e previsão do comportamento do utilizador, faculta o auxílio à navegação na Web e a criação de sítios Internet personalizados. A sua aplicação possibilitou a criação de motores de pesquisa alternativos, que realizam uma classificação personalizada, e são capazes de ultrapassar a indexação insuficiente e incompleta da informação armazenada. A exploração de conhecimento na Web engloba as seguintes disciplinas [16]:

- Exploração de conhecimento sobre utilização da Web[10] onde se procuram padrões de acesso e utilização dos dados;
- Exploração de conhecimento dos conteúdos na Web[11] onde se pretende extrair informação de documentos da Web, extracção da informação apresentada ou, onde se visa identificar padrões na estrutura de organização de documentos.

A aplicação da exploração do conhecimento na Web, no contexto desta dissertação, conduziu à utilização de técnicas de aprendizagem, à utilização de meios formais de representação de conhecimento e à utilização de sistemas de multiagentes. Estes assuntos são apresentados nas secções seguintes.

---

[8] URL - Uniform Resource Locators.

[9] Exploração de conhecimento na Web – Tradução de «Web Mining».

[10] Exploração da utilização na Web – Tradução de «Web Usage Mining»

[11] Exploração de conteúdos na Web – Tradução de «Web Content Mining»

2.2 Aprendizagem automática

Ao longo dos últimos anos têm sido diversas as propostas apresentadas para a definição de aprendizagem. Simon definiu aprendizagem como a adaptação do sistema, no sentido de permitir, no contexto, a realização mais eficiente e efectiva da mesma tarefa ou de tarefas semelhantes [17]. Langley definiu aprendizagem como o melhoramento do desempenho num ambiente, pela aquisição experimental de conhecimento sobre o mesmo [18]. Michalski apresenta a «Inferential Theory of Learning» (ITL), na qual define, aprendizagem como uma pesquisa no espaço do conhecimento, onde pesquisa significa a transmutação de conhecimento para atingir um determinado objectivo de aprendizagem [19]. Mitchell apresenta uma definição mais operacional defendendo que um programa de computador aprende, a partir de uma experiência *E*, relacionada com uma classe de tarefas *T*, e com uma medida de desempenho *D*, caso a execução das tarefas de *T*, mediadas por *D*, melhorem com a experiência *E* [20]. Comum às definições anteriores é o facto da adaptação dos sistemas resultar da experiência adquirida, obtendo-se como resultado uma melhoria do desempenho do sistema na execução de tarefas semelhantes.

Tradicionalmente, a aprendizagem consiste na indução de descrições para conceitos gerais. Parte-se de um conjunto de exemplos, adequados e significativos, para a generalização das características comuns. As descrições são utilizadas para reconhecer casos futuros [21]. Geralmente, a indução é estatisticamente demonstrável, tendo em conta que as descrições obtidas agregam conjuntos de exemplos com características comuns. Justificadamente, a indução é muitas vezes vista como uma tarefa de generalização. As dificuldades particulares da aprendizagem indutiva são causadas, por exemplo, pelo ruído excessivo (i. e., dados irrelevantes), por conjuntos de treino não representativos, por limitações da linguagem de descrição (falta de termos) e pela aprendizagem incremental.

Genericamente, a aprendizagem pode ser definida como a aplicação a um sistema-base da inferência produzida por um sistema de aprendizagem, tendo por base sequências de treino, que consistem em tuplos elemento de experiência, acção e avaliação. Por outras palavras [22]:

- Elementos de experiência $(e \in E)$;
- Espaço de acções disponíveis $(a \in A)$;
- Desempenho $d(a,e)$;
- Sistema-base $b : E \rightarrow A$;
- Sistema de Aprendizagem $s : (e_1,a_1,d_1),\ldots,(e_n,a_n,d_n) \rightarrow b$.

A aprendizagem está focada em problemas pouco estruturados e é baseada em métodos de pesquisa. A flexibilidade verificada nos métodos de aprendizagem assegura a sua

adequação a problemas em que não existe um grande conhecimento prévio do domínio, ou onde este é difícil de representar. Esta capacidade permite, muitas vezes, a utilização de bancos de dados que não foram criados com o intuito de serem utilizados, por forma a permitirem a inferência de conhecimento. Naturalmente, esta flexibilidade dificulta a validação teórica dos métodos de aprendizagem e a garantia da correcção dos resultados obtidos, que é compensada, muitas vezes, por verificação empírica. O reconhecimento desta limitação é frequentemente reflectido, teoricamente, pela apresentação de resultados na forma de intervalos de generalização do erro do estimador, dado que o seu erro empírico e o espaço efectivo de pesquisa, resulta na apresentação de resultados com taxas de erro em intervalos estipulados, sob uma garantia probabilística [23].

As aplicações de aprendizagem estão fortemente condicionadas pela disponibilidade da capacidade computacional e pela adequação dos dados, (elevadas quantidades de dados não correspondem, obrigatoriamente, a adequada representatividade). A incapacidade computacional conduz, usualmente, a ajustamento insuficiente[12], e a falta de representatividade ao sobreajustamento[13]. Por vezes, os dois fenómenos podem estar presentes em simultâneo.

Os métodos são frequentemente agrupados pelas suas características intrínsecas, sendo as divisões mais comuns as seguintes:

- **Métodos de aprendizagem de tempo diferido e de tempo real.** Na aprendizagem em tempo diferido, existem duas fases essenciais: **o treino** e **a utilização**, que correspondem a sistemas de síntese de modelos, abordando o problema numa perspectiva de exploração do espaço de hipóteses. Em contraponto, na aprendizagem em tempo real não existem fases separadas, sendo a aprendizagem incremental e resultado da utilização. Neste caso, é essencial determinar o ponto de equilíbrio entre o agir de forma acertada e o ganhar de experiência, para obter um melhor desempenho no futuro;
- **Métodos de aprendizagem com base em informação completa ou incompleta.** No caso dos métodos com informação completa para cada item de experiência são avaliadas todas as acções com o objectivo de identificar qual a melhor solução. Os métodos de informação incompleta para cada experiência, só avaliam um subconjunto de acções, disponibilizando o resultado obtido;
- **Métodos de aprendizagem causal** onde as acções não afectam futuras experiências (e. g., previsão do tempo), *versus* não causal em que as acções produzem reflexos nas experiências futuras;

[12] Ajustamento insuficiente – Tradução adoptada para «underfitting»

[13] Sobreajustamento – Tradução adoptada para «overfitting»

- **Métodos de aprendizagem em ambientes estáticos**, em que as premissas não se alteram *versus* ambientes dinâmicos, onde a avaliação das experiências sofre alterações;

Existem diversos processos de aprendizagem que optam por distintas representações e níveis de abstracção, que podem ser agrupados em: *i)* estatística tradicional que opta por representações $h : \mathbb{R}^n \rightarrow \mathbb{R}$ com determinação de erro quadrático e em que *h* é uma função linear; *ii)* reconhecimento de padrões que foca a sua atenção em representações $h : \mathbb{R} \rightarrow \{0,1\}$, com avaliação de certo ou errado, e em que *h* é discriminante linear; *iii)* aprendizagem simbólica representa $h : \{(a_1,\ldots,a_n),\ldots,(z_1,\ldots,z_n)\} \rightarrow \{0,1\}$ em que *h* é uma simples função linear; *iv)* redes neuronais $h : \mathbb{R}^n \rightarrow \mathbb{R}$ em que *h* é uma rede neuronal retroalimentada; *v)* programação de lógica indutiva $h : \{estrutura\_de\_termos\} \rightarrow \{0,1\}$ em que h é um programa lógico.

2.2.1 Aprendizagem supervisionada

A aprendizagem supervisionada começa a ser uma área científica consolidada, em especial no caso de espaços de reduzida dimensionalidade. Existe actualmente, um conjunto de metodologias, métodos e funções, que começam a ser intensivamente utilizadas em ferramentas comerciais [24].

Informalmente, a principal tarefa da aprendizagem supervisionada é a inferência de um modelo, a partir de um conjunto de exemplos previamente classificados. O modelo passa, então, a ser utilizado para estimar a classificação de novos exemplos. Os modelos gerados podem ser avaliados em função da sua eficiência, eficácia, compactação e compreensão. Esta tarefa é apelidada de aprendizagem supervisionada, por contraste com a aprendizagem não supervisionada ou agregação, em que os exemplos não estão previamente classificados.

A tarefa de aprendizagem pode ser descrita como a pesquisa de um modelo que aproxime uma função desconhecida que aplica um conjunto de variáveis independentes *X*, numa variável dependente *Y*, por outras palavras, sendo

$$f : X \rightarrow Y , \tag{1}$$

a função que descreve o fenómeno que se pretende aproximar, e dado o conjunto de desenho,

$$D = \{\vec{x}_m, y_m\}, m \in [1, N], \tag{2}$$

em que $\{\vec{x}_m, y_m\}$ representa o conjunto de pares ordenados de observações *x*, classificadas em *Y*, e uma algoritmo indutor de classificadores *F* , obtendo-se um modelo estimador de $\hat{f}$

$$\hat{f} = F(D) . \quad (3)$$

A aprendizagem supervisionada parte, assim, do princípio da possibilidade de inferir a classificação de uma observação, tendo por base um registo histórico de observações pré-classificadas do fenómeno em avaliação. Existem duas grandes categorias de aplicação do problema da aprendizagem supervisionada: *i)* a problemas de classificação, (na estatística, usualmente denominadas de análise discriminante), em que cada observação é classificada por um elemento de um conjunto finito $f(x) \in \{S_1, S_2, \ldots, S_k\}$; e *ii)* os problemas de regressão, em que cada observação tem uma classificação em valores reais, $f(x) \in \mathbb{R}$. O trabalho efectuado nesta dissertação centra-se, exclusivamente, no primeiro caso, tendo em conta que o objectivo é permitir a classificação dos documentos numa taxinomia de classes.

Independentemente das aplicações, os métodos de indução podem ser caracterizados pelo seu **esquema de representação** (*R*) e pela sua **estratégia de pesquisa do espaço** de hipótese (*P*). Por **esquema de representação,** entende-se a estrutura de decisão que permite a generalização dos exemplos e restringe o conjunto de hipóteses analisadas pelo algoritmo. Por **estratégia de pesquisa,** entende-se o conjunto de métodos e heurísticas que permitem guiar o algoritmo na pesquisa no espaço de hipóteses, com o objectivo de seleccionar a hipótese que melhor aproxima a função-objectivo. Juntos, a representação e a estratégia de pesquisa, definem, univocamente, um método de indução, condicionando os modelos inferidos. Esta dependência pode ser expressa, pela rescrita da equação (3), em

$$\hat{f} = F_{\{R,P\}}(D) \text{ [25].} \quad (4)$$

Neste sentido, a aplicação de distintos métodos de indução a um conjunto de desenho, permite a definição de um conjunto de modelos, que apresentam desempenhos muito distintos na classificação de novas observações. Na verdade, dependendo do domínio e da natureza dos dados, o mesmo método de indução apresenta desempenhos distintos, não existindo evidência de superioridade absoluta (independente do domínio de aplicação). Este facto é conhecido como o problema de superioridade selectiva [26]. Por outras palavras, o sucesso de um método está claramente comprometido pelo facto das suas tendências de indução serem adequadas ao problema em análise.

Exemplos de áreas em franco desenvolvimento em aprendizagem supervisionada são:

- **Aprendizagem activa/Desenho experimental:** Focada na selecção das experiências a efectuar, com vista à obtenção dos melhores resultados, e. g., selecção dos dados que permitam avaliar correctamente o perfil de clientes, e quais os melhores dados a utilizar na aprendizagem de robôs;
- **Aprendizagem cumulativa:** Aborda a criação de processos que permitam sintetizar os dados obtidos a partir de bases de dados incrementais, i. e., bases de dados que estão constantemente a receber novos dados;
- **Aprendizagem a partir de dados não classificados:** Procura de processos de aprendizagem que, partindo de um conjunto mínimo de dados classificados consigam iniciar o processo de inferência, que é consolidado com a utilização de dados não classificados. Esta disciplina visa ultrapassar o problema da morosidade e/ou dificuldade de obter dados devidamente classificados;
- **Aprendizagem relacional:** Procura identificar métodos de descoberta de relações entre dados, assumindo a inexistência de um conjunto fixo de características que permitam a sua caracterização. Estas abordagens permitem a descoberta de relações entre entidades distintas, sem o recurso à existência de vectores de características que as descrevem;
- **Aprendizagem com grandes bases de dados:** Identificação de processos de aprendizagem que permitam explorar grandes bases de dados cujas dimensões impossibilitam a utilização dos métodos tradicionais. Como seleccionar os exemplos representativos e qual o número de exemplos necessário, são alguns dos problemas abordados;
- **Aprendizagem com pequenas bases de dados:** Identificação de processos de aprendizagem que permitam explorar pequenas bases de dados permitindo a inferência de conhecimento;
- **Aprendizagem com o conhecimento prévio:** Como integrar conhecimento prévio sobre o fenómeno em estudo. A procura de processos que permitam integrar o conhecimento previamente existente sobre o fenómeno, com o conhecimento inferido na utilização de métodos estatísticos ou outros;
- **Aprendizagem sobre dados mistos:** Métodos de aprendizagem que permitam a inferência de conhecimento a partir de dados armazenados em formatos distintos, e. g., os dados médicos de um indivíduo existem em formato texto, vídeo, radiológico, etc. Estuda a viabilidade de integrar resultados obtidos a partir de dados em formato distinto, pela identificação de algoritmos que trabalhem a um nível mais abstracto, de modo a permitir a utilização dos dados em formato nativo;

- **Aprendizagem de relações de causalidade:** Métodos que, para além de detectarem correlações, permitam inferir causalidade. Estudam quais os tipos de assumpções necessárias para extrair relações causais de bases de dados puramente factuais, e quais as implicações resultantes.

2.2.2 Aprendizagem em texto

Com o uso generalizado dos computadores e o aumento da sua capacidade de computação, procurou-se identificar processos que permitissem o processamento automático de documentos, com vista à sua classificação e extracção de informação. O passar do tempo tem demonstrado a dificuldade da tarefa, pela dificuldade inerente ao processamento da linguagem natural, e pelo problema intelectual de formalizar o processo de classificação de informação. A introdução de novas técnicas e da maior capacidade computacional permite antever um contínuo progresso na tarefa de processamento de linguagem; todavia, replicar o processo intelectual de classificação automática continua a ser uma barreira difícil de ultrapassar. A dificuldade extravasa a definição do processo de extracção da informação sintáctica e semântica que permite a classificação da informação, residindo na complexidade de definição de relevância. Intelectualmente, é possível definir a relevância de um documento para uma pergunta, todavia a construção de sistemas automáticos obriga à definição de modelos onde definições de relevância possam ser quantificadas. A aplicação das técnicas de aprendizagem e extracção de conhecimento, a documentos de texto não estruturado, para extracção de padrões interessantes e não triviais é, usualmente, apelidada de aprendizagem em texto. Esta disciplina assume um elevado valor, tendo em consideração que o texto é a forma mais natural de armazenamento de informação. Estudos recentes defendem que oitenta por cento da informação registada de uma empresa está armazenada em texto [27]. A natureza não estruturada e difusa dos dados de texto conferem a esta disciplina uma dificuldade acrescida. São duas as técnicas utilizadas: *i)* a recuperação de informação[14] (IR) que classifica um texto desconhecido, pela aplicação de regras inferidas, a partir de um conjunto de textos previamente classificados e; *ii)* a extracção de informação[15] (IE) que processa textos desconhecidos produzindo um registo formatado e não ambíguo da informação, para apresentação directa ao utilizador ou para registo numa base de dados para posterior utilização. A recuperação de informação é, muitas vezes, utilizada para filtrar e seleccionar o conjunto total de textos disponíveis, cabendo à extracção de informação, a posterior análise dos textos seleccionados para extracção da informação.

[14] Recuperação de informação – Tradução para «Information Retrieval»

[15] Extracção de informação – Tradução para «Information Extraction»

Genericamente, as linhas de desenvolvimento na aprendizagem em texto são:

- o desenvolvimento de técnicas de linguagem natural que possam ser aplicadas para o melhoramento do desempenho dos métodos de aprendizagem;
- a criação de métodos de análise de padrões temporais em texto, que permitam explorar conteúdos, com vista ao reconhecimento de acontecimentos;
- a incorporação de conhecimento prévio nos métodos e algoritmos de aprendizagem;
- a criação de sistemas que aprendam a apresentar informação, alterando a estrutura e organização da informação fornecida;
- a combinação de evidências provenientes de múltiplas fontes de informação:
- o pré-processamento de dados como preparação para optimização dos algoritmos aplicados;
- a melhoria das técnicas de representação, tradicionalmente baseadas em estruturas de dados de «saco de palavras». Esta aproximação perde muita informação existente na estrutura dos textos (a título de exemplo existe um elevado potencial na utilização de modelos estatísticos para grafos de objectos interligados).

A área da aprendizagem em texto, especialmente na Web, é multidisciplinar, integrando diversos tópicos de investigação que cobrem áreas tão diversas como a estatística, a recuperação de informação, o processamento de linguagem natural, o planeamento e a interface humano-computador.

2.2.3 Extracção de informação

A extracção de informação (IE) automática é uma tarefa complexa, computacionalmente muito intensiva e fortemente dependente do domínio, sendo pouco eficaz por comparação com a extracção manual. Esta dificuldade está relacionada com a liberdade da linguagem natural, que permite expressar o mesmo facto de formas muito distintas e ao longo de diversas frases de texto, o que obriga à sua análise combinada. Todavia, a hipótese de analisar gigantescos bancos de dados, (tarefa impossível de realizar de forma manual), torna esta técnica extremamente atractiva.

As tarefas-base para realizar a extracção de informação são: *i)* o reconhecimento de nomes para identificação e classificação de entidades; *ii)* a resolução de co-referências para descoberta da existência de referências entre entidades; *iii)* o reconhecimento de elementos que conduzam ao enriquecimento de informação descritiva das entidades; *iv)* reconhecimento de relações complexas; *v)* construção de cenários [28].

A eficácia das técnicas utilizadas está muito relacionada com o tipo de documentos, o domínio e os cenários estudados. Quanto mais livre for o tipo de texto, quanto mais vasto o domínio em análise e mais complexos os cenários estudados, menor é a eficácia

demonstrada. Os sistemas são fortemente adaptados aos textos que analisam, não sendo transportáveis para contextos distintos [29].

O reconhecimento de nomes é a tarefa mais eficaz, tendo-se atingido taxas de sucesso idênticas ou superiores às manuais, no reconhecimento de nomes de pessoas, lugares, organizações, datas e valores monetários. Contudo, as restantes tarefas são fortemente dependentes do domínio e da estrutura dos textos, e as suas taxas de sucesso são muito inferiores [30].

De seguida são apresentadas alguns exemplos de sistemas que realizam extracção de informação:

- O sistema GATE – General Architecture for Text Engineering, permite a análise de artigos de jornais com vista à identificação de fusões e criação de joint ventures entre empresas ou extrair informações sobre actividades terroristas, classificando o tipo de ataques, a identificação dos responsáveis ou suspeitos e vítimas, entre outros.
- O RAPIER, numa abordagem de generalização, procura extrair regras compostas por três componentes: *i)* um padrão que corresponde ao texto que precede o «padrão de extracção»; *ii)* «o padrão de extracção»; *iii)* o padrão que sucede ao «padrão de extracção». O posicionamento do padrão de extracção é realizado com o auxílio dos padrões que o precedem e sucedem, aumentando as possibilidades de sucesso. O processo-base visa identificar, para cada texto exemplo, uma regra que permita fazer a relação entre a informação e as componentes do padrão. Em seguida, para cada componente, procuram-se identificar as regras mais genéricas que são adicionadas à base de conhecimento e que farão parte do sistema de extracção de informação [31].
- O WHISK, que foi desenhado para inferir regras de forma automática, em textos estruturados e semiestruturados. A utilização de um analisador sintáctico e semântico permite a marcação em textos livres, consequentemente, a extracção de informação [32].
- Em [33], é apresentado um sistema de mediação de seguros que recorre à utilização de um modelo de negociação flexível, que incluí ofertas muti-atributo e capacidades de aprendizagem, para auxiliar o utilizador na escolha do plano mais adequado às suas necessidades.
- O SRV, que aplica uma aproximação mista e combina classificadores Naive Bayes com um indutor relacional. O sistema processa um conjunto de treino de documentos previamente marcados e um conjunto de características (verbos, valores numéricos, caracteres únicos, letras maiúsculas, etc.) que permitem a generalização de regras para extracção de informação em novos documentos [34].

- O Qxtract, que aposta na redução drástica dos ficheiros a analisar, com vista à optimização do esforço computacional, assumindo, assim, a existência de uma baixa probabilidade de inter-relação de assuntos entre documentos. A utilização de técnicas mistas de aprendizagem permite a inferência de regras, a partir de um conjunto de documentos, características e resultados iniciais esperados [35].
- O Snowball, (uma extensão do sistema DIPRE que utilizava simplesmente um algoritmo de «bootstrapping» com um conjunto de tuplos fornecidos pelo utilizador) acrescentou ao DIPRE o reconhecimento automático de padrões e a avaliação de tuplos para melhorar a qualidade da informação extraída [36].
- O Proteus/PET, desenvolvido na NYU «New York University», é um conjunto de ferramentas que permite efectuar extracção de informação explorando capacidades de personalização. O utilizador adapta o sistema através da descrição de cenários fornecendo exemplos de eventos em texto, e das regras a associar na base de dados. O sistema utiliza estes dados para generalizar os padrões apropriados, todavia, está baseado em treino manual intensivo [37, 38].
- Em [39] é apresentada a utilização de «Hiden Markov Models» (HMM) para extrair informação de texto livre, tendo sido desenvolvido um algoritmo que incorpora a estrutura gramatical das frases, em alternativa ao treino do modelo. Para maximizar a probabilidade entre os dados e o modelo, é maximizada a probabilidade de previsão da sequência correcta dos elementos informativos. Os sistemas baseados em HMM são menos comuns e enfrentam, muitas vezes, a inexistência de dados de treino suficientes.

Genericamente, os sistemas apresentados são muito dependentes do domínio, não existindo uma solução que apresente, por comparação, desempenhos esmagadoramente superiores. As vantagens comparativas estão focadas ao nível do domínio, na capacidade de lidarem com múltiplas ordenações dos elementos informativos e na capacidade de refinamento de regras.

2.2.4 Recuperação de informação

A área de recuperação de informação, ao longo desta dissertação, refere-se a sistemas de recuperação automática de informação. Apesar do nome, a verdade é que estes sistemas não informam (i. e., não alteram o conhecimento do utilizador), limitando-se a notificar o utilizador da existência e localização de informação. As secções seguintes apresentam alguns dos assuntos/tarefas mais relevantes no âmbito da recuperação de informação.

2.2.4.1 Avaliação

O processo de avaliação de desempenho dos algoritmos de aprendizagem representa um papel crucial, tanto para quem desenvolve o sistema, como para os utilizadores. Neste

sentido, é necessário identificar metodologias que permitam extrapolar, com alguma segurança, valores indicativos do desempenho das soluções propostas. Esta avaliação é importante para permitir a selecção dos modelos adequados e para prever o seu desempenho [40].

O desempenho pode ser analisado segundo duas vertentes essenciais: a eficiência e a eficácia. A medida da eficiência é, usualmente, determinada em termos da utilização de recursos computacionais, tais como tempo de CPU, espaço em disco, espaço em memória, etc. Esta avaliação é difícil de realizar de forma independente do equipamento utilizado e não é fácil de generalizar [41]. Por outro lado, a eficácia de um sistema está relacionada com a sua capacidade de obter bons resultados, o que, no caso do processo de reconhecimento de documentos, está relacionada com o acerto das classificações efectuadas. A avaliação do desempenho, realizada nesta dissertação, está relacionada, essencialmente, com a eficácia do sistema e engloba a tarefa da determinação de métricas e o seu processo de generalização, permitindo processos comparativos.

As métricas

É consensual que, na avaliação de um sistema de recuperação de informação, é necessário quantificar a capacidade de identificação de documentos relevantes e de documentos normais, erroneamente identificados como relevantes. Por outras palavras, determinar quantos documentos relevantes não foram identificados, e quantos dos documentos identificados como relevantes, não o são de facto [42, 43].

As duas situações, têm que ser analisadas com muita atenção. Pode ser tão importante identificar o maior número possível de documentos relevantes, (pois a sua não identificação é sinónimo de perda de informação), como evitar a falsa identificação de relevância (que conduz a perdas de eficiência do sistema). Existem, todavia, casos em que uma das situações, é significativamente mais relevante que a outra, sendo possível ignorar ou minimizar uma das análises. (exemplo: sistemas de alerta em que não se pode perder nenhum caso relevante).

As duas medidas-base, que permitem avaliar os casos descritos são:

- a **precisão** que visa a avaliação da capacidade de acerto de um sistema, i. e., avalia o número de classificações correctas. Uma interpretação possível da precisão, à luz da teoria das probabilidades, é a probabilidade de um documento ter sido classificado correctamente como relevante;

- a **rechamada**[16] que visa, por sua vez, a avaliação da capacidade de identificação de todos os documentos relevantes, i. e., avalia quantos documentos não foram reconhecidos. Uma interpretação do conceito da rechamada é a probabilidade de um documento relevante ser encontrado.

As medidas acima mencionadas são, muitas vezes, consideradas como avaliadoras de desempenho na perspectiva dos utilizadores, uma vez que medem a capacidade de apresentar somente os resultados apropriados (precisão) e a capacidade de identificar todos os resultados relevantes (rechamada).

A apresentação das duas medidas, para o caso de uma classificação binária, pode ser realizada através de **tabela de contingência**, também denominada **matriz de confusão**.

		Classificação estimativa	
		Negativo	Positivo
Classificação real	Negativo	a	b
	Positivo	c	d

Tabela 3 – Tabela de contingência para o caso de estimativas para duas classes

De uma forma genérica, a matriz de confusão pode ser descrita da seguinte forma

$$M_{ij} = \{ \hat{w} = c_j \,/\, w = c_i \}, \tag{5}$$

representando $w = c_i$ o número de documentos da classe i, e $\hat{w} = c_j$ o número de documentos para os quais foi estimada na classe j. No sistema ideal, a soma dos valores da diagonal principal é igual ao total, i. e., todos os documento são correctamente identificados, não sendo cometido nenhum erro. Por outras palavras, as medidas para o caso positivo podem ser traduzidas da seguinte forma:

$$Precisão = P = \frac{d}{b+d} \tag{6}$$

$$Rechamada = R = \frac{d}{c+d} \tag{7}$$

Com o objectivo de determinar uma medida global de desempenho é necessário conjugar os dois valores obtidos.

[16] Tradução do autor para «Recall».

Nos sistemas que apresentam uma ordenação dos resultados, em função da confiança na classificação estimada, é possível utilizar curvas de rechamada-precisão. Esta forma gráfica de representar o desempenho global do sistema utiliza, nas abcissas, o valor de rechamada e, nas ordenadas, o respectivo valor de precisão. Usualmente, estas curvas apresentam uma elevada precisão, para rechamadas na ordem dos 10 por cento, um equilíbrio para rechamadas de 50 por cento (em cenários complexos) e uma baixa precisão para rechamadas de 100 por cento [43].

Em sistemas que não estimam a credibilidade das classificações efectuadas, opta-se por uma combinação de métricas. Um dos processos de combinação mais utilizados é o F_β, que permite combinar as duas medidas, ponderando o seu peso relativo em função do valor atribuído ao parâmetro ß,

$$F_\beta = \frac{\left(\beta^2 + 1\right) \times R \times P}{\left(\beta^2 \times R\right) + P} \tag{8}$$

onde $ß \in [0,\infty]$. No caso de $ß=1$ estamos no caso particular proposto em [41], a medida

$$F_1 = \frac{2RP}{R + P} \tag{9}$$

onde se atribui igual relevância às duas medidas.

A relevância de um documento, num sistema que permita a classificação em diversos temas, é calculada a partir do grau de eficácia no reconhecimento de todas as classes. É, assim, necessário fazer a composição dos valores obtidos para cada classe com vista ao cálculo do valor da precisão e da rechamada. A forma de cálculo mais comum, utilizada na avaliação de classificadores, é baseada na *micro-averaged* que consiste no cálculo global, baseado nas decisões binárias para cada classe [44]. Assumindo a matriz de confusão descrita na equação (5), a precisão e a rechamada são calculadas, respectivamente:

$$Precisão = P = \frac{\sum_{i=0}^{|C|} w = c_i \;/\; \hat{w} = c_i}{\sum_{i=0}^{|C|} \hat{w} = c_i}, \tag{10}$$

$$Rechamada = R = \frac{\sum_{i=0}^{|C|} w = c_i \;/\; \hat{w} = c_i}{\sum_{i=0}^{|C|} w = c_i}. \tag{11}$$

De (10) resulta que a precisão é o rácio obtido pela divisão do número de documentos correctamente identificados, pertencentes a cada classe, pelo número de documentos identificados como pertencentes a cada classe.

De (11) resulta que a rechamada é o rácio obtido pela divisão do número de documentos correctamente identificados pertencentes a cada classe, pelo número dos documentos de cada classe.

Generalização de estimativas

Em teoria, não seria necessário realizar a estimativa dos erros uma vez que, após a indução do modelo, o seu desempenho seria avaliado pela análise do conjunto de treino suficientemente grande e representativo do universo. Todavia, esta abordagem não é viável, tendo em conta que os dados disponíveis raramente são suficientes, pelo que é necessário recorrer a mecanismos de generalização da estimativa do erro. Os métodos de generalização de estimativas estão intimamente relacionados com a fase de indução dos modelos. Tipicamente, do conjunto de dados inicial reserva-se uma amostra que é utilizada exclusivamente como conjunto de treino (i. e., só é usada na fase de indução). Os restantes dados são utilizados como conjunto de teste e empregues para avaliar o desempenho dos modelos induzidos. Os principais métodos estatísticos não paramétricos seguem essa filosofia: validação cruzada, métodos de Monte-Carlo e «boot-strapping». A principal distinção entre os referidos métodos está relacionada com o processo de amostragem dos exemplos [25].

A validade das abordagens de amostragem assenta na premissa de se trabalharem problemas de aprendizagem bem formados, ou seja [45]:

- na capacidade de indução dos modelos pelo algoritmo em uso, i. e., o modelo não apresenta tendências que inviabilizem a sua utilização no domínio em causa;
- na existência de uma solução;
- na existência de um modelo de indução estável, i. e., insensibilidade relativa do algoritmo de indução à presença de pequenas perturbações do conjunto de treino, (i. e., obtenção de modelos com pequena variância).

As premissas enunciadas estão, genericamente, asseguradas pela característica intrínseca dos modelos utilizados na aprendizagem automática.

Validação cruzada

O método de validação cruzada baseia-se no princípio da generalização de uma estimativa baseada na média dos resultados obtidos em diversas amostragens. Supondo que o *corpus* J é composto por n observações classificadas, obtém-se:

$$J = \{(x_i, y_i)\}, |J| = n \,. \qquad (12)$$

Os dados são divididos em dois conjuntos, o primeiro com n_D observações, que são utilizadas no conjunto de desenho *D,* e o segundo com $n_T = n - n_D$ observações, utilizadas

para o conjunto de teste *T* . A generalização da estimativa é determinada pela média dos valores de erro estimados a partir de todos os dados (validação cruzada completa) ou de um conjunto de divisões dos dados [46].

A validação cruzada completa é, habitualmente, apelidada de deixa-v-fora[17] e consome recursos computacionais usualmente incomportáveis, tendo em conta que existem $C_{n_d}^{n}$ diferentes maneiras de dividir o conjunto *J*. Claramente, a complexidade computacional torna impraticável esta aproximação, em especial com o aumento da dimensão do n_d , pelo que, na prática, à excepção dos casos em que o n_d é reduzido (i. e. $n_d \leq 2$), somente é utilizado um subconjunto das divisões.

Por esta razão, um dos casos mais estudados é precisamente aquele em que $n_d = 1$, situação apelidada por deixa-um-fora[18]. Neste caso o número de divisões possíveis dos dados é igual à sua dimensão, sendo a indução do modelo realizada n vezes e utilizando em cada caso, um *D* composto por todos os elementos menos um, que é usado como *T*.

O método deixa-um-fora, apesar de muito popular, e de ser equivalente a outros métodos, (tais como Akaike information Criterion (AIC), o C_p e o «bootstrap»), é assimptoticamente inconsistente, no sentido de não assegurar que a probabilidade de selecção do modelo com melhor estimativa converge para 1 quando $n \to \infty$ [47, 48]. Acresce o facto do método ter um desempenho elevado, na generalização de estimativas de funções-erro contínuas, tais como o erro quadrático médio, mas apresentar um desempenho inferior em funções de erro descontínuas, tais como o número de páginas classificadas erroneamente.

Uma solução para limitar o número possível de divisões dos dados passa pela adopção da restrição de utilização de módulos de dimensão fixa e igual, (ou equivalente), mutuamente exclusivos. O modelo é induzido *k* vezes, excluindo em cada caso, do conjunto *D* uma secção diferente, que é utilizada para o conjunto *T*. O número de possibilidades de divisão dos dados passa, então, a ser idêntico ao número de secções, realizando-se uma estimativa da validação cruzada completa, pela utilização de uma única divisão dos dados.

Neste caso, estamos na presença do método validação cruzada com k-subconjuntos[19], sendo a generalização da estimativa, E_{gen} , calculada da seguinte forma:

$$E_{gen} \approx \langle E_{val} \rangle = \frac{1}{k} \sum_{1}^{k} E_{val}(k) , \qquad (13)$$

[17] Deixa-v-fora – Tradução para «Leave-v-out»

[18] Deixa-um-fora - Tradução para «Leave-one-out»

[19] Usualmente apelidados de «*k-fold cross-validation*» ou «*stratified cross-validation*».

em que $E_{val}(k)$ representa o valor de desempenho calculado no decurso da K-ésina amostragem.

No caso particular do $k = n$ estamos em presença do método deixa-um-fora (validação cruzada com n-subconjuntos), um dos únicos casos viáveis de validação cruzada completa.

No caso de $k = 2$, o método de validação cruzada, assume a sua forma mais simples, apelidada de divisão da amostra[20] em que os dados são divididos uma única vez. Neste caso, não existe o «cruzamento», uma vez que os subconjuntos são utilizados alternadamente com *D* e *T*. A distinção entre a validação cruzada e a divisão da amostra é substancial, tendo em conta que, no método de divisão de amostra não existe cruzamento, o que diminui a robustez dos resultados. Esta é tanto mais importante quanto menor for a dimensão da amostra, uma vez que a representação do universo das observações e a relevância estatística dos dados, na maioria das vezes essencial para o bom funcionamento dos algoritmos de indução, ficam definitivamente comprometidas. Para além disso é um método, que usa os dados de forma ineficiente (visto não utilizar em nenhum caso os dados contidos no conjunto *T* na indução do modelo), resultando também numa generalização pessimista (estimador pessimista)[49].

A Figura 4 apresenta a divisão dos dados para o caso em que K tem valor cinco.

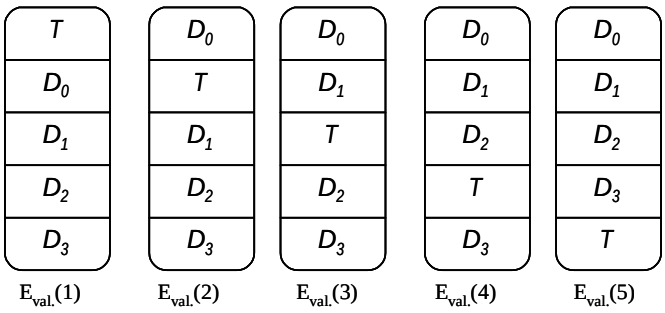


Figura 4 – Exemplo de uma segmentação dos dados para um k=5, sendo o conjunto de treino *D* a reunião dos subconjuntos D_i

Para a generalização de funções descontínuas, o método mais usual é a validação cruzada de k-subconjuntos com $k = 10$.

As técnicas apresentadas para a generalização de métricas resultam do cálculo médio dos valores obtidos em cada experiência. Em consequência, é comum as generalizações serem acompanhadas do correspondente desvio-padrão permitindo, assim, uma avaliação da dispersão dos valores em relação ao valor médio.

[20] Usualmente apelidados de «*slip-sample*» ou «*hold-out*».

Ao erro da aprendizagem, é necessário adicionar o erro introduzido pela generalização da estimativa, composto por três componentes aditivas:

- a variância do ruído (σ_{ξ}^{2}), que quantifica o erro residual e é, na realidade, um limite mínimo expectável. Reflecte a variabilidade das futuras observações sendo independente do algoritmo de aprendizagem;
- a variância das estimativas, que quantifica a dependência dos classificadores em função do conjunto de desenho. Quanto maior for a sensibilidade do algoritmo às variações do conjunto ,maior é o seu valor;
- o desvio quadrático das estimativas, que quantifica o erro na selecção do modelo pelo cálculo da diferença quadrática entre a média dos valores da função-objectivo e a média dos valores obtidos pelo algoritmo. Quanto menor for o desvio da função-objectivo, menor é o seu valor.

Por outras palavras, o erro da aprendizagem pode ser representado da seguinte forma:

$$Erro = \langle E_{gen} \rangle = Desvio^{2} + variância + \sigma_{\xi}^{2}\text{, [48].} \qquad (14)$$

No caso específico da validação cruzada com k-subconjuntos, a σ_{ξ}^{2} é idêntica para todas as divisões, e o desvio situa-se tão mais próximo de zero quanto melhor forem as divisões, pelo que o erro, no caso das boas divisões, depende, fortemente, da variância das estimativas.

2.2.4.2 Representação dos documentos

A representação dos documentos empregue na Web não é adequada para a utilização em técnicas de recuperação de informação, dada a sua natureza ser heterogénea (múltiplos formatos) e por conter excessiva informação irrelevante.

Tradicionalmente, a representação utilizada em recuperação de informação em texto, recorre a modelos vectoriais definidos pelas palavras contidas no texto. O vector pode ser de valores binários, correspondendo, à existência ou não da palavra; ou então, de valores inteiros ou reais, (determinados por uma equação), correspondendo neste caso, por exemplo, à ocorrência e localização da palavra na página [50].

Consequentemente, é possível guardar a estrutura do texto, a ordem das palavras, a vizinhança, etc. Todavia, considera-se necessário ponderar quais as vantagens desta aproximação e os ganhos efectivos conseguidos numa óptica de aprendizagem em texto.

Os casos descritos, em que se mantém a estrutura dos documentos ou, somente, a ordem das palavras, estão, usualmente, associados a análises semânticas ou ao estudo dos textos não por palavras, mas por conjunto de palavras (anagramas). A agregação de palavras é normalmente realizada entre palavras contíguas, resultando na criação de estruturas n-gramas que passam a representar uma nova característica (e. g., «Professor Doutor» é

um 2-grama, «World Wide Web» é um 3-grama). Todavia, a sua aplicação não é consensual, por exemplo em [50] defende-se a sua aplicação especialmente em documentos de elevada dimensão, enquanto que em [51] são registadas melhorias de desempenho para documentos de reduzida dimensão. Especificamente nas aplicações para a Web, a maioria das aproximações continua a recorrer à utilização de um saco de palavras, que identifica a existência ou a ocorrência da palavra no texto em análise. Porém, existem estudos recentes que indicam como promissora a utilização de informação adicional, tal como as instruções do hipertexto, as articulações, o grafo das páginas e mesmo o endereço das próprias páginas [52, 53].

2.2.4.3 Pré-processamento dos dados

A dimensionalidade dos espaços de pesquisa condiciona a eficácia dos métodos, visto que o seu aumento obriga ao acréscimo exponencial de observações para garantir representatividade. Investigadores e utilizadores concluíram que o processamento prévio dos dados é essencial para a utilização eficaz de algoritmos de exploração de dados, aprendizagem e visualização. A aprendizagem supervisionada aborda esta questão, essencialmente pela redução do espaço e através da utilização de técnicas de selecção das características mais eficazes, i. e., não correlacionadas e que não introduzem ruído.

Apesar de existirem extensos estudos e vasta literatura sobre o processamento dos dados, a tarefa de pré-processamento é por vezes descorada. Todavia, o estudo teórico e a prática demonstram que muitos métodos têm desempenhos pouco escaláveis com o aumento de características que, muitas vezes, são irrelevantes e redundantes [18]. Todas as evidências indicam a necessidade de utilização de algoritmos para ultrapassar estas dificuldades.

Como técnicas mais frequentes para o pré-processamento dos dados destacam-se a transformação e a selecção de características, respectivamente, com o objectivo de preservar as características topológicas da representação, ou aumentar a eficiência e eficácia das técnicas de exploração de dados [54].

A transformação de características é o processo de criar novas características, obtidas por construção ou extracção, igualmente apelidadas de descoberta de características[21].

Assumindo um espaço inicial de características $\Delta = \{A_1,...,A_{|\Delta|}\}$ podem-se definir os seguintes processos de transformação:

- **Construção de características** é o processo de criar um novo conjunto de características, ocultas na informação inicial, através da descoberta de relacionamentos entre as características iniciais, aumentando o espaço de

[21] Extracção de características –Tradução adoptada de *Feature extraction*.

características. Em consequência, o espaço de características pode ser aumentado em K novos elementos, i. e., A_i novos elementos $(n < i \leq n + k)$. A título de exemplo, num espaço multidimensional, o volume hiperbólico pode ser uma nova característica;

- **Extracção de características** é o processo de criar um novo espaço de características pela utilização de uma função de transformação. Por outras palavras, obtêm-se $B = \{B_1,\ldots,B_m\}$, $B_i = F_i(A_1,\ldots,A_n)$, em que F_i é a função de transformação da i-ésia característica.

Nas aplicações reais, a dimensão do espaço de características é muito vasto, todavia não é invulgar que a identificação da variável de classe dependa de um número comparativamente muito reduzido de características. Nestes casos, a utilização de todas as características não contribui para a eficácia dos métodos de aprendizagem; pelo contrário dificulta, por não conterem informação ou contribuírem pouco para a discriminação dos dados.

A selecção de características é um processo diferente e consiste em eliminar um subconjunto de características que são irrelevantes, não contribuindo para o bom desempenho dos algoritmos de exploração de dados. Não existe, assim, a criação de novas características mas sim a eliminação e consequente redução da dimensão do espaço de características original, por vezes interpretado como uma compactação horizontal dos dados [55]. Normalmente, o vector de características que define uma observação apresenta uma elevada cardinalidade, o que impossibilita a utilização de todas as características no processo de classificação. A impossibilidade da utilização de vectores de características de elevada cardinalidade, advém não só da dificuldade de armazenamento e manipulação computacional dos dados mas, igualmente, da dispersão das observações pelo espaço multidimensional que inviabiliza a identificação de agregados.

O processo de selecção de vectores de características é uma área de investigação actual [56], está intimamente relacionado com a natureza do problema em estudo e obriga à interpretação dos fenómenos envolvidos [57].

É essencial realizar a redução do espaço de características desnecessárias, com vista a reduzir o tempo de treino dos algoritmos, aumentar a qualidade das regras induzidas e aumentar a eficiência e eficácia dos algoritmos de classificação.

A transformação e a selecção de características não são necessariamente dois assuntos independentes, por vezes é imperativa a redução do espaço de características para evitar redundância e elementos irrelevantes. Todavia, a natureza do problema obriga igualmente à sua transformação com vista a enriquecer a linguagem, ultrapassando assim as insuficiências iniciais. A utilização das duas técnicas em conjunto é muito comum.

Selecção de características para catalogação de texto

A catalogação de texto é um exemplo onde o espaço de características tem uma dimensão extremamente elevada, resultante da riqueza lexical da linguagem natural. A aproximação imediata entre a construção do espaço de características e os textos em linguagem natural consiste na utilização das palavras ou frases únicas o que conduz, inevitavelmente, a dezenas ou centenas de milhar de características [58]. Esta cardinalidade é excessiva para a maioria, se não para a totalidade, dos algoritmos de aprendizagem, pelo que é imperativo identificar um conjunto de técnicas que reduzam drasticamente o número de características, sem diminuir a eficácia da catalogação. Esta redução deve ser conseguida, obviamente, de forma automática, i. e., sem o recurso à intervenção manual.

A elevada dimensionalidade do vector que representa o texto, não só levanta problemas de tratamento computacional (um problema intratável), como, igualmente, inviabiliza a utilização de qualquer técnica de indução de classificadores, devido ao ruído introduzido por variáveis indesejáveis e pela dispersão dos exemplos no espaço de variáveis.

Existem diversos métodos para identificar as características que garante uma boa representação do documento. As estratégias seguidas podem ser cegas, sem ter em consideração as especificidades do caso em estudo (filtros), ou de considerar os dados em análise e seleccionar as palavras segundo critérios de ordenação.

O objectivo é reduzir o conjunto total de termos a considerar, viabilizando uma análise mais detalhada e computacionalmente mais exigente. A Figura 5, ilustra este objectivo identificando como relevantes as características que estão compreendidas entre os limites inferior e superior de frequência. Por outras palavras, são consideradas igualmente irrelevantes, as palavras raras e as palavras de elevada frequência.

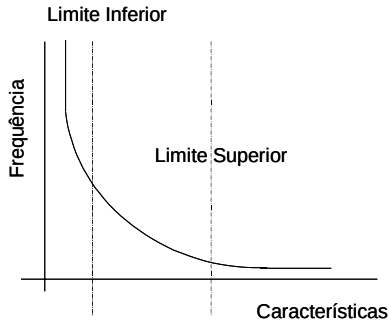


Figura 5 – Distribuição das características por frequência

O método Frequência no Documento (FD) baseia-se na definição de limites mínimos de frequência de ocorrência de palavras no total de documentos, abaixo dos quais as palavras não são consideradas como boas candidatas. A premissa-base é a de que termos pouco frequentes não são informativos e/ou não são influentes no desempenho global [58]. Esta técnica cega, que elimina todas as palavras abaixo de um valor predefinido, é muito

eficaz. Contudo, a definição do limite é de extrema relevância para evitar cortes excessivamente agressivos.

O processamento prévio dos documentos é, usualmente, iniciado com o recurso a técnicas cegas de redução de características: *i)* listas de paragem e; *ii)* redução a radicais.

As **listas de paragem** permitem fazer um processamento prévio para eliminar, do vector de características, as palavras que constem da lista. Estas listas contêm, para cada língua, um conjunto de palavras não discriminantes, i. e., a sua presença não auxilia à discriminação dos conteúdos apresentados (tais como os artigos, os pronomes, etc.). Estas listas, de dimensão inferior a 1000 palavras, chegam a reduzir entre 30 por cento a 50 por cento a dimensão dos documentos. A eficiência e eficácia deste método conduziu à sua generalização.

Igualmente dependente da língua, outra aproximação possível baseia-se na **redução das palavras ao seu radical**, por exemplo eliminando todos os plurais, ou passando os verbos para a sua forma infinitiva. Naturalmente que a redução ao radical contribui para a diminuição do total de palavras, uma vez que as variações são eliminadas e representadas por um único termo. Todavia, esta técnica não é simples e obriga à criação de *thesaurus* e regras extremamente complexas para minimizar os erros potenciais. A sua aplicação é usual na área do processamento de linguagem natural.

Finalizado o processamento prévio, são utilizadas técnicas que permitem ordenar as características sobrantes com o objectivo de poder seleccionar, exclusivamente, as melhores. Por outras palavras, permite identificar o conjunto (S) que inclui as características com maior grau de discriminação da classe (C), i. e., sendo

$$\Delta = \{A_1, \cdots, A_{|\Delta|}\}, \tag{15}$$

um conjunto de características, e

$$e : A_1 \times A_2 \times \cdots \times A_{|\Delta|} \rightarrow C , \tag{16}$$

uma relação em que, cada A_i , toma um conjunto discreto e finito de valores

$$\{a_{i,1}, \cdots, a_{i,|A_i|}\}, \tag{17}$$

e

$$C = \{c_1, \cdots c_{|C|}\} \tag{18}$$

determina o conjunto de características

$$S = \{S_1, ..., S_k\} \subseteq \Delta \tag{19}$$

que melhor discriminam C .

O método da Informação Mútua (IM) é um critério comum da teoria de informação, e é utilizado, essencialmente, em modelos de linguagem estatística de associação de palavras, procurando determinar o grau de relação entre cada termo e cada uma das classes, atribuindo o valor mais elevado aos termos mais discriminantes.

O conjunto S , descrito na equação (19), é, neste caso, determinado por

$$I(S_1;C) \geq I(S_2;C) \geq \cdots \geq I(S_K;C) \geq I(Z;C) \quad para\ todo\ o\ Z \in \Delta - S \text{ ,(20)}$$

considerando que

$$I(S_i;C) = H(C) - H(C \mid S_i). \tag{21}$$

Como em $H(C)$ é uma constante para todos os S_i , a equação (20) pode reescrever-se como

$$H(C \mid S_1) \leq H(C \mid S_2) \leq \cdots \leq H(C \mid S_K) \leq H(C \mid Z) \quad para\ todo\ o\ Z \in \Delta - S \text{ , (22)}$$

sendo por definição,

$$H(C \mid S_i) = -\sum_{k=1}^{|C|}\sum_{j=1}^{|S_i|} p(c_k, s_{i,j}) log\, p(c_k \mid s_{i,j}). \tag{23}$$

A principal fragilidade deste método é a influência das probabilidades marginais dos termos, como se denota na forma equivalente

$$I(S_i, C) = log\, P(S_i \mid C) - log\, P(S_i) \tag{24}$$

Entre termos com a mesma probabilidade condicional, os mais raros são privilegiados, o que enfraquece a validade da ordenação para termos com frequências distintas.

O método do Qui-quadrado (χ^2) procura avaliar o grau de independência entre os termos e as classes, sendo a sua utilização comparável à aplicação estatística baseada no método da distribuição de χ^2 com um grau de liberdade, para avaliar extremos.

O conjunto S , descrito na equação (19) é, neste caso, determinado por

$$\chi^2(S_1) \geq \chi^2(S_2) \geq \cdots \geq \chi^2(S_K) \geq \chi^2(Z)\ para\ todo\ o\ Z \in \Delta - S \tag{25}$$

Utilizando uma tabela de contingência para dois termos, (*A* – termo e classe existem; *B* - termo existe e classe não existe; *C* – não existe termo e existe classe; *D* – não existe nem termo nem classe; *N* – total de observações), obtemos a seguinte definição:

$$\chi^2(Y_i, c_j) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{26}$$

No caso do termo e da classe serem independentes, a estatística χ^2 tem o valor natural zero, aumentando o seu valor com o grau de relação. A combinação dos valores obtidos para cada uma das classes pode ser determinada pela média, valor mínimo ou valor máximo, respectivamente

$$\chi^2_{média}(Y_i) = \sum_{i=1}^{|C|} P_r(c_i)\,\chi^2(Y_i, c_i) \tag{27}$$

$$\chi^2_{min}(Y_i) = \min_{i=1}^{m} \left\{ \chi^2(Y_i, c_i) \right\} \tag{28}$$

$$\chi^2_{max}(Y_i) = \max_{i=1}^{m} \left\{ \chi^2(Y_i, c_i) \right\} \tag{29}$$

A principal fragilidade deste método reside no facto de não ser fiável para termos de baixa frequência, deixando de ser comparável ao método da distribuição de χ^2 [59].

O método ReliefF (RF) estima a importância dos termos, assumindo que os mais úteis apresentam valores distintos para classes diferentes e valores semelhantes para classes iguais. Inicialmente propostos por Kira e Rendell [60], como Relief para problemas com duas classes, foi desenvolvido por Kononenko [61] para lidar com ruído e com classes múltiplas. O método proposto selecciona aleatoriamente uma observação e procura os k-vizinhos mais próximos de todas as classes. Os valores dos atributos são comparados por classes, com o valor médio dos vectores vizinhos (o que atenua o ruído), sendo valorizados os atributos que apresentem valores semelhantes para classes iguais e distintos para classes diferentes. Este processo é repetido *m* vezes. Quanto maior for o número de experiências *m*, maior a fiabilidade. Os valores sugeridos por Kononenko são um m = 250 e um k = 10 [62, 63].

O método da Força do Termo (FT) visa estimar a importância dos termos, tendo por base o facto de ser comum a sua existência em documentos semelhantes. O conjunto de treino é utilizado para gerar pares de documentos considerados semelhantes (o que habitualmente é determinado por um valor acima de um limite do cosseno dos vectores de documentos). A força do termo é, então, determinada pela estimativa da probabilidade condicional do termo ocorrer em dois documentos semelhantes tendo ocorrido no primeiro. Por outras palavras, sendo d_1 e d_2 documentos assumidos como semelhantes e *Y* um termo, a força do termo é

$$FT(Y) = P(Y \in d_1 \,|\, Y \in d_2) \tag{30}$$

Este método, baseado na agregação de documentos, não utiliza informação sobre as classes, assemelhando-se, nesta propriedade, aos métodos «cegos». A sua premissa-base é a de que documentos semelhantes possuem termos comuns, e que a relevância informativa do termo, está relacionada com a sua presença no maior número de

documentos semelhantes. O factor crítico neste método é a determinação da semelhança para formação dos pares de documentos, sendo, usualmente, determinado de forma experimental.

Um exemplo dum método de aprendizagem não supervisionada é a Análise das Componentes Principais (ACP), uma técnica estatística que reduz a dimensionalidade pela representação dos dados, num novo subespaço, obtido pela transformação do espaço original de atributos. A transformação do espaço é realizada pelo cálculo da matriz de co-variação dos atributos originais e pela subsequente extracção dos vectores próprios com maior variância, nos dados originais. Os vectores próprios, (componente principal) definem uma transformação linear do espaço de atributos original, para um novo espaço, em que os atributos não são correlacionados.

Alguns métodos procuram utilizar subconjuntos de atributos para determinar a utilidade dos termos. O método CFS (Correlation-based Feature Selection) procura determinar o valor dos atributos avaliando a capacidade de previsão das classes, tendo em atenção a correlação entre termos. A métrica de avaliação de mérito utilizada é

$$Mérito_S = \frac{K\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \qquad (31)$$

onde $Mérito_S$ avalia o mérito de um subconjunto contendo *K* características, sendo $\overline{r_{cf}}$ o valor médio da correlação característica da classe, e $\overline{r_{ff}}$ o valor médio da inter-relação das características. O numerador pode ser encarado como uma medida de avaliação da capacidade preditiva do grupo, e o denominador, como medida da correlação no grupo. As aproximações mais comuns à determinação da correlação das características, utilizam a incerteza simétrica, para determinar o valor da inter-relação entre X e Y, por outras palavras,

$$Incerteza(X,Y) = 2 \times \left[\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}\right]. \qquad (32)$$

Outros métodos, procuram avaliar a consistência dos subconjuntos, por determinação de combinações de atributos que subdividam os exemplos por classes, privilegiando os subconjuntos de menor dimensão. Uma proposta para avaliação da consistência dos subconjuntos, da autoria de Liu e Setiono [64], consiste em

$$Consistência_S = 1 - \frac{\sum_{i=0}^{J} |A_i| - |C_i|}{N}, \qquad (33)$$

onde *S* é um conjunto de atributos, *J* o número de distintas combinações dos valores dos atributos de *S*, $|A_i|$ o número de ocorrências da i-ésima combinação de atributos, $|C_i|$ o número de ocorrências da classe maioritária na i-ésima combinação de atributos, e *N* o

número total de observações. A aplicação deste método obriga à identificação de subconjuntos de atributos candidatos, através de uma técnica de ordenação de atributos.

Uma análise crítica aos métodos apresentados, permite afirmar, de forma genérica, que os estudos demonstram que a redução das características permite obter bons resultados, contudo não é possível identificar um método que seja o melhor, em todas as situações.

A selecção de características reduz o espaço de pesquisa de hipóteses e permite, aos algoritmos de indução de estimadores, maior eficácia, rapidez, e melhoria de desempenho dos estimadores induzidos, em especial se os algoritmos de indução não eliminarem características, como é o caso do método vizinhos mais próximos. Genericamente, a selecção de características melhora a compactação, e a facilidade de interpretação.

A aplicação de métodos simples, em especial FD (Frequência do Documento) produz resultados surpreendentes, sendo comum o relato da semelhança entre os métodos FD, IM e χ^2, pelo que a aplicação do FD deve ser tida em conta, em especial nos casos em que o custo computacional é elevado. Todavia, a aplicação dos métodos que fazem uso da informação de associação termo-classe são, usualmente, apontados como os mais eficazes [58].

No caso da existência de fortes inter-relações entre atributos, o ReliefF e o CFS são defendidos como a escolha mais acertada, em especial o CFS pela capacidade de escolha de poucas características, pela rapidez e pela compactação dos modelos baseados em árvore [62].

Os resultados obtidos com o método FT (Força do Termo) são comparáveis aos restantes métodos, para reduções de características até 50 por cento. Acima deste valor o desempenho fica seriamente comprometido. A aplicação do método da IM e χ^2 deve ser analisada cuidadosamente. No caso da IM, devido ao seu desvio que favorece termos raros e, no caso do χ^2, por ser posta em causa a sua validade para termos de baixa frequência.

2.2.4.4 Classificadores

O projecto de um classificador deve ser, tanto quanto possível, independente do problema específico em análise. Genericamente, um classificador é um interpretador que aplica $e : X \rightarrow C$, i. e., que associa a cada observação $x \in X$ uma classe $c_i \in C$. Por outras palavras, um classificador fica definido se, para cada observação de entrada, identificar a classe que lhe está associada. Noutra interpretação, um classificador realiza a partição do conjunto *X* em *n* subconjuntos disjuntos $R_1,\ldots,R_n$, em que R_1 é um subconjunto de *S* contendo, exclusivamente, observações da classe c_i, $R_i = \{x \in S : e(x) = c_i\}$. O projecto de um classificador baseia-se na procura das melhores regiões de decisão [56].

Os últimos anos têm sido prolíferos na utilização de novas soluções com vista à catalogação de textos que vão desde os métodos de classificação estatísticos até à utilização de técnicas de aprendizagem automática, incluindo modelos de regressão multivariáveis, os k-vizinhos, aproximações probabilísticas de Bayes, Árvores de Decisão, redes neuronais, *Support Vector Machines (SVM)*, etc.

Os métodos de indução de classificadores podem ser agrupados tendo em consideração algumas das suas propriedades em: *i)* métodos baseados em modelos e *ii)* métodos baseados em instância.

Os métodos baseados em modelos (*eager learning*) procuram inferir um modelo que permita capturar o «sentido» das observações com base no registo de um histórico disponível, assumindo, assim, a existência de uma função tal que $C_i = f(A_1, A_2, \ldots, A_k)$. Pretende-se inferir essa função de forma a ser possível estimar a classificação de novos casos. O método Naive Bayes e as Árvores de Decisão são exemplos desta aproximação.

A aprendizagem baseada em instâncias (*lazy learning*), pelo contrário, não procura identificar um modelo, nem assume a existência de uma função, mas limita-se a armazenar o histórico adiando o processamento para o momento da classificação das observações. Nessa altura identifica quais os exemplos mais semelhantes e estima a classificação baseada nas suas classificações. São exemplos os métodos dos vizinhos mais próximos, regressões localmente ponderadas e métodos de raciocínio baseado em casos [20]. As principais diferenças entre os métodos situam-se no processo de representação das instâncias de treino, no cálculo das distâncias entre instâncias e no processo de avaliação do valor da função de classificação.

Os modelos que utilizam funções que calculam probabilidades condicionais são apelidados de modelos baseados em funções-discriminantes. Dependendo dos pressupostos, obtêm-se funções distintas e, consequentemente, classificadores igualmente distintos. São exemplos os classificadores Naive Bayes, os classificadores discriminantes, e os classificadores discriminante lineares, quadráticos, ou logísticos [65].

A análise comparativa dos algoritmos de indução é complexa, sendo poucos os estudos comparativos de raiz, por contraste com os estudos de avaliação do desempenho dos algoritmos de indução de classificadores. Este cenário permite a obtenção de resultados inconsistentes, e de difícil confrontação. A título de exemplo, sobre o *corpus* Reuters-21578[22], Joachims reporta para as SVM um desempenho superior aos métodos tradicionais, (e. g., *Naive Bayesian* e as Árvores de Decisão) [66], enquanto Apte, Damerau,

[22] http://www.daviddlewis.com/resources/testcollections/reuters21578 compilado e classificado pelo Carnegie Group, Inc, e pela Reuters, Ltd, no decurso do desenvolvimento do sistema CONSTRUE.

and Weiss relatam ainda melhores resultados utilizando «boosted decision trees» [67]. Todavia, Yang identificou desempenhos semelhantes no caso dos SVM, k-vizinhos, e linear «least-square fit», tendo obtido resultados inferiores no caso do classificador Naive Bayesian e redes neuronais [44]. Em paralelo, Pazzani e Bilsus concluem que o classificador Naive Bayesian, e as redes neuronais apresentam excelentes desempenhos, defendendo, inclusive, ser mais promissor procurar melhorar a representação dos documentos do que procurar novos algoritmos de indução de classificadores [68, 69].

A realização da análise, a partir de estudos isolados, é complexa, tendo em conta que os casos de estudo não são idênticos e o desempenho dos algoritmos está intimamente relacionado com os dados e com os cuidados metodológicos (incluindo a determinação das medidas de desempenho). É necessário ter sempre presente que o erro na aprendizagem tem duas origens essenciais: *i)* sistemática: que decorre da representação, da estratégia de pesquisa e ainda do erro intrínseco ao algoritmo; *ii)* e a dependente do conjunto de desenho.

Igualmente relevante, é a usual inexistência de robustez para distribuições enviesadas de categorias, em especial devido à verificação experimental de que a maioria dos dados não são normalmente distribuídos.

A inexistência de evidência da superioridade de um algoritmo de indução de classificadores, obriga a considerar, sobre reserva, generalizações abusivas de resultados experimentais obtidos para casos concretos. Os bons resultados obtidos por um algoritmo estão, normalmente, associados ao domínio de aplicação em estudo, ao conjunto de exemplos utilizado, etc.

Os métodos analisados a seguir são de natureza distinta: *i)* Vizinhos mais próximos; *ii)* Árvores de Decisão; e *iii)* Naive Bayes.

2.2.4.5 Vizinhos mais próximos (k-vizinhos)

O método dos vizinhos mais próximos, conhecido igualmente por k-vizinhos, proposto nos anos sessenta, é um método de aprendizagem baseado em instâncias. Tendo em conta que o vector de representação do documento está predefinido, podemos afirmar que é um método em que não existe qualquer processo de aprendizagem, estando reduzido à memorização das observações presentes no *corpus*. O processo de classificação desenrola-se pela identificação do conjunto de observações que estão a menor distância da observação a classificar, sendo atribuída a estimativa de classificação à classe predominante. Os exemplos são encarados como pontos no espaço Euclidiano discreto ou real e, dependendo da utilização de todo o conjunto de exemplos ou dos vizinhos mais próximos, realiza-se a distinção entre uma aproximação global ou local. Apesar de não existir um modelo no método dos k-vizinhos, é possível imaginar uma representação para

k=1 através da construção de superfícies de decisão e poliedros que envolvem cada um dos exemplos (assume-se que o conjunto de exemplos está estável). O conjunto de poliedros resultante, corresponde à construção de *Voronoi Tessellation*, permitindo assim identificar a área de «influência» de cada exemplo. A Figura 6 apresenta um diagrama de Voronoi.

Figura 6 – Diagrama de Voronoi apresentando os poliedros que definem a área de «influência» de cada observação para um k=1. (O diagrama foi construído com o recurso à ferramenta disponibilizada em http://www.cs.cornell.edu/Info/People/chew/Delaunay.html)

Neste caso, a classificação atribuída à observação é definida pelo exemplo que está dentro do poliedro. A principal vantagem deste método relaciona-se com a criação de uma superfície de decisão que se adapte à forma de distribuição dos dados de treino de forma detalhada e à não assunção de um modelo prévio. A principal desvantagem está relacionada com a natureza intrínseca dos métodos de instância; isto é, o processo de estimativa é muito demorado, uma vez que obriga ao cálculo da distância da observação a todos os elementos do conjunto de treino. O desempenho do método depende da definição de semelhança que, normalmente, é definida à custa de uma métrica de distância num espaço multidimensional, sendo mais semelhantes entre si as observações que estiverem a menor distância. As técnicas mais comuns que permitem corrigir os problemas identificados são a pesagem dos vizinhos, (em função da distância até à observação em avaliação), a eliminação de características irrelevantes e a distorção dos eixos de representação. A definição genérica de distância é

$$D(x_i, y_j) = \left(\sum_{v=1}^{k} w_v d_v \left(x_i, y_j \right)^r \right)^{1/r}, \qquad (34)$$

em que $d_v(x_i, y_i)$ é a distância entre valores do atributo v, e w_j o peso relativo atribuído a cada característica.

O bom funcionamento deste algoritmo está condicionado à decisão acertada sobre o número de exemplos a incluir na vizinhança *K*, uma vez que tem influência directa, não só no desempenho do algoritmo como na sua eficácia. Quanto maior for o número de vizinhos, mais lento é o desempenho do algoritmo. Todavia, o aumento do valor do *K* permite eliminar exemplos-excepção, (e. g., uma observação isolada num vasto conjunto de classificação homogénea distinta). Com o aumento do número de vizinhos as fronteiras de decisão

correspondem a agregados locais, onde uma classe domina. A influência do *K* é tanto mais relevante quanto maior for a intersecção das observações, ou seja, quanto menor for a definição clara de agrupamentos de observações da mesma classe e nos casos-limite, como sejam observações isoladas e nas fronteiras dos agregados.

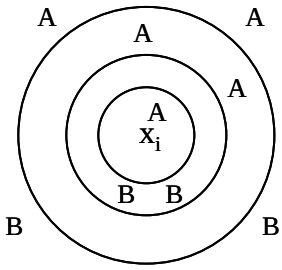


Figura 7 – Exemplo da influência do número de vizinhos para a classificação estimada para o vector x_i

A Figura 7 apresenta um exemplo ilustrativo da relevância e da influência para a classificação final da escolha do valor de *K*, uma vez que, para um *K = 1* a observação é classificada como pertencendo à classe *A*, com *K = 3* a *B* e com um *K = 5* de novo como a *A*. O erro cometido sobre o conjunto de desenho aumenta com o valor de *K*. Naturalmente que, com um *K = 1*, o erro sobre o conjunto de treino é igual a zero, aumentando progressivamente com o incremento do *K*.

Ponderar a distância aos vizinhos?

Considerar indiscriminadamente a influência de todos os exemplos identificados, como os vizinhos mais próximos, pode induzir em erro. Um processo de contornar esta limitação do método dos vizinhos mais próximos, consiste na discriminação positiva dos exemplos que estão mais próximos. O algoritmo pode ser descrito da seguinte forma:

K-NearestNeighborWheigh(x, D)
Sejam $\{z_1, \ldots, z_K\} \in D$ *, os k-vizinhos mais próximos de x.*

Retorna $$\hat{c}(x) = \arg\max_{v \in C} \sum_{i=1}^{K} \frac{1}{S(x, \boldsymbol{z}_i)} \delta(v, c(\boldsymbol{z}_i))$$

Tendo em conta que o método pondera a distância dos exemplos ao caso em análise, passa a ser possível eliminar a definição da vizinhança, i. e., podem assim ser considerados na determinação da classificação a estimar, todos os exemplos presentes no conjunto de treino, uma vez que a influência dos exemplos é ponderada em função da distância à observação em análise. O método passa, então, a ser considerado como um método global.

Considerações finais

O método dos vizinhos mais próximos é pouco estável, sendo fortemente dependente da posição relativa dos exemplos presentes, no conjunto de desenho. A posição dos vizinhos presentes num subespaço é determinante e pequenas alterações podem conduzir a

alterações significativas nos resultados finais, o que conduz à existência de uma grande variação de desempenho, dependendo do conjunto de desenho utilizado [70].

A existência de um vasto conjunto de desenho pode parecer suficiente para assegurar a aplicação teórica óptima do método dos k-vizinhos, por viabilizar um amplo conjunto de vizinhos próximos da observação a classificar, permitindo determinar correctamente a classificação a atribuir. Todavia, a intuição é traída pelo aumento da dimensionalidade, que conduz ao crescimento exponencial do hipercubo, facto que foi apelidado pela maldição da dimensionalidade [71]. Tendo em conta que o método dos vizinhos mais próximos pode ser visto como um processo de mapeamento do conjunto de entrada para o conjunto de saída, fazer a cobertura do espaço de entrada ocupa recursos proporcionais à dimensão do hipercubo que descreve o espaço de representação. A manutenção da distância média entre exemplos obriga ao seu crescimento exponencial em função do acréscimo da dimensionalidade. Partindo do princípio que o número de exemplos é estável, ao aumento do hipercubo corresponde: *i)* a dispersão dos exemplos; *ii)* a representação de zonas de espaço irrelevantes, tendo em conta que estão vazias. Concluindo, aumentar o número de exemplos não contribui, necessariamente, para preencher as porções vazias de hiperespaço, que são, muito provavelmente, definidos por características irrelevantes e que não contribuem para a melhor definição do problema. A solução nestes passa pela redução da dimensionalidade com a realização de uma melhor escolha das características que fazem parte do vector de representação.

2.2.4.6 Árvores de Decisão

Os algoritmos mais conhecidos para a indução de árvores de decisão, têm por base o ID3 e o C4.5. Formalmente, uma árvore de decisão é um grafo acíclico, em que cada nó, ou constitui um nó de decisão, com dois ou mais sucessores, ou é um nó folha. O nó de decisão possui um método de selecção baseado nos valores de um atributo, que permite navegar na árvore até se atingir um nó folha, ao qual está atribuída uma classificação.

Genericamente, o modelo é induzido por aproximação à função-objectivo, utilizando, para tal, funções em formato de árvore de decisão. Este método pode ser considerado como um método de pesquisa no espaço de hipóteses de todas as árvores passíveis de serem construídas com o conjunto de atributos. A estratégia seguida por esta categoria de métodos baseia-se no princípio da divisão e conquista, caracterizando-se pela abordagem recursiva de um problema complexo através da sua divisão em problemas mais simples.

A eficácia do algoritmo está fortemente relacionada com o processo de selecção dos atributos, sendo diversos os métodos propostos com vista à resolução deste problema. De uma forma geral, há concordância nos casos-limite: inutilidade de atributos que mantêm as proporções dos exemplos por classe, e relevância dos atributos que permitem a criação de

partições, em que apenas estão presentes exemplos de uma classe. A diferença reside nos casos intermédios. Os algoritmos mais utilizados são o ID3 [72] e C4.5 [73] que implementam métodos de pesquisa sobre todo o espaço. O ID3 e o C4.5 utilizam o conceito de entropia, que enfatiza a selecção dos atributos, baseada na pureza das partições [65].

O processo de pesquisa das hipóteses materializa-se a partir de árvores simples para complexas, utilizando uma estratégia trepa-colina[23] no espaço de hipóteses. A estratégia trepa-colina, aliada à inexistência de processos de «backtracking», conduz à identificação de soluções óptimas locais, mas compromete a determinação de soluções óptimas globais. Os métodos são particularmente robustos no que respeita à existência de ruído no conjunto de desenho, mas apresentam um desvio indutivo, para árvores de pequena dimensão e para árvores que possuam os atributos de maior ganho de informação, perto da raiz.

As ideias-base, que suportam o algoritmo ID3, são:

- a árvore de decisão – Numa árvore de decisão, a cada nó corresponde um atributo e, a cada arco, um valor possível do atributo. As folhas da árvore especificam a classificação a atribuir a uma observação que seja descrita pelo percurso efectuado até si, desde a raiz da árvore;
- a selecção de atributos – A cada nó da árvore de decisão deve ser associado o atributo mais discriminante dentre os que não foram seleccionados para o percurso;
- a medida Entropia, introduzida por Claude Shannon [74], é a medida utilizada para efectuar a selecção dos atributos mais discriminantes.

O C4.5 extende as capacidades do ID3, entre outras, permitindo:

- a manipulação de atributos sem valores, permitindo: *i)* na construção da árvore, o cálculo do ganho da informação, utilizando somente as observações que possuem valores, ou pela atribuição do valor mais comum; e *ii)* no processo de decisão pela associação de probabilidades às estimativas possíveis;
- a manipulação de atributos contínuos, pela partição do atributo, num conjunto discreto de intervalos;
- a ponderação do custo do atributo, através da normalização do valor do ganho de informação, pelo valor estimado para o custo do atributo;
- a realização da poda das árvores de decisão.

Nos dois casos, a determinação do ganho de informação é, assim, fundamental para realizar a selecção dos atributos. Para tal, é necessário assumir que as N observações do conjunto *D* são equiprováveis, logo que a probabilidade de ocorrência de cada uma é de $p_m = 1/N$,

[23] Trepa-colina – Tradução de «hill-climbing»

sendo a informação transmitida por cada observação de $-log_2(p) = log_2(n)$, (e. g., no caso de existirem 8 observações, $log_2(8) = 3$). Por outras palavras são necessários três *bits* para identificar cada observação. Genericamente, sendo $P = (p_1, p_2, ..., p_N)$ uma distribuição probabilística, então, a transmissão de informação associada a esta distribuição, apelidada de «Entropia de P» é descrita por:

$$Info(D) = E(P) = -p_1 log_2(p_1) - p_2 log_2(p_2) - ... - p_N log_2(p_N)$$ [24] (35)

O particionamento do conjunto de exemplos *D* em k classes, induz a que informação necessária para identificar um elemento de *D* seja $Info(D) = E(P)$, em que P é a distribuição de probabilidades da partição $(C_1, C_2, ...C_N)$.

Todavia, o cálculo da informação necessária para identificar um exemplo, (caso o conjunto *D* tenha sido particionado em R subconjuntos, pela utilização do atributo A_i), passa a ser uma média ponderada do valor necessário para cada partição, por outras palavras:

$$Info(A_i, D) = \sum_{i=1}^{R} \frac{|D_i|}{|D|} Info(D_i) .$$ (36)

A diferença entre a equação (35) e a equação (36) representa o ganho de informação obtido pelo conhecimento do atributo A_i, i. e., o facto de se conhecer o valor do atributo altera o valor da Entropia, conduzindo a um novo estado de conhecimento sobre o exemplo em análise, sendo assim:

$$GI(A_i, T) = E(T) - E(A_i, T)$$ (37)

O ganho de informação, associado a cada atributo, pode ser utilizado na construção de árvores de decisão, onde, para cada nó, é seleccionado o atributo que fornece maior ganho de informação, desde que ainda não esteja presente nesse caminho, desde a raiz da árvore.

O anexo A.1 apresenta em maior detalhe alguns dos conceitos base essenciais ao domínio desta temática.

O Sobreajustamento[25] e a poda

O algoritmo ID3 conduz à criação de árvores de decisão que lidam correctamente com a maior parte das observações presentes no conjunto de desenho. Todavia, as árvores obtidas nesses casos são, muitas vezes, não balanceadas, longas e pouco eficazes, quando aplicadas ao conjunto de teste.

---

[24] Para questões de cálculo define-se $0 \times \log_2(0) = 0$.

[25] Sobreajustamento – Tradução de «overfiting»

O baixo desempenho sobre o conjunto de treino está relacionado com o fenómeno de sobreajustamento, traduzido na selecção de uma árvore de decisão que incorporou o ruído e regularidades ocasionais do conjunto de treino, conduzindo à sua desadequação, quando testada sobre o conjunto de teste. Por outras palavras, verifica-se sobreajustamento se, $h \in H$, for a hipótese seleccionada por análise do conjunto de treino e se existir uma hipótese $h' \in H$, que tenha um desempenho superior no conjunto de teste. Como foi referido, existem duas razões principais que concorrem para a criação desta situação: *i)* a existência de ruído, (i. e., observações mal classificadas ou excepcionais); e *ii)* a identificação de regularidades ocasionais nos subconjuntos criados pela partição induzida pelos atributos seleccionados. Contribui, para a verificação desta ocorrência, a selecção dos atributos para os nós terminais, tendo em conta um conjunto muito limitado de exemplos, em resultado das sucessivas partições e, logo, sem relevância estatística. Os dois casos contribuem para a indução de árvores que, apesar de adequadas ao conjunto de treino, apresentam resultados inferiores quando validadas. Existem estudos que indicam que o sobreajustamento pode contribuir para uma taxa de erro entre 10-25 por cento [75].

A correcção desta situação pode ser conseguida pela operação de poda, que é considerada, em muitos casos, o momento mais importante do processo de construção de uma árvore. A poda de uma árvore consiste na substituição de ramos por folhas, com o objectivo-base de reduzir a sua dimensão. Existem métodos de «poda *a priori*», que são aplicados durante a indução e que procuram controlar a dimensão das árvores e/ou suspender a selecção de atributos, quando se perde a relevância estatística. Estes métodos apresentam, como vantagem, reduzir o tempo na construção de ramos de árvores que não serão utilizados, porém obrigam à definição de limiares, muitas vezes artificiais. O limiar pode conduzir à interrupção do processo de indução de forma prematura ou, pelo contrário, permitir a indução de árvores de elevada profundidade.

Contudo, os métodos mais comuns são de «poda *a posteriori*», que iniciam o processo após a criação da árvore, considerando como candidatas todas as subárvores e efectuando a substituição sempre que o erro estimado for superior ao obtido pela utilização de uma folha [76]. Existem ainda métodos de poda aplicados às árvores, após a sua representação em regras, designados de *rule post-pruning* [73].

Genericamente, todos os métodos procuram estabelecer um equilíbrio entre a dimensão da árvore e o seu erro estimado, porém é necessário manter presente, tal como é referido em [77], que mais do que uma ciência, estamos na presença de uma «arte», em que os resultados nem sempre são os esperados, por vezes, as árvores originais apresentam melhores desempenhos do que as árvores podadas.

2.2.4.7 Classificador Naive Bayes

O classificador Naive Bayes utiliza a noção de incerteza, com a qual estamos intuitivamente habituados a lidar no nosso dia-a-dia, como conceito-base para a construção do seu algoritmo de indução. Tal como prevemos, o tempo que nos levará a ler um livro em função da nossa disponibilidade e interesse despertado pelo assunto, também fazemos a revisão da estimativa, em função da nossa progressão na leitura e no real interesse estimulado. A cada momento, revemos as nossas expectativas em função da nossa interpretação da realidade condicionada ao conhecimento obtido. No fundo, definimos crenças que modificamos em função da informação que adquirimos.

A aprendizagem Bayesiana tenta capturar essa realidade pela definição de um enquadramento de hipóteses e pela alteração das suas probabilidades, em função do conhecimento adquirido. Este modelo, atribui aos graus de incerteza valores contínuos, contidos no intervalo dos reais entre zero (que corresponde à inviabilidade) e um (que corresponde à certeza), permitindo a obtenção de um sistema de probabilidades convencionais [78]. Após a representação das hipóteses, como probabilidades, podemos fazer a sua manipulação, tirando partido do enquadramento probabilístico clássico (apresentado de forma resumida no anexo A.2). Neste enquadramento, $P(h_i)$ é a probabilidade *a priori* de uma hipótese, (ou seja a nossa crença de que h_i ocorrerá sem qualquer tipo de observação), e $P(h_i / x_j)$, a probabilidade, *a posteriori,* de h_i, (ou seja a revisão da nossa avaliação após tomarmos conhecimento da ocorrência de x_j).

Este enquadramento permite definir a avaliação em termos probabilísticos, em que os valores são resultado da leitura dos acontecimentos factuais precedentes. A aplicação deste princípio, na determinação das probabilidades de ocorrência das hipóteses em estudo, permite a determinação, em cada momento, de estimativas em função dos acontecimentos ocorridos, utilizado o cálculo da probabilidade *a posteriori,* para cada uma das hipóteses. Por outras palavras,

$$Escolha = \underset{hi \in H}{arg\,max}\,(p(h_i / D)), \qquad (38)$$

em que D representa o conjunto de acontecimentos conhecidos que condicionam as nossas hipóteses.

Uma das muitas aplicações possíveis para a aprendizagem *Bayesiana* é a indução de um classificador *Naive Bayes*. O espaço de hipóteses corresponde ao conjunto das classes possíveis e, os acontecimentos, às observações. A aplicação directa da equação (38) resulta agora em

$$\hat{c} = \underset{C_{i \in C}}{arg\,max} \quad p(c_i / x), \qquad (39)$$

sendo x a observação em estudo, c_i as possíveis classes e $\hat{c}$ o valor estimado, devido à sua maior probabilidade. Qualquer função que permita a determinação das probabilidades condicionais $p(c_i / x)$ é apelidada de função discriminante.

Tendo em conta que

$$p(c_i / x) = \frac{p(x / c_i)}{p(x)} p(c_i) , \tag{40}$$

resulta, por substituição de (40) em (39), que

$$\hat{c} = \underset{C_{i \in C}}{arg\ max} \quad \frac{p(x / c_i)}{p(x)} p(c_i) . \tag{41}$$

Assumindo que os acontecimentos são equiprováveis, podemos aproximar a equação (41) obtendo

$$\hat{c} \approx \underset{C_{i \in C}}{arg\ max} \quad p(x / c_i) p(c_i) . \tag{42}$$

Tendo em conta que os acontecimentos são representados por vectores de atributos, $\vec{x} = \langle a_1, a_2, \ldots, a_k \rangle$, a equação (42) assume a seguinte forma:

$$\hat{c} = \underset{C_{i \in C}}{arg\ max} \quad p(a_1, a_2, \ldots a_k / c_i) p(c_i) , \tag{43}$$

$$\text{ou ainda, } \hat{c} = \underset{C_{i \in C}}{arg\ max} \quad p(c_i) p(a_1 / c_i) p(a_2 / a_1, c_i) \ldots p(a_k / a_1, \ldots, a_{k-1}, c_i) . \tag{44}$$

Todavia, a equação (44) tem reduzida aplicabilidade, devido à necessidade de cálculo de um elevado número de probabilidades condicionadas. Dependendo das aproximações efectuadas, obtêm-se funções discriminantes distintas e, também, diferentes modelos.

O classificador *Naive Bayes*, utilizado nesta dissertação, obtém-se assumindo a independência dos atributos. Deste modo, a equação (44) pode ser rescrita, obtendo-se:

$$\hat{c} = \underset{C_{i \in C}}{arg\ max} \quad p(c_i) \prod_{j=a_1}^{a_k} p(a_j / c_i) . \tag{45}$$

em que as probabilidades podem ser estimadas por contagens das observações do conjunto de Desenho. O termo *Naive* provém da assunção simplista da independência dos atributos.

As aproximações realizadas na construção do classificador Naive Bayes, assumem as seguintes características:

- os atributos são independentes, não apresentando interdependências comuns;
- os atributos são identicamente distribuídos, não estando a sua ocorrência condicionada à sua posição relativa;

- a probabilidade de ocorrência das observações é semelhante, i. e., as observações são equiprováveis.

As aproximações descritas não estão asseguradas na classificação de textos, todavia diversos estudos comprovam um excelente desempenho em diversos domínios, em especial tendo em consideração que existe uma forte interdependência entre os diversos atributos. Uma justificação possível, apresentada em [79], passa pela manutenção da ordenação das probabilidades condicionais para os casos em análises.

2.2.4.8 Comparação dos algoritmos de indução

As principais vantagens da aprendizagem baseada em instâncias, *versus* baseada em modelos são: *i)* a ausência de assunção relativa ao modelo a inferir; *ii)* uma elevada eficiência computacional durante a indução, (limita-se a memorizar); *iii)* a fácil aplicabilidade e vasta aplicação, uma vez que, teoricamente, pode ser utilizada em qualquer problema desde que os parâmetros sejam correctamente definidos.

Como desvantagens, são, usualmente, identificadas: *i)* a baixa eficiência computacional no processo de estimativa, o momento mais exigente no que respeita ao desempenho do sistema; e *ii)* a elevada sensibilidade a variáveis irrelevantes e a excepções.

Os algoritmos apresentados utilizam mecanismos de representação e estratégias de pesquisa distintos. Porém, é comum fazer a sua análise segundo outras dimensões, tais como a do erro estimado, tempos de aprendizagem e de classificação, compreensibilidade, e graus de liberdade, entre outros.

A avaliação do desempenho, tendo por base o critério do erro estimado, é dependente dos dados e será realizada nos capítulos seguintes, tendo por base os resultados experimentais. Nesta secção, procura-se realizar a análise das restantes dimensões, (menos significativas no estudo de caso desta dissertação), tendo como base os resultados apresentados na literatura e na experiência adquirida.

A análise genérica dos tempos de aprendizagem e de classificação é difícil de efectuar, uma vez que depende fortemente da implementação realizada. Todavia, o método k-vizinhos apresenta o menor tempo, tendo em conta que o processo de aprendizagem está limitado ao registo eficiente dos exemplos presentes no conjunto de desenho. As árvores e o *Naive Bayes* apresentam valores muito semelhantes. Quanto ao tempo de classificação, as árvores de decisão são os métodos mais eficazes, seguidos do *Naive Bayes* e, finalmente, pelo k-vizinhos.

A compreensibilidade é muito subjectiva de avaliar e depende do sujeito em causa. Contudo é comum aceitar-se que as árvores de decisão são os modelos de mais fácil compreensão,

(em especial convertidos para um sistema de regras), seguidas do *Naive Bayes* e dos k-vizinhos.

No caso das árvores, o grau de liberdade na pesquisa está fortemente condicionado ao número de exemplos do conjunto de Desenho, sendo a dimensão da árvore directamente influenciada pelo número de exemplos. Pelo contrário, os k-vizinhos e o *Naive Bayes*, são condicionados pelo número de atributos, pelos seus valores possíveis e pelo número de classes. Neste caso, o número de exemplos não condiciona os graus de liberdade, apesar de, no caso dos k-vizinhos, contribuir, linearmente, para o aumento do espaço de pesquisa, no momento da avaliação de novas observações.

O comportamento dos algoritmos pode ser realizado, igualmente, pela análise da sua variância e tendência, que avaliam, respectivamente, a influência dos dados e o desvio indutivo. De modo genérico, uma baixa variância é sinónimo de uma alta tendência e vice-versa. As árvores e o k-vizinhos apresentam uma elevada variância, pelo que pequenas alterações no conjunto de treino podem induzir fortes alterações nos modelos, alterando o seu comportamento. Pelo contrário, no caso do classificador Naive Bayes a variância é muito baixa, pelo que a sua tendência é mais acentuada.

No processo de avaliação de uma nova observação, as árvores utilizam, exclusivamente, os atributos que se encontram ao longo do percurso de avaliação. Por contraste, tanto o Naive Bayes como os k-vizinhos utilizam todos os atributos da observação em análise. Para além disso, no caso das árvores, o percurso nos ramos descreve a condição suficiente para que uma nova observação seja classificada. No caso do Naive Bayes é necessário realizar a análise de cada observação, tendo em conta um número total de classes, com vista a determinar qual a hipótese mais provável. Os k-vizinhos obrigam, por sua vez, ao cálculo da vizinhança sendo, então, determinada a classe predominante no perímetro delineado.

2.2.4.9 Sistemas de Apoio à Decisão (SAD)

Independentemente do domínio de aplicação, é sempre possível gerar estimadores distintos para o mesmo Conjunto de Desenho, o que obriga a identificar um processo para lidar com múltiplas estimativas para a mesma observação. Para além da criação de processos de selecção do melhor, a intuição de que diferentes estimadores se comportam de forma qualitativamente distinta, tem motivado a pesquisa de processos da sua combinação em meta-estimadores.

Esta área de investigação visa identificar processos de combinação das estimativas individuais de diversos modelos. A combinação dos múltiplos modelos é conseguida através da definição de um Processo de Tomada de Decisão (PTD), que permite agregar as estimativas individuais.

A Figura 8 ilustra o modelo subjacente à noção de tomada de decisão, realizada pelo PTD incorporado no SAD. Como se pode observar, após a atribuição de uma classificação individual por cada estimador, cabe ao PTD realizar a tomada de decisão final conducente à estimativa do SAD. O SAD pode ser decomposto nas seguintes tarefas atómicas: *i)* representação do documento na forma vectorial; *ii)* avaliação do documento; *iii)* elaboração de estimativa por cada um dos classificadores; e *iv)* tomada de decisão.

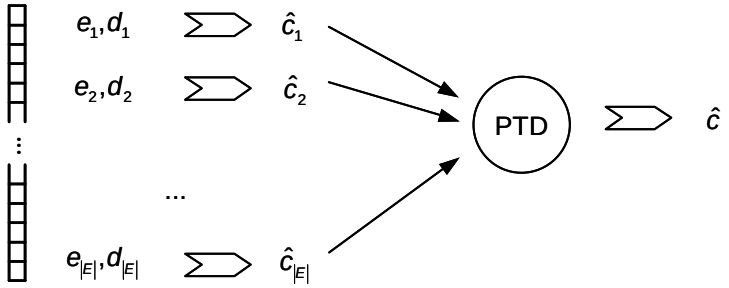


Figura 8 – Modelo referência do Processo de Tomada de Decisão (PTD) incorporado no Sistema de Apoio à Decisão (SAD)

Para aferir a qualidade do método de combinação é, muitas vezes, utilizado o modelo abstracto *oracle*. Neste modelo, as estimativas dos classificadores são comparadas com a classe da observação, e na existência de uma estimativa correcta, o PTD estima, igualmente, de forma correcta. Desta forma, captura-se o modelo ideal que reconhece imediatamente um classificador correcto, independentemente dos restantes classificadores.

Por outras palavras,

$$I = \{i_1, \ldots, i_{|E|}\}, \quad i_j = (e_j, d_j), \tag{46}$$

correspondendo e_j a um modelo e d_j ao seu respectivo desempenho, e seja

$$PTD(I): S_1 \times \ldots \times S_K \rightarrow C, \tag{47}$$

uma relação em que S_i toma um conjunto de valores discretos e finitos

$$S_i = \{s_{i,1}, \ldots, s_{i,|S_i|}\}, \text{ e} \tag{48}$$

$$C = \{c_1, \ldots, c_{|C|}\} \tag{49}$$

encontrar o PTD que melhor permita discriminar C.

A aproximação mais comum para a criação do PTD é a **selecção *a priori* do classificador** com melhor desempenho global. Os restantes não são considerados, pelo que a decisão final de classificação resulta, exclusivamente, de uma única estimativa. Por outras palavras:

$$SAD = \underset{e_j \in |E|}{arg\,max} \quad (d_J) \tag{50}$$

sendo a tomada de decisão efectuada, tendo em conta a estimativa do melhor classificador, e_j .

Esta aproximação é largamente utilizada e oferece um conjunto de vantagens consideráveis. Apesar de ainda não ter sido identificado nenhum algoritmo que apresente um desempenho superior em todos os problemas de aprendizagem, na maioria dos casos, para cada problema específico, existe um tipo de algoritmo que apresenta um comportamento mais adequado. Esta constatação conduz à selecção do referido algoritmo e à sua optimização. Sendo assim, a selecção de um classificador adapta-se a uma estratégia de especialização e optimização do melhor tipo de algoritmo de indução.

Outras vantagens são: *i)* a redução da complexidade do problema por especialização (reduzindo o SAD a um único classificador final); e *ii)* diminuição dos custos computacionais associados à tomada de decisão (por contraste com a utilização de diversos classificadores). Todavia, a selecção de um único classificador para servir de SAD não deixa de ser potencialmente redutora, em especial se existir uma forte semelhança de desempenho dos modelos e, como tal, dos factores de confiança atribuídos às estimativas. Quanto maior for a dúvida sobre a certeza da selecção efectuada, maior a necessidade de identificar estratégias de combinação de modelos.

A relevância da aplicação de métodos mais elaborados para combinação de modelos está directamente relacionada com a existência de um grau substancial de desacordo entre estimadores. A inexistência de desacordos retira o valor acrescentado, devido à forte correlação entre os resultados produzidos. Assim, assume-se *a priori* a utilização de modelos com um desempenho superior à estimativa aleatória e com erros não correlacionados. A utilidade da combinação de modelos pode ser justificada pelo facto de se poder assumir que a componente do erro de cada modelo está relacionada com a sua estabilização, num máximo local, sendo expectável um comportamento distinto de cada modelo condicionado à localização no espaço de pesquisa das observações a serem classificadas. O processo de combinação ideal possibilitaria, em cada momento, explorar o modelo, (ou conjunto de modelos), que melhor se adequem à localização da observação em análise.

Tendo por base a criação de PTD, a solução mais comum, passa pela utilização da **regra da maioria**, i. e., a classificação final atribuída pelo SAD é o resultado da tendência da maioria dos classificadores.

Por outras palavras,

$$PTD = \underset{c_j \in C}{arg\,max} \sum_{i=1}^{|E|} \hat{c}_i = c_j \qquad (51)$$

em $\hat{c}_i$ é a estimativa atribuída a um documento pelo i-ésimo classificador.

A utilização deste PTD permite a obtenção de empates, pelo que é necessário definir uma regra de desempate que possibilite, nesses casos, a obtenção de uma classificação. Esta regra não é simples, obrigando à utilização de arbitrariedades, por isso deve ter-se em consideração a natureza do caso em estudo. As soluções típicas entram em conta com o número e desempenho dos classificadores, o que, na maioria dos casos, só permite reduzir a probabilidade de se obter um empate, pelo que uma solução arbitrária é, muitas vezes, equacionada. Existem casos particulares em que a solução é simples, tais como a utilização de um conjunto de classificadores ímpar que permite evitar a situação de empate.

Uma evolução possível é a **regra da maioria ponderada,** em que as contribuições de cada classificador, são pesadas em função do seu desempenho. Por outras palavras,

$$PTD = \underset{c_j \in C}{arg\,max} \quad \sum_{i=1}^{|E|} (\hat{c}_i = c_j) \times d_j \tag{52}$$

em $\hat{c}_i$ é a estimativa atribuída a um documento pelo i-ésimo classificador, e d_i a ponderação atribuída ao i-ésino classificador.

Esta aproximação permite entrar em conta, no processo de tomada de decisão, com o valor intrínseco associado a cada estimativa, premiando, naturalmente, as estimativas produzidas por classificadores com desempenho superior. O factor de ponderação pode ser trabalhado, permitindo uma discriminação não linear dos classificadores, através do aumento do peso relativo dos melhores. Todavia, elaborações nesta direcção devem ser realizadas na presença de casos concretos e, se possível, validadas de forma experimental, evitando decisões arbitrárias que possam, ao contrário do que se espera, penalizar o desempenho final do SAD. A utilização de um sistema de apoio à decisão, baseado na regra da maioria ponderada, permite assegurar a capacidade de decidir, tendo, à partida, um conjunto de classificações que podem não ser coincidentes. Todavia, nem sempre a maioria tem razão. A análise detalhada das experiências efectuadas com grupos de classificadores, permitiu concluir que, apesar de existir um consenso alargado, na maioria dos casos existe, pelo menos, um classificador a estimar uma classificação diferente. Esta classificação, que pode estar correcta, é abandonada em favor da maioria. O ideal seria a utilização de um método que:

- na presença de modelos correlacionados, diminuísse o seu peso relativo na tomada de decisão;
- identificasse a especialização espacial de cada modelo, assumindo que os modelos estão presos em máximos locais, aumentando o seu peso relativo, dependendo da proximidade da localização da observação em análise;

- fosse estável a pequenas variações de estimativas dos modelos, o que permitiria evitar que o processo de combinação ficasse longe do óptimo.

Uma proposta de solução, **os modelos probabilísticos,** intensivamente utilizados [80-83], (para classificadores que estimam as classes de forma probabilística), foi formalizada em [84]. Estes modelos, pressupõem que $E = \{E_1,...,E_N\}$ é um conjunto de classificadores e que se verificam as seguintes premissas:

- Todos os classificadores estimam uma classe para cada observação de forma probabilística, i. e., $d_{j,i}(x) \in [0,1]$ é $p(c_i \mid x)$ (probabilidade, *à posteriori,* da observação x da classe c_i, atribuída pelo E_j, para $i = 1,...,|C|$);
- Existem duas classes, $C = \{c_1,c_2\}$, e $d_{j,1}(x) + d_{j,2}(x) = 1$ para $j = 1,...,N$;
- Uma observação é considerada pertencente a c_1 no caso de $d_{j,1}(x) > 0.5$;
- Os estimadores são independentes.

É possível definir como PTD:

$$d_i(x) = F(d_{1,i}(x),...,d_{N,i}(x)), \quad i = 1,...,|C| \tag{53}$$

onde F representa o método de combinação. Os métodos mais usuais são: *i)* mínimo; *ii)* máximo; *iii)* média; *iv)* mediana; *v)* produto.

Tendo em consideração as premissas, a equação (53) pode ser reescrita

$$d_1(x) = F(d_{1,1}(x),d_{2,1}(x),...,d_{N,1}(x)) \tag{54}$$

$$d_2(x) = F(1 - d_{1,1}(x),1 - d_{2,1}(x),...,1 - d_{N,1}(x)) \tag{55}$$

Selecciona-se, como estimativa do grupo, a classe que obtiver maior probabilidade. É necessário ter em atenção que, à excepção dos casos de utilização de um único classificador; a média; ou a mediana, $d_1(x) + d_2(x)$ podem ser diferentes de um.

Os resultados apresentados variam muito em consequência dos casos específicos. Em [84] para distribuições uniformes de $d_{j,i}(x)$, a função mínimo e máximo, obtêm os melhores resultados e, para distribuições normais, todos os métodos apresentam um erro semelhante.

De âmbito mais geral**, os modelos combinatórios lineares** realizam a fusão dos resultados de cada estimador, utilizando um método de pesagem definido experimentalmente, para ponderar as contribuições individuais.

$$Score_c(x) = k_1 \times score_{E_1} + k_2 \times score_{E_2} + ... + k_N \times score_{E_N} \tag{56}$$

A ponderação das contribuições está fortemente relacionada com o desempenho dos estimadores. Em [85] aplica-se esta aproximação a estimadores k-vizinhos e classificadores

probabilísticos, (para a realização de anotações), com melhores resultados do que as soluções individuais. Em [86] é apresentado um estudo com combinações lineares pesadas em pares e trios de classificadores do tipo vizinho mais próximo, Naive Bayes e «Relevance Feedback», em que os pesos atribuídos são proporcionais ao desempenho. Os resultados obtidos são superiores à utilização individual dos classificadores. Uma evolução com bons resultados, apresentada em [87], é a **combinação linear normalizada**, que permite ponderar a influência individual de cada estimador. Em [88] é apresentado um método de indução do PTD de forma dinâmica com o recurso à determinação dos pesos a atribuir a cada classificador, através de algoritmos genéticos.

Outra abordagem, **a combinação hierárquica** de classificadores é extremamente útil, especialmente no caso de classificações organizadas de forma hierárquica, comuns nalguns domínios, e. g. Medicina. Para além de poder tirar partido na natureza hierárquica da estrutura de classificação, permite a especialização de classificadores, uma vez que o encadeamento corresponde a assumir uma divisão do espaço de pesquisa realizada pela camada anterior. Os resultados obtidos em [89] descrevem um aumento de precisão e de rechamada para a classificação IC9 (um vocabulário alfabético hierárquico de termos com a utilização de um modelo hierárquico de encadeamento).

Finalmente, a **generalização por empilhamento**[26], proposta em [40] concebe o PTD recorrendo a uma arquitectura de níveis. Cada nível recebe, do seu anterior, os dados originais e/ou as estimativas previamente efectuadas. Os novos dados, recebidos em cada nível, são tratados como um novo problema de aprendizagem, sendo induzido um novo estimador responsável por atribuir uma estimativa. Apesar do modelo poder ser aplicado de forma multinível, as aplicações mais comuns reduzem a sua utilização a dois níveis e adoptam a seguinte denominação: para o nível zero - dados nível zero e estimadores; para o nível um - dados nível um e generalizador.

A generalização por empilhamento pode ser vista como uma versão sofisticada dos modelos de validação cruzada em que é escolhido o melhor estimador ou, na combinação por votação através de combinações probabilísticas. Nesta abordagem, os PTD são induzidos com técnicas de aprendizagem, a fim de seleccionarem, em cada momento, os melhores estimadores-base a utilizar. Uma interpretação possível do processo de generalização é determinação de um filtro adequado ao desvio colectivo dos estimadores, com o objectivo de aumentar o desempenho global do sistema [40].

A distinção na operacionalização da arquitectura está fortemente relacionada com: *i)* o tipo de estimadores a utilizar para o primeiro nível; *ii)* o tipo de atributos utilizados como dados de entrada e; *iii)* o tipo de algoritmo de indução dos generalizadores.

---

[26] Generalização por empilhamento - Tradução do autor para «*Stacked Generalization*»

Em [90] demonstrou-se o sucesso da aplicação utilizando: como estimadores de nível zero - modelos de regressão linear multivariável; como dados nível um - algumas das suas saídas; e para o generalizador - o algoritmo «least-squares linear regression».

Em [91], por seu lado, esta abordagem foi aplicada a três *corpus* distintos, utilizando diversas combinações para estimadores de nível zero: C4.5, Naive Bayesian e k-vizinhos; para os dados de nível um: as sua estimativas probabilísticas; e como generalizadores: C4.5, Naive Bayesian e k-vizinhos, a maioria, e a «multi-response linear regression» *(MLR)*. Os melhores resultados foram obtidos com a utilização de probabilidades de classificação e com o generalizador MLR (em especial quando um dos algoritmos apresenta um desempenho muito superior aos restantes).

Em [92] os estimadores de nível zero utilizados, foram árvores de decisão, Naive Bayesian, estimadores probabilísticos, baseados em unigramas, e SVM; para os dados nível um foram utilizadas as estimativas e indicadores de fiabilidade dos estimadores relacionados (por exemplo com a dimensão do texto, o número de variáveis, etc.); e para os generalizadores Naive Bayesian e SVM. A utilização dos generalizadores permitiu melhorar o desempenho final do sistema, devido à sua capacidade de capturarem informação sobre as zonas do universo de pesquisa, em que os estimadores-base apresentam um desempenho superior.

Num comentário final, é genericamente aceite que a combinação de estimadores permite melhorar o desempenho do sistema. A votação simples e a votação com ponderação são os métodos mais habituais, sendo comum o relato da melhoria do desempenho dos sistemas. A explicação reside no facto da média reduzir o efeito da variância dos classificadores, diminuindo, assim, a sobreposição da avaliação entre documentos relevantes e irrelevantes, permitindo a sua melhor identificação. Todavia, estes métodos não têm em consideração o grau de confiança individual de cada classificador, para cada classe.

A combinação de classificadores através de estatística linear, e de estatística de ordem, é igualmente apresentada como veículo para a obtenção de bons resultados. Todavia, apesar da estatística de ordem ser robusta, não é muito eficiente, por exigir elevadas quantidades de dados (i. e., classificadores), o que nem sempre é viável.

As premissas assumidas são bastante razoáveis, com excepção da independência dos estimadores, tendo em consideração que, mesmo os classificadores construídos de forma independente, possuem correlações, devido a desvios semelhantes, (impostos pelos métodos de indução), e por existirem zonas do espaço de pesquisa difíceis para todos os classificadores [84]. Com vista à redução da correlação dos estimadores é possível utilizar processos distintos de indução de múltiplos modelos, baseados na alteração da distribuição dos dados, pela utilização de amostragem; entre outros, o «bagging» [93] e «boosting» [94, 95].

Para terminar, realça-se que o processo de elaboração de cada estimativas intermédias é uma tarefa executada por cada um dos classificadores de forma individual e independente, o que permite a sua execução em paralelo. Considerando a possibilidade de execução em paralelo, o processo de tomada de decisão pode ser reinterpretado da seguinte forma: após a representação do documento em formato vectorial, cada classificador inicia a análise e elabora a sua estimativa de classificação. Sempre que um classificador termina sua estimativa, passa essa informação para o PTD, cabendo a este, e assim que possível, perante o conjunto de informações disponíveis, elaborar a estimativa final de classificação do documento.

Esta abordagem permite acelerar o processo de tomada de decisão, devido à possibilidade de realização de uma estimativa final de classificação com uma visão parcial, mas suficiente das estimativas intermédias, e por explorar o processamento em paralelo.

2.3 Representação de conhecimento suportado por ontologias

Numa época marcada pela Web de primeira geração, vislumbra-se a sua maturação numa Web de segunda geração, a Web Semântica, onde a formalização e utilização de conhecimento, em sentido lato, representam factores centrais no desenvolvimento de sistemas de informação. Todavia, uma aproximação consensual à sua especificação está longe de ser atingida, existindo diversas aproximações, visões e normas «de facto». Na área da exploração de informação, uma das aproximações mais comuns baseia-se na utilização de ontologias. As ontologias são um dos pilares na concretização da Web Semântica, tendo em conta que permitem a formalização e a partilha de conhecimento. Numa primeira definição, uma ontologia pode ser encarada como o conjunto dos meta-dados, processáveis de forma automática, que explicitamente descrevem a semântica dos dados. O estudo de ontologias tornou-se num tópico popular de investigação em diversas comunidades, tais como, engenharia do conhecimento, processamento de linguagem natural, sistemas de informação cooperativos, integração de informação inteligente e gestão de conhecimento [96]. A primeira referência conhecida, a **ontologia**, remonta ao século dezassete, como sinónimo de metafísica e de primeira filosofia, tal como foi definida por Aristótles no século IV a. C. Contudo, com o passar do tempo, deixa de ser sinónimo de metafísica, que evoluiu separadamente incluindo um novo conjunto de estudos. Actualmente, a Enciclopédia Britânica define-a como: «a teoria ou estudo do ser como tal; questionando quais as características-base de toda a realidade» [97], sendo assim encarada como uma teoria filosófica acerca da natureza da essência. Abreviando, **ontologia** é uma teoria filosófica acerca da natureza da existência, procurando responder à pergunta: «O que é um conceito?», ou de outra forma, «Quais as especificidades que descrevem um conceito, atribuindo-lhe um significado único?». Neste sentido a utilização do plural de ontologia não tem significado, uma vez que a existência de mais do que uma ontologia, implica distintas

descrições para um conceito, perdendo-se a propriedade de unicidade. Contudo, em informática é comum a utilização do plural, muitas vezes associada a uma divisão por domínios de aplicabilidade. Nesta área, a utilização de ontologias promove o entendimento comum e partilhado de um domínio de conhecimento que pode ser comunicado entre pessoas e sistemas aplicacionais [98]. Consequentemente, as ontologias são ferramentas que, permitindo um entendimento comum de um domínio, facultam a partilha de conhecimento. Tim Berners-Lee, Director do World Wide Web Consortium em [99], defende que uma ontologia típica é constituída por uma taxinomia que define os conceitos e as suas relações e por um conjunto de regras de inferência que potenciam capacidades de raciocínio. Uma definição, genericamente aceite para ontologia, foi proposta em [100] como sendo «uma especificação, explícita e formal, de uma conceptualização partilhada», em que os termos assumem os seguintes significados:

- Explícita: a inexistência de graus de liberdade, encontrando-se cabalmente definidos quer os tipos de conceitos, quer as relações e restrições existentes;
- Formal: o facto de poder ser processada automaticamente;
- Conceptualização: a criação de um modelo abstracto de um fenómeno, permitindo a sua descrição e identificação;
- Partilha: a necessidade de representação de um conhecimento consensual, que não se restringe a um indivíduo, mas que seja aceite por um grupo.

A representação de conhecimento é tema de investigação na informática desde o início dos anos sessenta, tendo sido introduzida por Ross Quillian, que propôs a utilização de redes semânticas para o processamento de linguagem [101, 102]. Nos últimos anos sofreu um vigoroso incremento, principalmente devido a factores exógenos à IA, entre outros:

- à disponibilização de dicionários, *thesaurus*, e glossários consistentes e substancialmente completos, com extrema relevância na representação normalizada de conhecimento sobre um conjunto alargado de domínios;
- à necessidade de alterar a situação actual de disponibilização de informação não estruturada na Web, o que dificulta a sua manipulação automática, e também na sua conversão num meio de disponibilização de recursos com significado semântico, permitindo uma utilização mais produtiva desse conhecimento [103];
- a uma visível desadequação dos sistemas aplicacionais disponíveis, causada pelo crescimento constante da informação, pela pouca eficácia dos motores de pesquisa e pelo caos dos universos de discurso, fomentados pela inexistência de normas de partilha de conhecimento, entre humanos e/ou sistemas aplicacionais.

Importa realçar que, apesar deste tema apresentar semelhanças funcionais com os meta-dados das bases de dados, existem diferenças marcantes, tais como [104]:

- a linguagem de definição de ontologias é sintáctica e semanticamente mais rica do que as aproximações comuns para as bases de dados;
- o conhecimento é descrito através de textos em linguagem natural semiestruturada, o que não acontece com as bases de dados;
- numa ontologia é necessária a existência de uma terminologia comum e partilhada, tendo em conta que o objectivo último é a partilha e transferência de informação universal;
- uma ontologia disponibiliza uma teoria de domínio, não sendo um contentor de dados estruturado;
- uma ontologia permite uma partilha e um entendimento comum de um domínio de conhecimento, através de um registo sintáctico e semântico, viabilizando a comunicação entre actores (sistemas e pessoas).

São vastas as áreas de aplicação da representação de conhecimento na Engenharia Informática. Uma primeira, relaciona-se com a sua utilização no desenvolvimento de motores de pesquisa, permitindo antever uma nova geração de aplicações, que ultrapassa alguma das actuais limitações, entre outras: *i)* pesquisas limitadas a perguntas por palavra, (geradoras de respostas irrelevantes por utilização das palavras em contextos inapropriados); *ii)* inadequação de navegação por humanos e máquinas por falta de estruturação; *iii)* difícil manutenção devido à elevada complexidade. Ultrapassadas estas limitações será possível realizar pesquisas «inteligentes», com o recurso a mecanismos de partilha de informação, de definição de vistas, e de processos criativos de pergunta/resposta.

Outra área relaciona-se com a representação do conhecimento corporativo. A flexibilidade de resposta e a competitividade de uma organização estão intimamente ligadas à sua capacidade de manter e gerir conhecimento. O conhecimento, mesmo na maioria das organizações mais evoluídas, continua a ser conservado na memória dos colaboradores, o que, na prática conduz a uma dependência de aprendizagem baseada na tradição oral. A informação, quando registada em meios digitais, não está armazenada de forma estruturada, está dispersa numa teia de ficheiros de distintos formatos (texto, imagem, áudio, multimédia, vídeo), sem qualquer organização global, dependente do indivíduo directamente responsável pela sua manutenção. Esta situação constitui uma dependência e fragilidade inaceitável para algo tão sensível como a memória corporativa. A informação é inútil se não puder ser aplicada. Na prática, deixa de ser informação, transformando-se num conjunto de dados avulsos. Consequentemente, a representação do conhecimento conquista um lugar central como método de aquisição, manutenção e acesso ao conhecimento de uma organização, com o objectivo de explorar o conhecimento corporativo,

de forma a permitir uma maior produtividade e valor acrescentado, resultando daí uma maior competitividade.

Para além da gestão de conhecimento dentro das organizações, abrem-se novas perspectivas de interoperabilidade com o exterior, quer seja a nível de relações privilegiadas e duradouras, e. g., B2B «Business-to-Business», ou mais ocasionais, e. g., B2C «Business-to-Customer». Existem iniciativas com vista à criação de mecanismos de interoperabilidade entre empresas desde os anos sessenta, tendo, desde então, sido criados diversos protocolos de interligação que permitem a troca de dados. Na prática, as empresas definem um modelo de dados e a sua semântica e implementam conversores que permitem a troca de informação. A norma EDIFACT, uma iniciativa das Nações Unidas, é um exemplo que permite a realização de transacções comerciais [105]. Todavia, mesmo esta norma de sucesso, não atingiu as expectativas. Entre as principais causas estão o seu isolamento (e. g., não integra processos de partilha de documentação); e o elevado custo associado à sua implementação por exigir uma estrutura de procedimentos muito repetitiva e indutora de erros. A utilização da Internet perspectiva uma alteração significativa neste cenário. A utilização das ferramentas de navegação na Web como programas de interface-base no acesso aos dados, previamente descritos de forma estruturada, com vista a não perderem a sua semântica, contribui para uma adopção mais rápida por parte do utilizador, tendo em conta o domínio prévio da ferramenta. Os primeiros portais a recorrerem a esta tecnologia, numa perspectiva comercial, foram o Mysap.com, Vertical.com e Harbinger.net [96]. Todavia, por estarem na vanguarda, não tiraram partido de um novo conjunto de tecnologias que atingiram a sua maturidade mais tarde, tendo sido essencialmente desenvolvidos em HTML estão limitados à sua falta de expressividade sintáctica e semântica. Como solução para ultrapassar esta situação o recurso a novas normas, que fazem uso de ontologias, assumem um papel relevante, uma vez que evitam a definição, *a priori*, de estruturas de dados normalizadas e de terminologias-base.

2.3.1 Paradigmas de representação de ontologia

Os paradigmas mais utilizados para a representação de ontologias são as *frames*, a lógica descritiva, e, naturalmente, soluções híbridas que procuram explorar as vantagem de cada aproximação.

Na representação por *frames* o conhecimento é descrito através de uma estrutura de dados, visualmente apresentada como uma rede, em que os nós são conceitos (estereótipos) e os elos, (entre os nós), as relações. Um conceito contém (ou pode conter) atributos, valores por omissão, relações com outros conceitos e restrições sobre relações de conceitos. A conceptualização é, assim, realizada através da utilização da rede de conceitos e das suas relações.

As principais vantagens deste paradigma residem: *i)* no facto de ser genericamente aceite que a representação por *frames* é simples e fácil de ser interpretada por humanos; *ii)* na forma natural como a relação de herança é transposta; e *iii)* na facilidade de realizar inferência sobre uma estrutura de *frames*.

Como principais desvantagens são apontadas: *i)* o obstáculo à representação natural da negação e da disjunção; *ii)* a inexistência de qualificadores na linguagem (e. g., conceitos de agregação como «todos os»); e *iii)* a dificuldade de modelar as relações de N para N. Acresce o facto dos sistemas de *frames* serem, por vezes, erroneamente interpretados como uma linguagem de programação orientada por objectos, particularmente, devido ao mecanismo de herança que é assegurado através de elos. Finalmente, existe o problema de identificar os conceitos estáticos, em especial, num mundo em mudança. Este problema foi descrito em 1969, [106] como o problema das *frames*, sendo vastamente discutido em [107], onde é reapresentado como o problema de descrição completa, i. e., a impossibilidade de fornecer as condições necessárias e suficientes sobre uma «coisa». Todavia, apesar da impossibilidade de resolução teórica deste problema, é de referir que os humanos sofrem da mesma incapacidade, apresentando um elevado sucesso na representação de conhecimento.

A alternativa à representação por *frames*, é a representação através de uma linguagem de lógica descritiva [108, 109]. A lógica descritiva pertence à família de linguagens formais de representação de conhecimento baseado em lógica, adequada não só à «representação sobre», mas igualmente ao «raciocínio acerca de». A lógica descritiva é um subconjunto da lógica de predicados de primeira ordem, focada em categorias, e na sua definição em termos das suas relações. Os blocos-base de construção são os conceitos, os papéis[27] e os indivíduos. Os **conceitos** descrevem as propriedades comuns a um conjunto de elementos e podem ser considerados predicados unários que são interpretados como conjuntos de objectos. Os **papéis** são interpretados como relações binárias sobre objectos. Os **indivíduos** são instâncias de conceitos.

A lógica descritiva é muito expressiva e, tendo em conta que os princípios lógicos estão muito próximos da estrutura de ontologia, a capacidade de inferência é natural, permitindo a identificação de subconjuntos (e. g. a categoria A é um subconjunto de B?). Genericamente, conceitos e papéis complexos são definidos através de um conjunto de construtores, a partir de conceitos atómicos-base [110]. Cada linguagem define o seu conjunto de construtores (tais como conjunção, intersecção, qualificadores de papéis, etc.) [111].

As principais áreas de aplicação são, para além da representação de ontologias, a representação de conhecimento terminológico, de configurações, entre outras [112].

[27] Papéis - Tradução de roles.

Todavia, o formalismo lógico resulta numa barreira para a generalidade dos utilizadores, dificultando a sua interpretação. A linguagem formal é considerada mais complexa e os mecanismos associados desencadeiam efeitos secundários, entre outros, a reestruturação da base de conhecimento de forma autónoma, através dos mecanismos de inferência, o que pode confundir o utilizador.

Numa análise comparativa, as *frames* são um paradigma de representação mais intuitivo para o utilizador comum, contribuindo para isso a possibilidade de uma representação gráfica e a não utilização de uma linguagem formal baseada em fórmulas lógicas. Uma vantagem da lógica descritiva é a capacidade de determinação por inferência de subconjuntos que, no caso da representação por *frames*, tem que ser explicitamente declarada. Tendo em conta que outras relações entre conceitos, tais como disjunção e consistência, podem ser representadas por subconjuntos, a propriedade de determinação de subconjuntos permite, igualmente, por transitividade, a sua determinação. Esta capacidade de determinação automática das relações pode ter um papel extremamente crítico na verificação da consistência da ontologia, em especial se a sua complexidade for elevada.

Independentemente do paradigma utilizado, existem dois princípios-base para o desenho de ontologias: a observação, por identificação directa de conhecimento do meio físico; e o raciocínio, que busca o sentido geral na procura de um enquadramento abstracto das observações efectuadas. Todavia, a determinação dos conceitos nem sempre é acessível, aumentando o grau de dificuldade com o crescimento da dimensão do domínio ou nos casos em que os conceitos são partilhados em diversos domínios. Genericamente, conceitos muito especializados ou muito complexos nunca são fáceis de identificar. Esta dificuldade é um sério obstáculo à obtenção de ontologias consistentes, ao qual se acrescentam: a existência de casos raros; a omissão de casos no processo de abstracção (definição incompleta); conflito de pressupostos; e aplicações da ontologia a casos não esperados.

2.3.2 Linguagens de representação de ontologias

Nos últimos anos existiu um vasto esforço na criação de linguagens de representação de ontologias todavia, nenhuma se impôs de forma marcante. Neste período, a maioria das iniciativas foi abandonada prematuramente, devido ao aparecimento de novas aproximações, resultantes da união de resultados obtidos por grupos distintos. Deste cenário, surgiu um conjunto de imbricado de linguagens que, no início do seu processo de afirmação, ou desapareceram ou foram aglutinadas em novos esforços.

Em seguida, são apresentadas algumas das linguagens que atingiram maior protagonismo, ilustrativas dos diversos paradigmas-base utilizados: lógica de primeira ordem; *frames*; ou adopção directa de tecnologia Web.

A Ontology Interchange Language (OIL) foi proposta no âmbito do projecto OnToKnowledge (http://www.ontoknowledge.org). Esta linguagem procura tirar partido do melhor dos três paradigmas, propondo modelação por *frames*, descrição semântica em lógica de primeira ordem, e sintaxe através de tecnologia Web com recurso a esquemas XML [113] ou RDF [114]. A linguagem OIL foi utilizada com sucesso em diversas áreas das quais se destaca o comércio electrónico.

A linguagem DARPA[28] Agent Markup Language (DAML) foi desenvolvida com o suporte do programa DARPA desde Agosto de 2000, como uma extensão ao XML e ao RDF [115]. Em Março de 2001, num esforço de unificação, o comité conjunto US/EU ad hoc Agent Markup Language, propõe uma nova linguagem de representação de ontologias DAML+OIL (ver www.daml.org/2001/03/daml+oil-index). Esta nova linguagem aglutina as linguagens anteriores sendo apresentada como capaz de ultrapassar os problemas das linguagens originais, tornando a semântica da linguagem mais clara e, permitindo a sua interoperabilidade com várias ferramentas baseadas em lógica descritiva [116]. Esta linguagem é construída, uma vez mais sobre extensões, que aumentam as primitivas de modelação das normas RDF e esquemas RDF. A DAML+OIL é, assim, escrita em RDF, que é, por sua vez, escrita em XML (utilizando XML *namespaces* e URI) [117].

A linguagem permite a definição de conceitos, através de classes, de relações entre conceitos, através de propriedades, e de restrições através de restrições às propriedade [118]. O universo é representado recorrendo a dois processos distintos: valores que pertencem a tipos de dados (representados em esquema XML), apelidados de domínio de tipo de dados, e objectos, descritos através das classes, (representadas em RDF) utilizando a notação DAML+OIL [117]. O mecanismo de criação de restrições, via restrições às propriedades, define, implicitamente, classes anónimas que contêm todos os objectos que satisfazem a restrição. A existência de dois processos de descrição do universo, conduz à existência de dois tipos de restrições: restrições de objecto (que aplicadas a propriedades «objecto» permitem descrever restrições entre objectos); e restrições de tipos de dados, (que aplicadas às propriedades «tipo de dados» permitem descrever restrições entre objectos e tipos de dados). Contudo, no final de 2003, o gestor do programa DAML, apesar de considerar o programa um sucesso, (por ter criado uma linguagem de representação de conhecimento revolucionária, que produziu um impacto determinante na estrutura-base da Web), reconhece os resultados obtidos pela linguagem OWL, (ver 2.3.3), nomeadamente, o vasto conjunto de ontologias, as ferramentas criadas, e o reconhecimento pelo W3C como candidato a recomendado[29]. O que aconteceu, em pleno contraste com a DAML+OIL, que não obteve sucesso paralelo na disseminação, tendo ficado confinada à sua esfera de

[28] DARPA - *Defense Advanced Research Projects Agency*

[29] Candidato a recomendado – Um estado no processo de certificação de uma norma W3C.

influência directa. Neste sentido o gestor do programa propõe que os dois últimos anos do programa, sejam utilizados na migração do trabalho efectuado para a linguagem OWL e no suporte à promoção dessa linguagem. Naturalmente que esta postura compromete o futuro da linguagem DAML+OIL e fortalece a esfera de influência do OWL [119].

A Knowledge Interface Format (KIF) é uma linguagem de representação de conhecimento desenhada para permitir a partilha de conhecimento em sistemas distribuídos. A linguagem é baseada em lógica de predicados de primeira ordem. As suas características mais relevantes são a semântica declarativa e o facto de ser compreensiva. A KIF não foi desenvolvida para servir de interface com o utilizador nem para armazenamento, em formato interno, de conhecimento, embora seja muitas vezes utilizada nesses sentido [120]. A linguagem foi, assim, definida para ligações entre aplicações, em que os formatos nativos de representação de conhecimento são traduzidos para KIF. A KIF possui uma semântica declarativa, o que permite a sua interpretação directa, permitindo a utilização de expressões lógicas. A KIF é uma versão prefixa do cálculo de predicados de primeira ordem, com extensões para aumentar a sua expressividade. Permite, entre outras, a definição de dados, restrições, negações, disjunções, regras e expressões quantificadas.

A CycL foi proposta como ferramenta para suporte à criação da ontologia CYC e para representação de conhecimento senso comum. A linguagem é baseada em lógica de predicados de primeira ordem. A construção da base de conhecimento é realizada pela combinação de termos, com o objectivo de criar frases fechadas CycL (independentes de variáveis). Os termos CycL podem ser vectores de caracteres, números, variáveis, termos não atómicos (NAT) e constantes semânticas [121].

Para além das linguagens apresentadas são ainda de referir outros exemplos, tais como, a Loom, a CML, a Classic, a Frame Logic e a ontolingua.

2.3.3 A linguagem OWL e a influência do W3C

No processo de maturação das linguagens de representação, o Consórcio W3C[30], tem desempenhado um papel relevante, enquadrado com o seu objectivo de estimular o desenvolvimento da Internet de segunda geração, a Web Semântica. O consórcio suporta um conjunto de iniciativas que visam, de forma incremental, e sem obrigar a rupturas, atingir uma linguagem de representação de conhecimento. Numa primeira fase, foi responsável por introduzir a norma XML para ultrapassar as limitações do HTML, que se limita a definir processos de apresentação dos dados. O XML permite a descrição da informação com recurso a um conjunto de marcadores, o que permite não perder a semântica associada aos dados. Todavia, o XML possui sérias limitações à descrição de relações entre conceitos,

[30] W3C - World Wide Web (http://www.w3.org/)

conduzindo à introdução da linguagem Resource Description Framework (RDF), que permite a referência entre recursos de forma independente da sua localização.

O passo seguinte, patrocinado pelo W3C, foi a linguagem DARPA previamente apresentada.

Finalmente, surge a linguagem Web Ontology Language (OWL) como resultado da revisão da linguagem DAML+OIL. Esta linguagem, que apresenta uma expressividade semântica superior às anteriores, utiliza classes e relações entre elas para a representação do conhecimento [122], permitindo a interpretação automática de conteúdos. Desenhada com o objectivo de permitir níveis crescentes de compatibilidade, foi estruturada de raiz, para conter três sublinguagens de expressividade incremental: versão OWL Lite, OWL DL e OWL Full [123].

- OWL Lite suporta as necessidades básicas de hierarquia e de restrições simples (e. g., embora permita restrições de cardinalidade, está limitada a 0 e 1). A complexidade formal do OWL Lite é, comparativamente, inferior à complexidade das versões superiores. O objectivo foi a criação de uma linguagem para a qual fosse fácil a criação de uma ferramenta de suporte, estimulando a sua adopção;
- OWL DL suporta todas as características do OWL Lite acrescidas de uma expressividade superior, mantendo, todavia, a completude computacional (i. e., está assegurada a capacidade de determinação de todas as soluções) em tempo finito. O OWL DL integra todos os construtores do OWL, porém são impostas restrições; (e. g., uma classe pode ser subclasse de outras, todavia uma classe não pode ser uma instância de outra classe). A denominação de OWL DL advém da correspondência com a lógica de descrição (*description logic*);
- OWL Full assegura ao utilizador a máxima expressividade e liberdade sintáctica do RDF, sem garantias de ser computacionalmente tratável. A título de exemplo, uma classe definida em OWL Full pode ser tratada, simultaneamente, como uma colecção de indivíduos ou com um só indivíduo. A OWL permite, inclusive, a extensão do vocabulário do OWL e do RDF. A flexibilidade de expressão do OWL Full dificulta a implementação de aplicações que possam interpretar, cabalmente, todas as suas características.

Cada uma das linguagem é uma extensão da predecessora, tanto no que pode ser expresso como no que pode ser determinado e este facto assegura que a validade e correcção numa linguagem garante as mesmas propriedades na seguinte, não sendo todavia assegurado o inverso. A escolha sobre o nível da linguagem depende das necessidades de expressividade dos utilizadores. A escolha do OWL Lite e OWL DL está fortemente ligada aos requisitos dos construtores mais poderosos. A decisão entre o OWL DL e OWL Full está

ligada a requisitos de facilidades de meta-modelação (e. g., definição de classes de classes), todavia não existem, actualmente, implementações do OWL Full.

Uma apresentação mais detalhada do OWL pode ser encontrada no anexo A.3.

2.3.4 Exemplos de Ontologias

Existem diversas ontologias disponíveis, em áreas e graus distintos de conceptualização, representando o estado actual de desenvolvimento da tecnologia. Em [96], sugere-se que uma ontologia pode ser tipificada, dependendo do grau de generalização conceptual, da seguinte forma:

- Ontologia Genérica: Captura o conhecimento sobre conceitos gerais, tais como tempo, espaço e estados em diversos domínios de aplicação;
- Ontologia de Domínio: Captura o conhecimento de um domínio especifico. (e. g., Comércio Electrónico, Espaço, Biologia, Medicina);
- Ontologia de Meta-dados: Captura o conhecimento sobre conceitos descritos em fontes de informação, (e. g., o Dublin Core [124]);
- Ontologia de Representação: Captura o conhecimento para representação, sendo, assim possível a sua utilização para descrição de conhecimento. Um exemplo deste tipo de ontologias é a *Frame Ontology*, que descreve conceitos como *frames*, *slots*, e restrições, permitindo a definição de entidades numa estrutura de *frames*;
- Ontologia de Métodos e tarefas: Captura o processo de raciocínio sobre um domínio.

O WordNet é um exemplo de uma ontologia de grande dimensão que disponibiliza um *thesaurus* para mais de 100 mil termos Ingleses, explicados em linguagem natural. Foi desenvolvida pelo Cognitive Science Laboratory na Universidade de Princeton.
A aproximação proposta tira partido das novas correntes de teorias psico-linguísticas da memória lexical humana. A ontologia foi construída organizando nomes, verbos, adjectivos, advérbios, em conjuntos de sinónimos, descrevendo desta forma o conceito em causa. Em cada categoria, as palavras são ordenadas por conceitos e são descritas relações de sinónimos e antónimos, assim como relações morfológicas, de abstracção e «parte de», entre palavras [125].

Com características distintas, o CYC disponibiliza teorias formais axiomáticas para a representação de conhecimento de senso comum. O projecto iniciou-se 1984 estando, actualmente, a ser aplicado em situações reais. A ontologia contém a representação formalizada de uma vasta quantidade de informação de senso comum: factos, regras, heurísticas de raciocínio sobre objectos e eventos do dia-a-dia. A representação foi realizada com a linguagem CycL. A aproximação utilizada está baseada na utilização de micro-teorias estanques que representam conhecimento de senso comum. As micro-teorias

estão concentradas num detalhe particular de conhecimento, a um nível de abstracção particular [121]. A utilização de micro-teorias possibilita a utilização de mecanismos de inferência local e permite, inclusive, a utilização de asserções contraditórias. A ontologia CYC tem sido utilizada por motores de pesquisa «Hotbot and Lycos» para melhorar os resultados das pesquisas pela análise semântica das perguntas [121].

A TOVE é um exemplo de uma ontologia de tarefa e domínio com o objectivo de facultar um mecanismo de partilha de conhecimento que suporte a integração numa empresa. Foi desenvolvida no decurso do projecto TOVE (Toronto Virtual Enterprise) realizado pela Universidade de Toronto. A ontologia foi utilizada para representar o conhecimento de duas empresas, uma fábrica de computadores e uma companhia de engenharia aeroespacial [126].

A Enterprise Ontology foi criada com o intuito de capturar aspectos de negócio permitindo a sua análise para identificação de opções de operacionalização de acordos. Foi desenvolvida no decurso do projecto Enterprise, uma iniciativa promovida pelo governo Inglês, com o intuito de estimular a utilização de sistemas baseados em conhecimento. O foco do projecto visava a gestão de inovação e a utilização estratégica de tecnologias de informação, a fim de auxiliar a gestão da mudança. Aplicações em casos reais foram realizadas pela IBM, Lloyd, Unilever e AIAI [127].

A XML Common Business Library (xCBL) captura conhecimento para suporte ao comércio electrónico B2B, contemplando a necessária definição de documentos. O protocolo está representado num conjunto de documentos XML e seus componentes, podendo ser utilizado para criação de novos documentos. O objectivo foi criar um suporte à comunicação de dados comerciais via Internet, entre entidades heterogéneas, de forma a permitir um acesso global a clientes, fornecedores, revendedores. A versão xCBL 4.0 passou a estar disponível unicamente em linguagem esquemas XSDL, ao contrário das versões anteriores que estavam descritas em diversos formatos. Os documentos existentes permitem a integração de diversos tipos de aplicações, (e. g., integração de ERP) bem como o suporte a diversas normas, entre outras, RosettaNet ou OBI. Esta ontologia é resultado da colaboração entre a Commerce One, organismos de normalização XML, diversas empresas envolvidas no comércio electrónico e, também, fabricantes e vendedores de sistemas e equipamento informático. A ontologia foi criada tendo em conta diversas normas existentes, entre elas EDI – Electronic Data Interchange, RosettaNet, e o OBI – Open Buying on Internet [128].

O Commerce XML (cXML), em Fevereiro de 1999, assumiu a ruptura com o EDI, contornando uma das limitações que mais dificultava a sua aplicação ao comércio electrónico, a necessidade de estabelecer ligações ponto-a-ponto. Baseado em tecnologia Web está descrito em DTDs XML que contemplam a especificação de um conjunto alargado

de documentos necessários à implementação do processo de negócio, desde o início da negociação até ao suporte pós-venda [129].

O DCMI – Dublin Core Metadata Initiative é uma organização que procura, desde 1995, a disseminação e a adopção de normas para a interoperabilidade de dados, promovendo, portanto, o desenvolvimento de meta-dados de vocabulários especializados para a descrição de recursos, o que permite, naturalmente, a sua pesquisa mais eficiente. DCMES Dublin Core Metadata Element Set, (a primeira norma de meta-dados produzida pelo grupo), contempla a definição semântica de quinze elementos básicos independentes de um domínio específico. A título de exemplo, entre os elementos-base estão: os elementos «relation» e a «source» utilizados para indicar uma ligação com recursos de qualquer tipo; os elementos «Creator», «Contributor» e «Publisher» que são elementos de relação entre recurso e o seu produtor; o elemento «Coverage» que permite a identificação espaço-temporal do recurso. Actualmente existem diversas aplicações, em especial em organizações de ensino, bibliotecas, instituições governamentais, sectores de investigação científica, ou seja, em geral, entidades que procuram a criação de sítios que sejam mais facilmente pesquisáveis. O âmbito do protocolo é a definição de um vocabulário de propriedades-base que permita a descrição de informação de qualquer tipo de recurso, (independentemente do formato, áreas de especialização ou origem cultural), mesmo que não esteja disponível na Web [130].

Assumindo que existe um conjunto de ontologias que descrevem diversos domínios e, dentro deles, conceitos de forma modular, a criação de uma ontologia global é uma área activa de investigação, que aborda os seguintes assuntos:

- a criação de ontologias reutilizáveis, o que passa pela criação de ontologias com elevada coerência interna e de pequena abrangência, sem interligações e dependências inter-domínio;
- a existência de metodologias de combinação de ontologias, com mecanismos de selecção e concordância de conceitos. Sendo exemplos, o Ontolingua server [131], que disponibiliza um conjunto de operações para combinação de ontologias, inclusão, restrição, etc. O sistema SENSUS [132] que procura disponibilizar um mecanismo de construção de uma ontologia de domínio, a partir de uma ontologia de senso comum; ou, ainda, o projecto SKC (Scalable Knowledge Composition) [133] que procura a combinação de ontologias através do recurso a operadores algébricos;
- a maturação das linguagens formais de descrição de ontologias.

2.3.5 Dados, informação, conhecimento

Tendo em conta a existência de diferentes definições sobre o significado de dados, informação e conhecimento, adopta-se nesta dissertação, a proposta de [134].

Entende-se por dados os sinais não interpretados, usualmente existentes em elevadas quantidades e presentes nos computadores em diversos formatos, entre outros: *bit, bytes*, caracteres, números, e palavras que são manipuladas mecanicamente em grandes quantidades (e. g., As cores verde, amarelo, e encarnado de um semáforo).

Entende-se por informação o significado a associar aos dados, (no exemplo do semáforo, a necessidade de parar na presença da cor vermelha).

Entende-se por conhecimento o conjunto de dados e informação, utilizados na prática com o objectivo de realizar tarefas e criar nova informação. O conhecimento acrescenta dois novos conceitos: um sentido de objectivo, uma vez que o conhecimento é o mecanismo intelectual utilizado para atingir um objectivo; e a capacidade generativa, tendo em conta que uma das principais funções do conhecimento é gerar informação.

Convém referir que a interpretação depende do actor. O que é conhecimento para um actor pode ser interpretado como sinal por outro, bastando, para tal, que não exista capacidade de inferência.

2.4 Engenharia informática suportada em agentes

Há mais de 30 anos que a Inteligência Artificial (IA) procura resposta para problemas que se caracterizam pela sua grande complexidade (número, variedade e natureza das tarefas) e por não apresentarem uma «solução algorítmica», apesar de existir conhecimento que permite a sua modelação através de comportamentos semelhantes a um ser inteligente (autonomia, aprendizagem, conhecimento, etc.). No fundo, a IA procura construir (e aprender a construir) programas que, segundo critérios definidos, exibem um comportamento coerente na realização de uma dada tarefa.

Em paralelo, o crescimento de recursos distribuídos e interligados por redes de dados, obrigam à definição de novas metodologias de desenvolvimento de sistemas informáticos, pois as técnicas tradicionais lidam de forma deficiente com problemas de distribuição e interoperabilidade.

Neste ambiente, no início dos anos noventa, surge o promissor paradigma de programação por agentes, concebido especificamente para ambientes distribuídos com vista a explorar as capacidades de comunicação. Apesar de poder conter um único agente, e. g., os assistentes pessoais, o maior potencial desta aproximação reside, inequivocamente, nos sistemas multiagentes. A definição de agente está longe de ser consensual, sendo possível identificar múltiplas descrições que nem sempre estão em concordância e dependem,

fortemente, da perspectiva e da utilização. Parunak afirma, provocante, que existem quase tantas definições para agente quanto investigadores na área [135]. De acordo com Stuart Russel e Peter Norvig, um agente percepciona o ambiente utilizando sensores e interage através de actuadores. Esta definição depende de três factores: os sensores, os actuadores e o ambiente [136]. Noutro contexto, Michael Genesereth defende que a característica essencial é a capacidade de comunicar através de uma linguagem [137]. Pattie Mae define agentes enfatizando a autonomia [138] que permite a realização de objectivos em ambientes complexos. Michael Coen acrescenta à autonomia, a robustez, e a capacidade de recolher, negociar e coordenar a informação entre agentes [139]. A IBM define agentes inteligentes como entidades aplicacionais que desempenham um conjunto de operações em representação de um utilizador, empregando uma representação de conhecimento sobre os desejos e objectivos do utilizador. Van Parunak define «agente» como um entidade de *software* que encapsula informação, código, o seu próprio controlo de execução (fazendo dele um objecto activo) e a capacidade de se activar sem evocação externa [135]. Outra proposta [140], define agentes através das suas capacidades, em especial:

- Autonomia: Capacidade de autonomia de raciocínio, em controlo das suas acções e do seu estado interno, e de operação sem a intervenção externa, resultando em pró-actividade de comportamento guiado por objectivos;
- Reactividade: Sensibilidade ao ambiente e capacidade de resposta às alterações para satisfazer objectivos, em tempo real;
- Adaptabilidade e aprendizagem: Capacidade de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência;
- Socialização: Capacidade de interacção com outros agentes e meio ambiente, através de actividades sociais (cooperação, negociação) com o recurso a uma linguagem de comunicação.

A autonomia é, provavelmente, a única capacidade genericamente aceite; as restantes levantam celeuma, incluindo a aprendizagem. Michael Georgeff, responsável por um dos sistemas de multiagentes de maior sucesso na área do controlo de tráfego aéreo, descreve os seus agentes como entidades desprovidas de capacidade de aprendizagem, o que assegura a possibilidade da sua aplicação. Caso contrário, não seriam aceites na execução de tarefas críticas e de tão elevada responsabilidade.

É essencial fazer a distinção entre agentes e objectos, evitando uma confusão muito comum sobretudo na comunidade familiarizada com a programação orientada por objectos. Apesar de existirem características semelhantes, (tais como o encapsulamento de código e estados, capacidade de execução de acções e métodos dependendo dos estados internos, e comunicação por mensagens), as distinções entre paradigmas são significativas.

A maior distinção reside na capacidade assumida pelos agentes de efectuarem escolhas sobre as suas acções e interacções. Um agente não pode ser directamente invocado como um objecto, recaindo sobre si a decisão de atender um pedido, o que é uma diferença importante, pois nenhum objecto se recusa a executar um método. Um objecto não exibe controlo sobre o seu comportamento. Uma outra diferença encontra-se na noção de flexibilidade (reacção, pró-actividade e socialização) que está ausente, no paradigma da programação por objectos. Finalmente, assume-se que um agente possui controlo sobre a sua execução, i. e., é um programa ou pelo menos um processo que executa, de forma autónoma, situação que não acontece com os objectos tradicionais, que são passivos; i. e., estão disponíveis para serem invocados.

Tendo em consideração estas distinções é, todavia, possível especificar objectos que se assemelhem a agentes, em especial objectos activos, dotados de controlo sobre a sua execução. Porém, neste caso, existe um afastamento da visão tradicional orientada por objectos.

Ao longo desta dissertação, assume-se como **sistema baseado em agentes**, um sistema em que a principal abstracção é o agente, e para **agente** adopta-se a proposta de [141], onde é definido como um programa de computador capaz de executar uma acção autónoma num ambiente dinâmico, imprevisível e aberto. Na definição anterior, **autonomia** é um conceito mais difícil de precisar, tendo-se adoptado a definição apresentada em [140], ou seja, a capacidade de um sistema actuar sem a directa intervenção de outros agentes (incluindo humanos), tendo o controlo das suas acções e do seu estado interno. Os **ambientes dinâmicos, imprevisíveis e abertos** são caracterizados por uma elevada actividade (não sendo possível conhecer, *a priori* os seus componentes), verificando-se constantemente mudanças, com elevada heterogeneidade de componentes (produzidos por diferentes entidades, com metodologias, técnicas e ferramentas distintas).

Nos próximos anos é expectável que a tecnologia de agentes desempenhe um papel relevante em diversas áreas, especialmente em ambientes inteligentes, onde pode auxiliar a ultrapassar as limitações actualmente impostas, permitindo a obtenção da computação ubíqua. Todavia, apesar de todas as potencialidade, a maioria das aplicações não são, nem devem ser, desenvolvidas com uma abordagem orientada por agentes. A utilização de agentes não é adequada a situações em que é necessário manter um controlo global do sistema, onde respostas em tempo real têm que ser asseguradas e não são admissíveis soluções de bloqueio, em especial se for relevante manter uma visão global do sistema.

A tecnologia de agentes é aplicável a ambientes complexos, dinâmicos e abertos, onde a interacção entre sistemas heterogéneos, com fronteiras pouco definidas, é uma realidade. Em particular se for determinante a presença de autonomia, para a obtenção de uma

resposta dinâmica às alterações de circunstâncias, com vista à realização dos objectivos propostos (que podem ser sobrepostos e concorrentes).

São muitas as razões que justificam as expectativas no paradigma de agentes, para a abertura de novas soluções para a engenharia informática. Entre outras, por ser: *i)* uma metáfora natural para lidar com sistemas complexos, consensualmente caracterizados por um elevado número de interacções; *ii)* adequados à produção de sistemas distribuídos a nível de dados e controlo; *iii)* aplicáveis à integração de sistemas legados através da sua agentificação e, finalmente; *iv)* capazes de responderem às necessidades dos sistemas abertos, com elevada dinâmica, de difícil especificação, com grandes exigências de flexibilidade e autonomia na tomada de decisão [142, 143].

Neste enquadramento, os sistemas poderão ser profundamente repensados numa perspectiva de parceria cooperante com os objectivos do utilizador, ultrapassando, assim, o papel até agora reservado de meros actores passivos. As maiores expectativas estão centradas na optimização da exploração de infra-estruturas, nos recursos distribuídos, na consolidação dos processos de automação de filtragem e recolha de informação, bem como na aplicação às áreas da bio-informática, monitoração e controlo.

Na produção de sistemas baseados no paradigma de agentes, a unidade principal de encapsulamento é o agente. O desenvolvimento passa, assim, a ser suportado na construção de agentes, e de comunidades de agentes, com o recurso a primitivas-base de elevada abstracção, entre outras: objectivos, escolhas, comportamentos, mensagens (e. g., pedidos, ofertas, recusas, permissões, etc.). Neste sentido, é necessário elevar os níveis de abstracção e construir os sistemas sobre novas unidades de re-encapsulamento, à semelhança do que se verificou na transição da programação imperativa, para a programação orientada por objecto, em que se passou, da unidade-base de abstracção função, para o objecto [144].

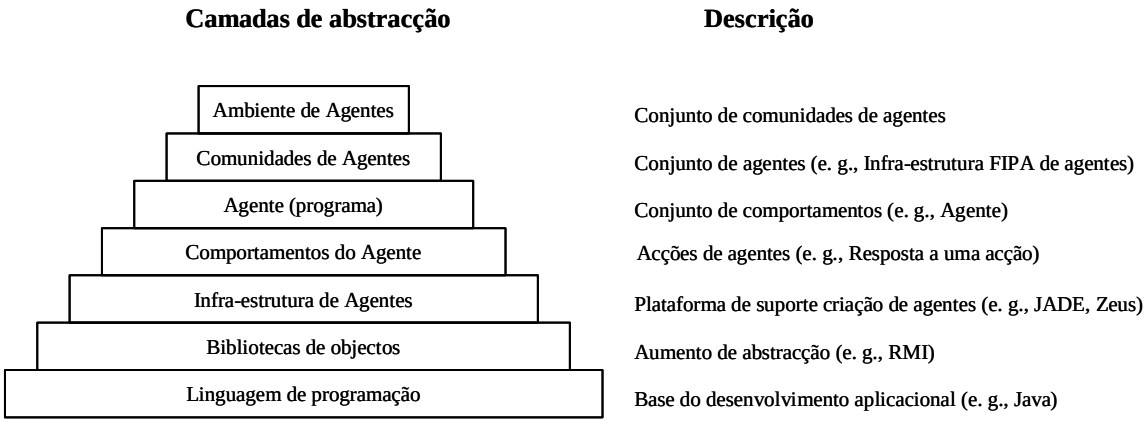


Figura 9 – Camadas de abstracção da programação de agentes.
(figura adaptada de [144]**)**

A Figura 9 apresenta as camadas de construção dos sistemas à luz do paradigma de agentes, desde a linguagem de programação (no mais baixo nível de abstracção), até ao ambiente de agentes (o mais alto nível). A biblioteca de objectos disponibiliza as capacidades fundamentais que permitem a um agente participar numa sociedade. A infra-estrutura de agentes, disponibiliza um motor de comportamentos construídos por objectos ou regras, que permitem a definição global do comportamento do agente, baseado numa filosofia multitarefa. A camada seguinte permite utilizar o agente como unidade de encapsulamento-base. As duas camadas seguintes, representam agregações de agentes: a Comunidade de Agentes, que define um conjunto de agentes que partilha objectivos comuns (exigindo consequentemente fortes interacções); e o Ambiente de Agentes, (concepção mais abstracta) que permite agregar comunidades. A camada que distingue a programação orientada por objectos, da programação orientada por agentes, é a «Infra-estrutura de agentes».

A adopção efectiva do paradigma de agentes está, naturalmente, condicionada ao amadurecimento das suas técnicas e metodologias, permitindo demonstrar a sua adequação, eficácia e robustez. As áreas de maior trabalho intensivo estão representadas na Figura 10, onde se procurou sistematizar as etapas e abordagens adoptadas.

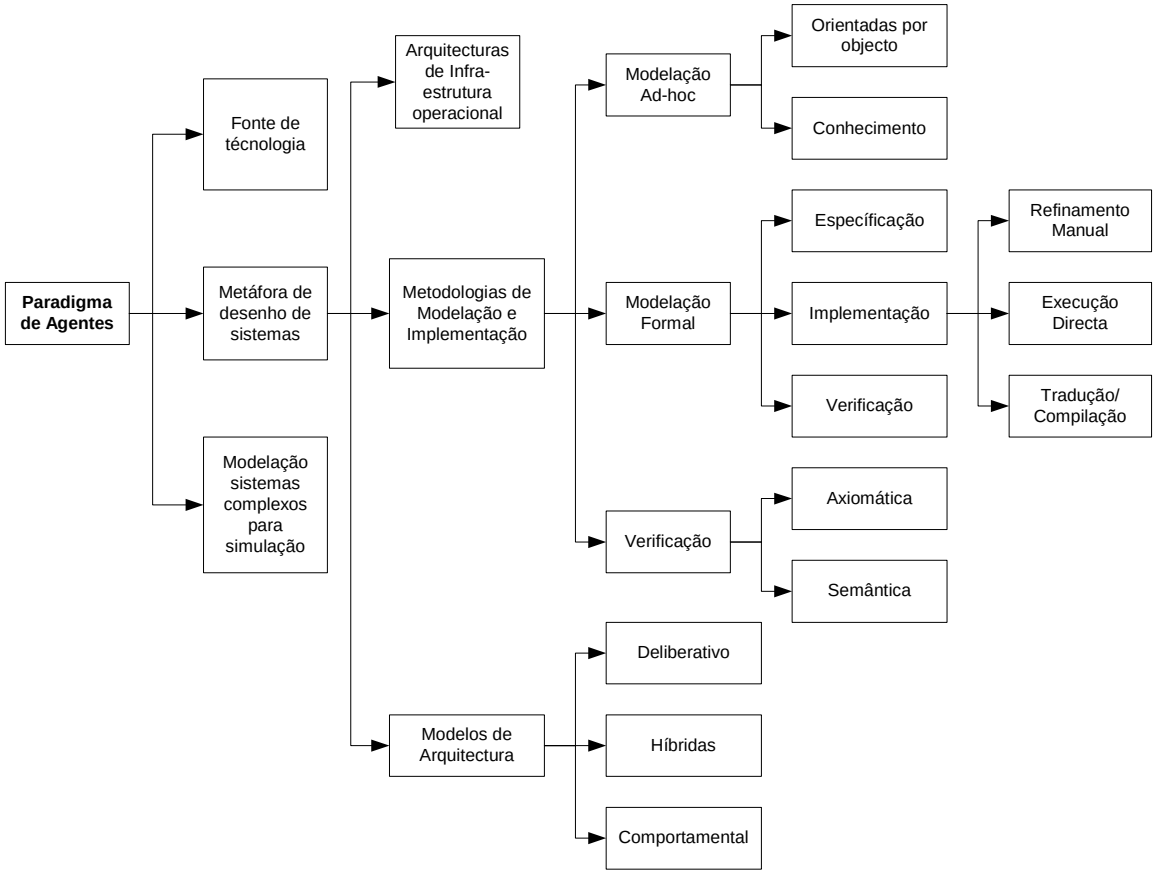


Figura 10 – Áreas de trabalho intensivo de maturação do Paradigma de Agentes e as suas abordagens mais comuns

As principais áreas são: *i)* uma metáfora de desenho de sistemas; *ii)* uma fonte de tecnologia para a construção dos referidos sistemas, ou ainda; *iii)* um processo de modelação de sistemas reais complexos para simulações, tais como os biológicos, os económicos e os ambientais.

2.4.1 Uma nova metáfora para o desenho de sistemas

O ímpeto inicial dos agentes deveu-se à modelação de ambientes complexos e distribuídos. Isto aconteceu devido ao facto do paradigma lidar, de forma elegante e versátil, com a heterogeneidade e a independência, resolvendo, com naturalidade, as interacções entre componentes, sendo estas características difíceis de capturar por técnicas de modelação tradicionais. Todavia, o paradigma ainda não atingiu um estado de maturação suficiente, não existindo metodologias genericamente aceites para o desenvolvimento de sistemas de agentes. O trabalho a realizar é vasto, e abrange: *i)* a consolidação das metodologias de modelação de sistemas; *ii)* das arquitecturas de desenvolvimento; e *iii)* das infra-estruturas operacionais.

Metodologias de desenho de sistemas complexos

A área de desenho de sistemas pode ser dividida em **metodologias de modelação**, que permitem auxiliar o desenho dos sistemas (modelação, verificação e suporte ao desenvolvimento) e em **modelos de arquitectura conceptual**.

As **metodologias de modelação** mais comuns, embora sedimentadas em princípios de base, não dispõem de técnicas formais. Podendo ser divididas em metodologias baseadas na aproximação da programação orientada por objecto (adaptação ou extensão) ou metodologias da área da engenharia do conhecimento. Entre outras, destacam-se pela sua repercussão, a metodologia GAIA de Wooldridge [145], AAII de Kinny[146], Cassiopeia de Collinot e Drogoul [147], DESIRE de Treur [148] ou enquadramento Z, de d'Inverno e Luck [149, 150].

Outro tipo de metodologia de modelação, recorre à definição de **métodos formais** com vista: *i)* a especificação dos sistemas; *ii)* a implementação directa de sistemas; e *iii)* a verificação de sistemas [151].

Os **métodos formais da especificação de sistemas** procuram identificar os requisitos e propriedades que especifiquem agentes, e sistemas. A aproximação predominante aborda os agentes como entidades intencionais, atribuindo-lhes estados mentais como «crenças», «desejos», «intenções», «planos», e «objectivos» fazendo uso de lógica modal temporal [152, 153]. Destacam-se, de entre eles, o método proposto por Rao-Georgeff, *belief-desire-intention model* (BDI) [154] e a teoria da intenção de Cohen-Levesque [155]. As atenções têm estado centradas no desenvolvimento de sistemas formais que definem as

inter-relações das diferentes atitudes apresentadas por um agente, porém são ainda poucas as aplicações a sistemas reais.

Os **métodos formais de implementação** procuram identificar processos que auxiliem o desenvolvimento dos sistemas, em conformidade com as especificações, fazendo a ponte entre a especificação abstracta e o modelo computacional concreto. Existem, essencialmente, três abordagens a esta temática:

i) O refinamento manual das especificações (seguindo uma metodologia que auxilie a programação) é a aproximação mais comum, sendo partilhada pela maioria dos paradigmas. Após um processo de especificação abstracta, cabe ao programador a tradução das especificações em código-fonte, utilizando, para tal, uma metodologia que procura evitar os erros. Todavia, tendo em conta o factor humano, esta opção está longe de ser imune ao erro, pelo que as hipóteses seguintes são atractivas pelo seu cariz automático;

ii) A execução directa ou visualização das especificações abstractas pelo sistema METATEM é um exemplo de execução directa, em que os agentes são programados por especificação em lógica temporal [156];

iii) A tradução ou compilação das especificações para geração do modelo computacional de forma directa [142]. Um dos exemplos mais conhecidos de compilação directa é o «situated automata» de Rosenschein and Kaelbling [157].

A distinção essencial entre as duas últimas aproximações reside na separação entre a interpretação e a compilação das especificações. As duas aproximações são difíceis de obter devido à distância existente entre a especificação abstracta e o modelo computacional concreto. As aplicações existentes limitam-se a especificações abstractas extremamente simplificadas e longe das implementações reais.

Finalmente, os **métodos formais de verificação** focam a sua atenção na validação dos modelos computacionais concretos, procurando processos de verificação automática da sua integridade e em conformidade com as especificações abstractas. A importância destes métodos está directamente relacionada com o aumento da intervenção humana. Quanto menor for o grau de automatização dos processo de geração dos modelos concretos, maior é o risco de erro e, consequentemente, maior a necessidade de existência de processos automáticos de validação. Existem duas aproximações clássicas:

i) a axiomática: a verificação axiomática procura derivar uma teoria lógica que represente um modelo concreto procurando de seguida verificar a equivalência com a especificação original. No fundo, a verificação fica reduzida a um problema de prova de equivalências. Os trabalhos mais relevantes estão relacionados com a tradução das instruções dos programas em lógica temporal linear, iniciados por

Manna e Pnueli [158]. Todavia, a dificuldade associada aos problemas de prova, aumentada com a introdução do vector temporal, têm impossibilitado a utilização deste método a casos reais;

ii) a semântica: (por vezes apelidada por verificação de modelo), é caracterizada pela abstracção do modelo concreto em lógica temporal e pela verificação da igualdade de reacções dos modelos a situações idênticas. Esta aproximação, apesar de apresentar um menor grau de dificuldade, em comparação com a axiomática, apresenta uma elevada complexidade computacional, tendo em consideração a necessidade de verificação de todos os casos.

A maioria das técnicas e metodologias apresentadas, está longe de ter atingido uma consistência que permita a sua utilização em casos reais, com a agravante de serem poucos os consensos, começando pela dificuldade em definir quais os conceitos a suportar. Todavia, as aproximações que adaptaram as metodologias tradicionais de modelação de dados por objectos, são as de maior sucesso, em especial na criação de extensões ao UML. Existe, contudo, um longo percurso a ser trilhado para se obterem soluções robustas, especialmente porque alguns conceitos e notações utilizados no UML não se adequam ao desenvolvimento de sistemas de agentes.

Arquitecturas de desenvolvimento

Para além das ferramentas de desenho é, igualmente, necessário consolidar os modelos que sustentam as arquitecturas de desenvolvimento. Existem diversas aproximações à criação de agentes, desde as baseadas em agentes meramente comportamentais (ou reactivos) que operam numa filosofia pura de estímulo-resposta, até aos agentes mais deliberativos que raciocinam sobre as suas acções, como sejam os agentes BDI [141].

O modelo comportamental foi proposto, inicialmente, por Brooks [159] com a «Subsumption Architecture», onde se defendia que a comunidade científica não devia tentar criar entidades com inteligência de nível humano, mas sim seguir um processo semelhante à evolução natural. A experiência adquirida com a criação de sistemas de inteligência primária permitiria, então, dar novos passos na direcção da produção de sistemas mais complexos. A arquitectura orientada ao comportamento, onde as acções têm um resultado comportamental visível, é desenhada para permitir as funcionalidades observadas nos níveis de vida inferior (nomeadamente nos insectos). Esta arquitectura disponibiliza um conjunto-base de funcionalidades e os sistemas são implementados tirando partido de máquinas sem controlo central, sem partilha de representações, com baixas taxas de mudança e de comunicação. Cada agente é independente e autónomo mantendo uma representação do estado global minimalista, podendo ser instanciado com a utilização de

microprocessadores económicos, efectuando algoritmos com baixos requisitos de memória e dotados de sensores simples.

No outro extremo, encontram-se as arquitecturas de construção de agentes deliberativos que permitem a criação de sistemas orientados aos objectivos. Sendo o resultado de mais de vinte anos de estudos, do campo da inteligência artificial, são muito comuns e baseiam-se na existência: *i)* de crenças, para representar o saber do agente; *ii)* de desejos/objectivos para representar o que o agente tenta obter; *iii)* de intenções para descrever os objectivos correntes e, finalmente; *iv)* de planos que permitem descrever combinações de actos para gerar as acções em resposta a eventos, e que são utilizados pelo agente para definir as suas intenções. Quando um evento ocorre, o agente pesquisa planos relevantes de resposta, faz a análise de cada opção em função da sua situação e executa o mais apropriado. Em paralelo, o agente desencadeia acções de raciocínio para decidir: *i)* quais os objectivos (ou, alternativamente, os eventos) aos quais tem que reagir; *ii)* como perseguir os seus objectivos e; *ii)* quando suspender, abandonar ou alterar os seus objectivos. Estas arquitecturas, prevêem a alteração da atenção atribuída à reacção ou à deliberação dos comportamentos, (permitindo concentrar as atenções nas reacções a eventos) ou, pelo contrário, à concretização de planos [160]. As plataformas de sucesso serão, provavelmente as híbridas, uma vez que equilibram melhor as capacidades de reacção e deliberação.

Existem, actualmente, mais de uma centena de plataformas de desenvolvimento de agentes [161]. Porém, entre as mais conhecidas e em conformidade com a FIPA, podem ser realçadas a FIPAOS, a JADE e a Zeus, que partilham o comportamento como filosofia comum, disponibilizando ao programador diversos mecanismos de agentes, um sistema de execução de processos, uma interface de mensagens ACL, um motor de controlo de comportamentos, etc.

Arquitecturas de Infra-estrutura operacional

A área de infra-estruturas de agentes está relacionada com o suporte operacional para os sistemas de agentes e tecnologia *middleware*. Esta tecnologia procura uniformizar os sistemas de redes e a heterogeneidade dos ambientes, agilizando funcionalidades e disponibilizando uma interface única ao utilizador. Os objectivos primordiais são a capacidade de anúncio de novas presenças na rede, a descoberta de parceiros (nas vizinhanças e remotos), a discrição de capacidades, a autoconfiguração e a interoperabilidade. Estas propriedades são difíceis de obter, pela inexistência de protocolos de comunicação normalizados e pela elevada especialização dos equipamentos. Actualmente, as infra-estruturas operacionais mais utilizadas são a Jini (www.jini.org) da Sun Microsystems, a UpnP (www.upnp.org), suportada inicialmente pela Microsoft e a Salutation (www.salutation.org) que definem protocolos de procura e registo, permitindo a criação de sistemas de descoberta dinâmica [162]. Nenhuma das plataformas parece

possuir as características necessárias que permitam a sua adopção universal, sendo difícil antever qual prevalecerá, pelo que é de esperar o surgimento de muitas alterações e novas propostas, nos próximos anos.

2.4.2 Uma fonte de tecnologia para a construção de sistemas

O paradigma de agentes pode, igualmente, ser encarado como receptor e como fonte de desenvolvimento tecnológico em diversas áreas de investigação aplicada e teórica. Entre outras, são de realçar as ontologias, a negociação, os mecanismos de aprendizagem, as linguagens de comunicação, os mecanismos de coordenação e os sistemas de decisão e planeamento. Todavia, a sua influência ultrapassa os limites das tecnologias de informação, expandindo-se às áreas das humanidades e ciências sociais. Entre as interacções teóricas e aplicadas mais relevantes, destacam-se:

- o planeamento, baseado em sistemas multiagentes onde se procura realizar a decomposição de problemas, o que permite a sua execução distribuída [163-165];
- as linguagens de comunicação de agentes que permitem a comunicação inter-agente, sendo o KQML [166, 167] e FIPA ACL [168] as duas mais utilizadas, apesar das suas limitações, principalmente no que concerne à comunicação humano-agente;
- a gestão de conhecimento, onde é crucial a criação de metodologias e ferramentas de suporte à aquisição, manutenção, partilha e avaliação de conhecimento, que permita a sua exploração nos sistemas multiagentes [169, 170];
- a aprendizagem, que permite aos agentes aprenderem e adaptarem-se ao ambiente em que estão integrados, uma vez que, na maioria dos casos em que os agentes são utilizados, não é possível prever todas as situações potenciais. Neste campo, a aplicação da tecnologia de agentes tem um especial sucesso nas vertentes de personalização e recolha de informação, apresentando resultados muito promissores nas áreas da robótica e das telecomunicações [171-174].

Actualmente, as principais áreas de aplicação são os sistemas de telecomunicações: controlo aéreo, gestão de tráfego e transportes, filtragem e recolha de informação, comércio electrónico, gestão de transacções (financeiras e de bolsa), gestão de processos de negócio, jogos, aplicações militares, medicina, controlo de processos e manufactura.

Num esforço de sistematização, as aplicações podem ser divididas em sistemas multiagentes (em que as tarefas e decisões são efectuadas colectivamente), ou sistemas de agente único (que apesar de poderem ter que interagir com terceiros, tomam as decisões de forma individual). Um sistema multiagentes distingue-se pela utilização de agentes que interagem, potencialmente, cooperando, incluindo ou não humanos, e onde podem surgir conflitos de interesse. Nos sistema de agente único, enquadram-se os assistentes

especializados; (e. g., sistemas de recolha de informação, assistentes de ajuda *on-line* ou de execução de transacções).

2.4.3 Um processo de modelação de sistemas reais complexos

Uma área de aplicação extremamente promissora é a modelação de sistemas complexos para realização de simulações. Nesta perspectiva, os agentes são utilizados para modelar sistemas complexos, por exemplo: a economia, o tempo, o tráfego, populações biológicas, centros de chamadas telefónicas, sistemas de gestão de redes de dados e de comunicação, eventos naturais, entre outros [175-179]. A modelação é realizada tendo por unidade-base de representação o agente que, após definido, possibilita a criação e simulação de cenários de forma a permitir um melhor entendimento dos sistemas complexos.

2.4.4 Que futuro?

A investigação e o desenvolvimento, baseados em agentes, encontram-se numa fase embrionária, sendo necessária, para a sua consolidação, a maturação de processos e de tecnologias. O estado actual de desenvolvimento está tipicamente centrado na utilização de soluções avulsas, sobre plataformas fechadas, com protocolos de comunicação predefinidos, em ambientes controlados e dificilmente escaláveis. Apesar dos esforços realizados na definição de metodologias (de desenho e de desenvolvimento) e de protocolos de comunicação, a maioria das implementações são realizadas de forma *ad-hoc*, resultando em meros protótipos de demonstração. Esta situação é limitativa, cingindo os sistemas a meros protótipos demonstrativos, dificilmente escaláveis para cenários reais.

Todavia, é expectável que, uma vez ultrapassada esta fase inicial conducente à validação e demonstração deste novo paradigma, se entre numa nova fase de consolidação das metodologias de desenho específicas (tais como GAIA), com a utilização de linguagens de comunicação semiestruturadas (tais como FIPA ACL) e com uma escalabilidade previsível, em ambientes predefinidos.

A fase seguinte, será atingida com o recurso à utilização de protocolos e de linguagens normalizados e com a utilização generalizada de metodologias de desenho específicas para agentes, em sistemas de agentes abertos, e/ou, em domínios específicos (tais como bio-informática, comércio electrónico, entre outros). Nesta fase, as limitações de escalabilidade de agentes, em número e tipo, deixarão de ser uma barreira, iniciando-se a intercomunicação de agentes entre domínios distintos.

Finalmente, entrar-se-á numa fase de obtenção de sistemas abertos e escaláveis, que operarão com agentes capazes de adquirir capacidades de comunicação e de compreensão do novo domínio, após a sua activação em novos ambientes.

Esta antevisão, está suportada pelas projecções realizadas pelo Agent Link Network of Excellence for Agent-Based Computing, e pelo projecto Agentcities, sendo corroborada por um vastíssimo conjunto de entidades e empresas [141]. A sua concretização está condicionada a que durante a próxima década se:

- atinja uma maturação tecnológica, semelhante à actualmente existente noutras áreas da informática, sendo imperativa a consolidação de metodologias de desenvolvimento aplicacional para sistemas baseados em agentes;
- adoptem normas que possibilitem a obtenção de sistemas facilmente interoperáveis e abertos, que viabilizem a criação de agentes, que uma vez activados num novo ambiente, iniciem uma conversação que conduza à sua adaptação e contextualização;
- crie uma infra-estrutura semântica para a comunidade de agentes que permita a descrição sintáctica e semântica da informação a um nível de abstracção elevado. A criação de ontologias, dicionários, ou sistemas de representação de conhecimento desempenharam aqui um papel primordial;
- produzam mecanismos de organização dinâmica, que permitam gerir um conjunto de agentes heterogéneos de origens distintas, onde será necessário lidar com processos de delegação, coordenação, obtenção de confiança, recuperação de falhas e alterações ambientais, entre outros;
- dotem os agentes com mecanismos de aprendizagem que capacitem a assimilação das necessidades dos utilizadores e a adaptação do seu comportamento a alterações ambientais, com vista a melhorar, não só o seu desempenho, como o desempenho de todo o sistema;
- assegure a confiança e fiabilidade dos agentes, permitindo uma efectiva delegação de tarefas por parte do utilizador. Neste sentido é necessário garantir que os agentes são seguros e robustos (não estando em perigo a informação a seu cargo ou a realização da tarefa), e que defendam os interesses do utilizador, respeitando, estritamente, um conjunto de características: a não repudiação, a não realização de comportamentos lesivos à comunidade e o respeito de contratos e de regras gerais.

O principal desafio reside na demonstração das qualidades dos agentes, que representam um valor acrescentado no que respeita aos métodos tradicionais e, ao mesmo tempo, na identificação de quais os novos conceitos e processos a adoptar, que permitirão melhorar os sistemas de agentes.

Acresce ao desafio da maturação das metodologias, a necessidade de maturação dos ambientes de trabalho. É necessário ter sempre presente a complexidade dos sistemas abertos, fortemente relacionada com a indefinição e a imprevisibilidades destes ambientes.

A actual incapacidade da completa especificação destes sistemas, garante um desafio acrescido: o seu estudo aprofundado. É necessário um melhor entendimento sobre como criar, de forma segura e previsível, sistemas dinâmicos com quantidades massivas de agentes dotados de elevado grau de autonomia. Estes ambientes são propícios ao surgimento de comportamentos instáveis, caóticos, de realimentação e são extremamente vulneráveis a acções maliciosas, tais como vírus, agentes destrutivos, etc.

Todavia, as aplicações potenciais de agentes são reconhecidas pelos principais fabricantes mundiais. A IBM considerou a tecnologia de agentes como capaz de acrescentar valor, tendo, inclusive, desenvolvido diversas plataformas de agentes, como seja o sistema «Aglets mobile» [180] e instituiu, mais recentemente, um programa interno de «Autonomic Computing»[181]. A Hewlett Packard esteve, igualmente, na primeira vaga com o ambiente NewWave [182]. Actualmente, o principal esforço comercial visa a criação de aplicações muito específicas, envolvendo fabricantes como a Microsoft e as principais operadoras de telecomunicações, entre outros.

2.4.5 Normalização

A evolução do paradigma de programação baseada em agentes, está muito condicionada ao surgimento de «normas de facto» ou ao esforço coordenado de normalização, que viabilizarão a sua consolidação e utilização em aplicações reais. Nos últimos anos foram diversas as organizações e grupos de investigação que abordaram o assunto, procurando definir normas que permitissem assegurar a interoperabilidade e a reutilização dos sistemas. O Knowledge-Sharing Effort (apoiado pela ARPA, AFOSR[31] e o NRI[32]) foi criado para desenvolver uma infra-estrutura tecnológica e convenções, de modo a suportar a partilha de conhecimento entre sistemas e, assim, assegurar a preservação e partilha de bases de conhecimento, com o objectivo de permitir a sua reutilização. Ao ser criado, estabelece como uma das quatro áreas fundamentais, a definição de protocolos de comunicação entre modelos e sistemas ou bases de conhecimento. Esta iniciativa contribuiu para a definição do KIF e do KQML que se viriam a tornar normas de comunicação, com enorme utilidade na comunicação entre agentes [183]. Em paralelo, no OMG[33] foi criado o grupo de trabalho «Agent Platform Special Interest Group» com a missão específica de estender a arquitectura de gestão de objectos, a fim de melhor suportar a tecnologia de agentes, identificar e recomendar novas especificações e extensões, promover a modelação normalizada de agentes e o desenvolvimento de aplicações suportadas em sistemas de agentes [184].

[31] AFOSR – Air Force Office of Scientific Research (www.afosr.af.mil).

[32] NRI – National Research Initiative.

[33] OMG – Object Management Group (www.omg.org).

Porém, em 1996 surge a FIPA – Foundation for the Intelligent Physical Agents, organização sem fins lucrativos registada em Genebra, com o objectivo de produzir normas aplicacionais para sistemas de Agentes e Multiagentes heterogéneos e interactivos [185]. A declaração de missão da FIPA é, na sua essência, a «promoção de tecnologias e especificações de interoperabilidade de modo a facilitar o trabalho cooperativo entre sistemas de agentes inteligentes nas arquitecturas comerciais e industriais»[34].

O seu surgimento conduziu ao desvanecimento das iniciativas paralelas, e à concentração dos esforços da comunidade industrial e científica, permitindo o seu actual papel de destaque. As motivações para a sua criação estavam ligadas ao facto de existirem, à data da sua criação, mais de sessenta plataformas de desenvolvimento de Agentes, na maioria dos casos soluções fechadas, o que conduzia a uma inevitável impossibilidade de interoperabilidade e consequente portabilidade de código [186].

Para atingir os fins enunciados, a FIPA desenvolve, actualmente, diversos esforços de especificação abrangendo entre outros: linguagens de comunicação de mensagens, protocolos de interacção e gestão de agentes e arquitecturas de comunicação [187]. O âmbito das especificações limita-se às interfaces que têm de ser implementadas, não existindo preocupação na definição ou na descrição de como devem ser desenvolvidos os sistemas internamente. As especificações mais relevantes são: *i)* a arquitectura abstracta; *ii)* o transporte de mensagens de agente; *iii)* a gestão de agentes; e *iv)* a comunicação de agentes.

O objectivo da especificação da **arquitectura abstracta** é assegurar a interoperabilidade e a reutilização [188]. Neste sentido, são identificados e descritos, os elementos arquitecturais-base e as suas inter-relações, que devem ser utilizadas como directivas. Com esta estratégia, as plataformas desenvolvidas, independentemente da linguagem de suporte e de arquitectura interna, partilham um modelo abstracto semelhante, o que capacita a interoperabilidade. A arquitectura estabelece uma distinção entre elementos-base propostos, assente no seu nível de abstracção. Não foram ainda concretizados os elementos de gestão e a mobilidade dos agentes, por serem considerados demasiado perto da implementação e, logo, inevitavelmente comprometidos com as tecnologias utilizadas para a sua realização. Todavia, num futuro próximo, será necessário incluir este tipo de elementos na arquitectura-base, de modo a poderem ser utilizados como indicadores de implementação. Actualmente, estão definidos, entre outros, os elementos abstractos de transporte de mensagens de agente, (FIPA – ACL), serviços de directórios e linguagens.

[34] Tradução do autor para a frase *The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.*

O objectivo da especificação FIPA visa a definição do **transporte de mensagens de agente** e prende-se com a necessidade de representação e de especificação de mecanismos de envio, bem como a entrega de mensagens, utilizando distintos protocolos de transporte de dados em redes de computadores. A especificação utiliza uma solução comum a diversos protocolos de comunicação, definindo a existência de um envelope e de um corpo de mensagem. O envelope é utilizado pelo nível de transporte, permitindo a realização do encaminhamento das mensagens aos serviços de transporte de mensagens *(MTS - Message Transport Service).* Naturalmente, foram definidos campos obrigatórios, que asseguram a informação necessária, de modo a permitir a realização do encaminhamento das mensagens por parte das plataformas de Agentes. O corpo da mensagem é expresso respeitando o formato FIPA – ACL *(Agent Control Language)*, todavia, não é utilizado pelo MTS, podendo, inclusivamente, ser codificado, (e. g., por questões de segurança, pode ser cifrado; por questões de diminuição de dimensão da mensagem, pode ser comprimido).

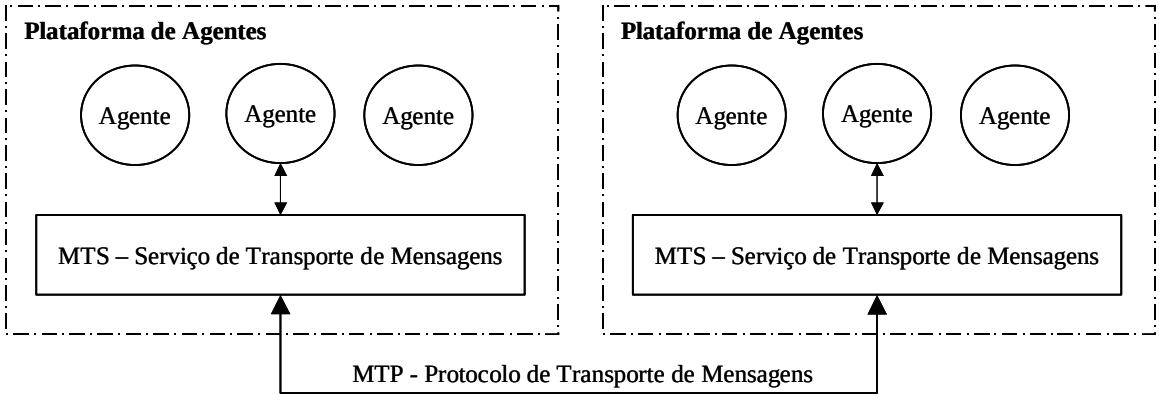


Figura 11 – Modelo de referência para o transporte de mensagens de agentes, definido pelas especificações FIPA. (Figura adaptada do original da FIPA)

O modelo de referência para o transporte de mensagens de agentes, representado na Figura 11, permite identificar os seguintes componentes:

- As funcionalidades gerais dos MTS das plataformas de agente [189];
- Indicações de utilização de protocolos de transporte de mensagens MTP - «Message Transport Protocols», tais como IIOP [190], HTTP [191] e WAP [192];
- Codificação do envelope de mensagem em conformidade com os diversos MTPs, tais como codificação XML para HTTP [193] e codificação eficiente para *bits* [194];
- Codificação da representação de mensagens FIPA – ACL, em conformidade com os diversos MTPs, tais como codificação para XML [195], codificação para cadeias de caracteres [196] e codificação eficiente para *bits* [197];

O MTS de cada plataforma deve ser desenvolvido de forma modular, com vista a poder incorporar facilmente protocolos de transporte de mensagem, envelopes de mensagem e futuras representações FIPA – ACL, que possam surgir. Compete ao MTS realizar as conversões necessárias entre os MTPs, assegurando, desta forma, a comunicação dos agentes com o exterior.

O objectivo da especificação de **gestão de agentes** é a definição de um modelo de enquadramento, onde os agentes FIPA existem e operam. Neste modelo de referência são definidas, entre outras, as formas de activação, registo, desactivação, localização e mobilidade de agentes. O modelo de referência define os serviços, as primitivas e as ontologias dos dois elementos responsáveis pela execução das tarefas de gestão de Agentes: o Sistema de Gestão de Agentes *(AMS – Agent Management System)* e o Facilitador de Directório *(DF – Directory Facilitator)* que podem ser identificados na Figura 12 (figura adaptada do original da FIPA).

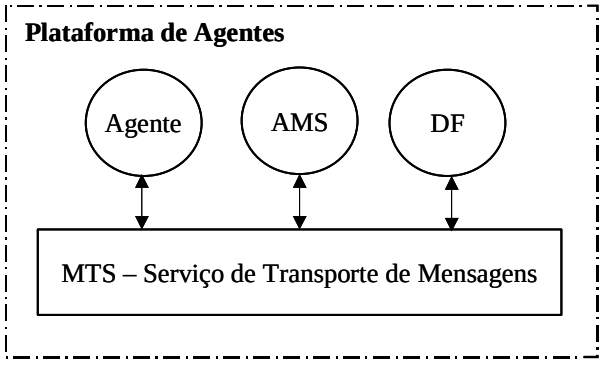


Figura 12 – Modelo de referência para a gestão de agentes

O AMS é responsável por disponibilizar o serviço de páginas brancas, que consiste no registo de nomes, localização e controlo de acesso dos agentes. O DF é responsável por facultar o serviço de páginas amarelas, que consiste no registo e localização de serviços prestados por agentes. O AMS e o DF são operacionalizados através de dois agentes obrigatórios, em todas as plataformas FIPA, sendo activados com a plataforma.

Compete a esta especificação, em conjunto com as especificações de transporte de mensagens de agente, a definição de mecanismos de sincronização e comunicação com sistemas que estejam acessíveis de forma intermitente, (e. g., computadores pessoais, agendas pessoais).

A especificação FIPA para **a comunicação entre os agentes** é baseada na criação de mensagens com semântica, i. e., as mensagens FIPA, que estão condicionadas a uma prévia definição de um conjunto de regras semanticamente ricas e conhecidas pelos intervenientes. A base da comunicação é realizada por actos de comunicação, usualmente denominados por performativas. As performativas condicionam a mensagem enviada, uma

vez que a interpretação do seu conteúdo depende directamente do seu valor. As performativas são definidas e organizadas dentro de ontologias. A comunicação entre Agentes só é efectiva caso os intervenientes sejam capazes de interpretar a ontologia utilizada.

As mensagens são enviadas através de envelopes de comunicação, apelidados de linguagem de comunicação de Agentes (*ACL – Agent Communication Language)* [168]. O ACL, para além de encapsular a mensagem, transporta o contexto da mensagem, destinatário, remetente, a ontologia e o protocolo de interacção da mensagem. A FIPA - ACL foi desenvolvida tendo como base o ARCOL, e o protocolo KQML [187].

Finalmente, o conteúdo da mensagem, pode ser representado por um conjunto de protocolos: SL – Linguagem Semântica *(Semantic Language)* [198], CCL – linguagem de escolha por restrição *(Constraint Choice Language)* [199], KIF [200] ou RDF [201].

2.4.6 Agentes Inteligentes

A utilização do termo «inteligente», à semelhança do termo «agente» não é consensual. Embora existam diversas definições sobre agentes inteligentes, no âmbito desta dissertação consideram-se somente os casos em que os sistemas desenvolvidos aplicam conceitos de aprendizagem automática e/ou exploração de dados e conhecimento, i. e., apresentam capacidade de aprendizagem baseada no acumular de experiências.

Os Agentes Inteligentes são, assim, caracterizados por serem sistemas baseados em técnicas de inteligência artificial que actuam autonomamente ou semiautonomamente em representação de um utilizador, com vista a descobrirem ou organizarem os dados. Alguns recorrem ao conhecimento que têm do perfil dos utilizadores para pesquisar informação relevante e interpretar e organizar os dados descobertos. Outros, utilizam técnicas de recuperação de dados e características do hipertexto para a identificação dos dados relevantes. Outros ainda, são construídos com a finalidade de aprenderem as preferências dos utilizadores e, depois, aplicar esse conhecimento na pesquisa de informação.

Existem duas grandes classes possíveis de aprendizagem: a aproximação baseada em exploração de conteúdo e a exploração colaborativa/comportamento, aplicadas respectivamente, a um individuo ou a grupos de utilizadores [202].

A **exploração do conhecimento colaborativo**, foca a sua atenção no comportamento do utilizador e explora a existência de grupos de utilizadores com interesses semelhantes. Esta técnica está centrada no processamento de registos diários dos acessos, incluindo não só os destinos, como também os conteúdos consultados e o tipo de interacções efectuadas. A classificação atribuída a um documento, por um elemento do grupo, pode indiciar a sua relevância para os restantes elementos do grupo. Neste caso, a semelhança dos itens e dos documentos não é relevante, passando a ser determinante a classificação atribuída pelos

utilizadores. O sistema acumula os pareceres atribuídos pelos utilizadores, conseguindo, desta forma, construir grupos, que utiliza para auxiliar cada indivíduo. Esta abordagem, que não está limitada ao texto é, igualmente, eficiente em documentos constituídos por imagens, sons, vídeo e texto. As suas limitações mais relevantes prendem-se com a dispersão dos dados, logo, não se aplica a pequenas comunidades (existe a necessidade de massa crítica, o que se agrava na sua aplicação à Internet, pela quantidade de informação disponível), dada a existência de indivíduos com gostos singulares, o que pode impossibilitar a sua inclusão em grupos. Este tipo de aprendizagem é, igualmente, apelidada de aprendizagem social. As aplicações mais comuns são no combate à fraude de personificação, com a identificação do intruso por alteração do comportamento do indivíduo, assim como na utilização de técnicas de *marketing* personalizado (pela adequação dos conteúdos) antecipando os movimentos e necessidades do utilizador.

Alguns exemplos de conhecidos agentes baseados em exploração do conhecimento colaborativo, são:

i) O Siteseer, um sistema de recomendação de páginas Web que utiliza os *bookmarks* e a sua organização em directorias para inferir as preferências do utilizador. A agregação dos utilizadores é realizada em função da semelhança de *bookmarks* e na da organização, sendo as sugestões realizadas tendo por base os URL partilhados nos grupos, bem como os partilhados nas directorias [203];

ii) O GroupLense foi desenvolvido para realizar a filtragem de notícias das Usenet e baseia-se nas classificações atribuídas às mensagens e no agrupamento dos utilizadores, em função das classificações atribuídas [204];

iii) O Firefly é um agente de interface que aprende com os utilizadores e com os outros agentes, podendo ser utilizado para recomendação musical personalizada. O agrupamento de utilizadores é realizado por comparação das classificações atribuídas aos conteúdos [205];

Em alternativa à criação de agentes inteligentes, através da aplicação de técnicas de exploração de conhecimento, **existem as técnicas de exploração de conhecimento dos conteúdos** que ocupam um papel determinante nesta dissertação. Nesta aproximação, o princípio-base assenta na comparação de itens semelhantes para sintetizar informação. Na maioria dos casos esta abordagem está limitada à análise de textos, estando, assim, reduzida a aprendizagem em texto e excluída, por maioria de razão, a análise de conteúdos multimédia, tais como as imagens, os vídeos e os sons. A aprendizagem em texto aplica as técnicas de aprendizagem automática a bases de dados de texto. A utilização da aprendizagem automática extravasa, em muito, o âmbito restrito da Web; todavia, a maioria das técnicas utilizadas noutros domínios pode ser directamente transposta para a Web, tendo em conta que a informação disponibilizada na Internet continua a ter como base

principal, o suporte em texto. A sua utilização permitiu a definição de um vasto conjunto de aplicações de extracção de informação, de descoberta de informação, de integração de fontes da Web, de indexação automática, de detecção de eventos, de classificação, de organização, de filtragem de informação (e. g., mensagens de grupo de notícias[35] ou correio electrónico), de apresentação de documentos, da reformulação de perguntas, etc.

Genericamente, estas aplicações, podem dividir-se nas seguintes categorias: *i)* disponibilização de informação em formato estruturado para utilização em perguntas complexas ou resolução de problemas, aumentando o nível de abstracção; *ii)* auxílio à pesquisa, à organização e à manutenção de informação.

Alguns exemplos de agentes baseados em aprendizagem de exploração de conteúdos são:

i) O BargainFinder foi um dos primeiros exemplos, de agentes passivos, que tinha como finalidade descobrir e coligir informação sobre a venda de CDs na Web, permitindo a comparação de preços nas diversas fontes [206];

ii) O WebWatcher, realiza a selecção de informação da Web, recebendo palavras-chave e devolvendo elos que são posteriormente avaliados, com vista a personalizar e melhorar o desempenho do sistema [207];

iii) O Mustang utiliza uma filosofia de pesquisa de informação semelhante, baseada nas palavras-chave fornecidas pelo utilizador. Todavia, mantém um *thesaurus* que relaciona conceitos semanticamente semelhantes, o que permite enriquecer a pesquisa com termos relacionados com as palavras-chave fornecidas [208];

iv) O Letizia, foi desenvolvido com o objectivo de auxiliar o utilizador a navegar na Web. O sistema é baseado na análise do comportamento do utilizador, com o objectivo de permitir antecipar os seus próximos movimentos. Assumindo que os utilizadores realizam, essencialmente, pesquisas em profundidade, o sistema antecipa a navegação e sugere as próximas páginas [209, 210];

v) O Anatagonomy constrói um boletim de notícias na Web e, por monitoração do comportamento do utilizador, actualiza o seu perfil com o objectivo de melhorar as selecções efectuadas. O formato do boletim é criado, tendo em conta a classificação atribuída às notícias que reflectem o perfil do utilizador [211].

2.5 Contribuições após o estado da arte

Terminada a apresentação do estado da arte das áreas que influenciaram directamente os trabalhos realizados, é possível fazer uma reanálise das contribuições reclamadas, nesta dissertação, no capítulo de Introdução.

[35] Grupo de notícias – Tradução de «newsgroup».

As contribuições de carácter geral estão relacionadas com a definição de um enquadramento global e de uma arquitectura abstracta que permite a recolha e catalogação de informação baseada em catálogos dinâmicos, e da sua aplicação a um estudo de caso, tendo como empresas-alvo as PMEs.

O enquadramento global proposto inverte o paradigma tradicional, em que o investimento é realizado no portal e ignora o papel do sistema do utilizador. A aproximação proposta, típica das soluções baseadas em agentes, apresentadas no final da secção dos sistemas de multiagentes, permite **a personalização do catálogo dinâmico e a recolha dos dados necessários, sem violar a privacidade do utilizador,** o que, apesar de aumentar a sua autonomia obriga a um conjunto acrescido de responsabilidades. Este assunto é aprofundado na Metodologia Geral do próximo capítulo.

A arquitectura proposta para a recolha de dados, baseada em agentes inteligentes, permite aplicar os métodos, processos e ferramentas existentes neste paradigma. Desta forma foi possível **ultrapassar a complexidade associada ao objectivo** proposto, facilitando a obtenção de resultados de forma elegante. O paradigma de agentes foi utilizado, numa aproximação *ad-hoc* como metáfora de modelação, implementação e fonte de tecnologia contribuindo, exclusivamente, como mais um exemplo de demonstração da aplicabilidade e viabilidade da utilização do paradigma. Este assunto é aprofundado na secção da Arquitectura abstracta do próximo capítulo.

A utilização das PMEs no papel de cliente Web permitiu criar um estudo de caso que explora **o modelo de negócio baseado em cadeias de fornecimento em rede**. Esta aproximação permite a criação de cadeias de fornecimento temporárias para a criação de produtos ou linhas de produtos de pequena tiragem. A proposta apresentada permite a **identificação dos produtos/serviços mais adequados e com melhor relação preço/qualidade,** sendo, assim, um factor de viabilização do modelo de negócio. As vantagens oferecidas estão relacionadas com a facilidade de utilização, aliadas a manutenção dos processos de negócio existentes e ao baixo custo, o que não acontece com as restantes soluções. Este assunto é aprofundado no capítulo Estudo de Caso.

As contribuições de carácter específico estão relacionadas, essencialmente, com a recuperação e extracção de informação.

A proposta de uma metodologia para assegurar a representatividade de um *corpus* permite garantir uma fiabilidade elevada nos resultados apresentados, evitando a introdução de erros resultantes de desvios verificados nos dados empregues. A confiança, sempre relativa, na representatividade do *corpus,* associada ao método de validação cruzada por dez secções para a generalização de erro, foi essencial, para permitir a comparação de resultados.

A apresentação de um modelo de representação de documentos, adequado à aprendizagem em texto, procurou identificar uma solução genérica que capturasse a informação sobre a localização e destaque das palavras. O enriquecimento da representação tradicionalmente utilizada, baseada num vector de existências ou ocorrência, está centrada na procura da manutenção da informação sobre a apresentação dos dados.

A utilização da Informação Mútua Condicional (IMC), após a ordenação através da IM e do QQ, para a optimização da selecção de características, permite melhorar os resultados habituais devido à eliminação das características co-relacionadas, aumentando a expressividade do vector de representação. Esta proposta, aliada a uma melhor representação dos documentos, visa a obtenção de descritores que favorecem a indução dos algoritmos de classificação.

O método C4.5 iterado, uma pequena variação ao algoritmo de indução de estimadores C4.5, procurou optimizar a compactação das árvores induzidas pela avaliação individual das características consideradas. A compactação das árvores é essencial para a obtenção de Sistemas de Apoio à Decisão (SAD) eficientes que permitam a classificação de um maior número de observações, no menor tempo e com o menor número de recursos computacionais possíveis.

A criação de SAD baseados em vários estimadores, especificamente através do método Fajé, permite tirar partido da diversidade de desempenho, dependendo da localização da observação. Desta forma, beneficia-se das contribuições de cada estimador, evitando perder a informação capturada (por diferentes métodos e conjuntos de treino), pela utilização simples do melhor.

Finalmente, a solução apresentada para reconhecimento de conteúdos (representação e regras) e classificação de conteúdos (regras «se-então» e referência inversa de palavras-chave) permite a extracção de informação de forma flexível, assegurando um processo contínuo de melhoramento e adequação a novos casos. A proposta sugerida, garante que o sistema é capaz de fazer a adaptação a alterações e a novos casos com um esforço mínimo por parte do utilizador, tendo em conta que a informação é armazenada em regras que podem ser facilmente alteradas.

3 Enquadramento global

O enquadramento global proposto visa permitir o acesso à informação disponibilizada na Web através de um catálogo dinâmico, com o objectivo de ultrapassar os obstáculos associados às pesquisas na Internet, que decorrem da existência de demasiada informação, não estruturada, e na maioria dos casos inútil, que conduz a resultados improdutivos e muitas vezes frustrantes.

Esta proposta não tem como objectivo fazer a substituição das soluções de pesquisa actuais, mas sim possibilitar um processo alternativo de monitoração dos dados disponibilizados num conjunto de sítios previamente seleccionado. O objectivo é permitir a consulta eficiente e eficaz da informação evitando a navegação na Web. Em alternativa à realização de pesquisas e/ou à consulta directa dos sítios envolvidos, o acesso efectiva-se através de um catálogo personalizado, previamente enriquecido com os dados recolhidos dos sítios que se pretende monitorar, permitindo a consulta num ambiente conhecido.

Esta abordagem, suportada numa inversão do paradigma tradicional aplicado aos portais de pesquisa da Web, transfere o motor de pesquisa para o utilizador, o que lhe permite, um total controlo dos seus dados pessoais e de todo o processo de pesquisa.

Naturalmente, uma consequência imediata do paradigma proposto é a redução do âmbito das pesquisas, tendo em conta que não é expectável a existência de capacidade computacional nos utilizadores finais que permitam efectuar pesquisas globais. Todavia, espera-se que o aumento de especificidade assegure um melhor desempenho local, e um maior grau de privacidade e sofisticação.

O enquadramento global assenta nos seguintes pilares: *i)* a definição da metodologia geral; *ii)* a proposta de uma arquitectura de referência; *iii)* a definição de uma metodologia específica de suporte à derivação de sistemas particulares; e *iv)* a apresentação do protótipo.

Sumariamente, a metodologia geral define as técnicas e soluções adoptadas, a arquitectura de referência substancia e define de forma abstracta o sistema, e a metodologia específica de suporte à derivação de sistemas particulares permite a operacionalização para casos reais. O protótipo é constituído por dois subsistemas fortemente interdependentes:

- Sistema de Catalogação (instância da arquitectura de referência): opera com as funcionalidades de pesquisa e os agentes contextualizados às necessidades particulares do utilizador;
- Sistema de Derivação (instância da metodologia de derivação): implementado pelo Agente Tutor, que permite auxiliar a captura do conhecimento, que é posteriormente, transferido para o sistema de produção.

O subsistema de derivação permite alterar o comportamento do subsistema de catalogação, pela troca de mensagens que descrevem o conhecimento sintetizado.

3.1 Metodologia Geral

A metodologia geral define um conjunto de aproximações que permitem atingir os objectivos propostos. As opções mais relevantes são: *i)* a adopção de um formato único normalizado utilizado por todos os módulos; *ii)* a formalização do conhecimento pela utilização de ontologias; *iii)* o recurso a técnicas de Recuperação e Extracção de Informação baseadas em aprendizagem; *iv)* a utilização de sistemas de multiagentes; *e v)* a definição de uma interface baseada num catálogo dinâmico [212].

Com este conjunto de opções é possível ultrapassar alguns dos obstáculos mais relevantes à pesquisa de informação da Internet:

i) a existência de diversos formatos de dados na Web;
ii) a necessidade de formalização do conhecimento com vista a adopção de uma terminologia comum;
iii) a capacidade de processamento dos dados, desde a filtragem de documentos tendo por base a sua relevância, passando pela identificação dos conceitos e sua posterior classificação;
iv) a natureza do problema, de carácter distribuído, complexo, descentralizado e com reduzida estruturação;
v) a existência de uma multiplicidade de interfaces de utilizador com especificidades próprias.

3.1.1 Formato neutro dos dados

O facto da Web ser composta por uma intrincada rede de documentos descritos em diversos formatos de dados, e. g., HTML, SHTML, XML, RDF, Word, Excel, obrigou à definição de um formato único que permitiu o isolamento de todos os componentes, das especificidades

de cada protocolo. Desta forma, a diversidade de protocolos, em parte responsável pelo sucesso da Web, deixa de criar problemas ao processamento automático (que passa a fazer todo o processamento sobre um formato de dados único). Cada novo elo deixa assim de, potencialmente, conduzir a um novo formato de dados o que obrigaria à utilização de um novo tipo de interpretador.

O recurso a uma representação interna única, formato neutro, força naturalmente à utilização de conversores de dados que se ocupam de realizar a tradução da informação a partir dos formatos originais. Consequentemente, a conversão do formato original, através do conversor correspondente, e o armazenamento em formato neutro passa a ser uma tarefa obrigatória para todos os dados recolhidos na Web. A Figura 13 permite ilustrar a localização lógica dos conversores de dados, à entrada do sistema, isolando, a montante, todos os módulos do sistema que utilizam exclusivamente o formato neutro.

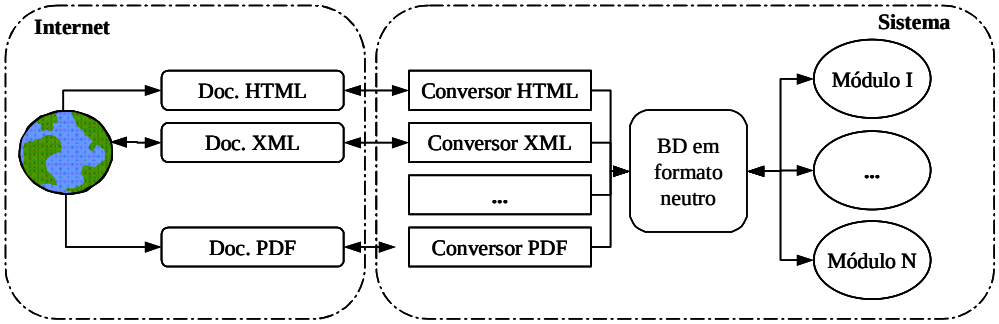


Figura 13 – Apresentação do modelo lógico de conversão de dados dos diversos formatos existentes na Web para a representação do formato interno

A introdução de um novo conversor de dados capacita imediatamente o reconhecimento do novo formato em todo o sistema. Esta abordagem permite uma aproximação incremental, permitindo testar o sistema com um conjunto de formatos pré-seleccionados, validando a consistência da arquitectura sem comprometer o futuro, uma vez que a utilização de dados descritos nos novos formatos está, somente, dependente da construção dos respectivos conversores [213].

3.1.2 Formalização do conhecimento

A formalização e utilização do conhecimento, em sentido lato, representa um factor central no desenvolvimento de sistemas de informação. Todavia, uma aproximação consensual à sua especificação está longe de ser atingida existindo diversas aproximações, visões e normas. Filósofos e psicólogos reconhecem que a memória humana é associativa, e que os humanos tendem a compreender e classificar a realidade pela criação de modelos e classificações abstractas, com o objectivo último de categorizar as observações e realizar a sua interpretação [214]. A aproximação adoptada para a representação do conhecimento baseia-se na utilização de uma ontologia. Os conceitos reconhecidos estão descritos na

ontologia adoptada, que serve inclusive de base à interface com o utilizador e com o sistema de processamento de dados. A ontologia adoptada é assim o elo agregador de todo o sistema, tendo em conta que os diversos «interlocutores» recorrem à sua utilização como base de interacção:

i) os utilizadores finais, pela navegação na ontologia para realização de pesquisas;
ii) os administradores de sistema pela inclusão de novas regras «se-então», palavras-chave, etc., nos conceitos descritos na ontologia;
iii) o sistema autónomo de processamento de dados pela sua utilização da ontologia como base na classificação dos conceitos identificados.

A adopção de uma ontologia como base na formalização do conhecimento permite genericamente:

- a fácil criação de uma interface intuitiva baseada na navegação num grafo acíclico e exploração de conceitos abstractos, de conceitos-detalhe e de conceitos semelhantes;
- um processo natural de descrição unívoca dos conceitos e suas inter-relações.

Para evitar a dependência da arquitectura de uma ontologia específica, recomenda-se uma linguagem de representação, em especial o OWL, devido à grande flexibilidade apresentada, e ao forte apoio do W3C, e pelo programa DARPA ter abdicado do DAML+OIL.

Esta solução permite o reconhecimento automático das ontologias descritas em OWL, garantindo à partida, a utilização do sistema em novos contextos consoante o surgimento de novas propostas.

3.1.3 Recuperação e Extracção de Informação

O processamento dos dados é a tarefa mais ambiciosa em todo o sistema, tendo em conta que o objectivo é sintetizar a informação de documentos que não contêm a respectiva informação semântica. O recurso a técnicas de Recuperação e de Extracção de Informação foi fundamental para a concretização desta tarefa.

Tendo em conta que a maioria dos documentos disponibilizados na Web não possui a informação semântica, este objectivo é essencial, permitindo explorar fontes de dados não frequentes. Existem diversas razões pelas quais os documentos continuam a não ser disponibilizados com informação semântica. A esmagadora maioria dos documentos continua a ser criada em formatos focados na apresentação dos dados, e. g., HTML o que inviabiliza imediatamente a inclusão de adicional. No caso da utilização de formatos de dados que permitem a inclusão de conteúdo semântico, e. g., XML, continuam a não existir terminologias universais que permitam a sua extracção de forma unívoca. Mesmo na eventualidade da maturação da Web semântica existirá sempre um alargado conjunto de

informação disponível sem a respectiva informação semântica. Isto acontecerá por não ter sido capturada de forma involuntária por não ser o enfoque da mensagem do autor ou, voluntariamente, no caso do autor querer apresentar a informação mas não pretender o seu processamento automático. A disponibilização de informação sobre produtos é um bom exemplo. Mesmo que venha a existir uma ontologia que defina de forma unívoca todos os produtos e serviços, dificilmente será adoptada uma vez que permitirá o processamento automático dos dados, o que poderia contribuir para diminuir a margem de lucro dos comerciantes.

A abordagem adoptada para a recuperação de informação segue a sequência de tarefas apresentada na Figura 14. A primeira tarefa visa a filtragem dos documentos que não apresentam informação relevante para o utilizador. A segunda tarefa, efectuada somente sobre os documentos seleccionados, visa o reconhecimento dos conceitos apresentados nos documentos e a última tarefa visa a sua classificação.

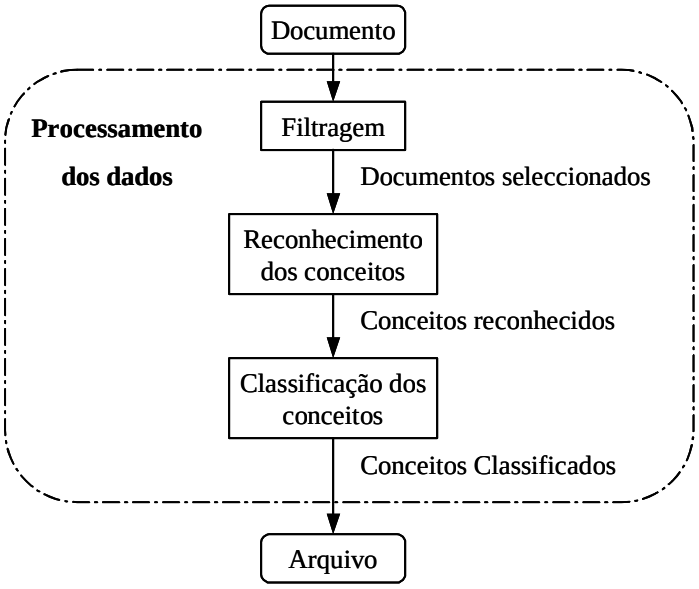


Figura 14 – Apresentação das tarefas associadas ao processamento dos dados

Tendo em conta que se pretende monitorar um conjunto de sítios Internet e sendo expectável que os sítios armazenem muita informação irrelevante para o utilizador, a primeira tarefa do processamento de dados consiste na identificação dos documentos relevantes na óptica do utilizador, com o objectivo de realizar uma filtragem de informação. Desta forma, reduz-se o esforço efectuado nas tarefas seguintes, uma vez que, somente, os documentos considerados importantes serão posteriormente analisados para extracção de informação.

Este processo de filtragem tem que ser extremamente eficiente devido à elevada quantidade de informação a processar. A sua eficácia é igualmente determinante, tendo em conta que as falhas contribuem negativamente para o desempenho do sistema de duas formas: *i)* na

eliminação de páginas relevantes, equivalendo à perda de informação e *ii)* na selecção de páginas irrelevantes, contribuindo para dificultar o processamento efectuado nas fases seguintes.

A determinação da noção de relevância é todavia uma das tarefas mais ambiciosa, tendo em conta que se pretende automatizar a aquisição da informação. A automatização permite evitar submeter o utilizador a um processo moroso e complexo de descrição dos seus interesses, o que diminuiria a atraccão global do sistema.

A utilização de técnicas de recuperação de informação em texto, com o recurso à aprendizagem supervisionada foi a aproximação seguida para a obtenção desta capacidade. A extracção do conhecimento sobre a relevância dos documentos é, assim, efectuada a partir de um *corpus* composto por exemplos positivos e negativos.

A actividade seguinte visa o reconhecimento de conceitos permitindo o seu armazenamento no catálogo. O objectivo é identificar os conceitos apresentados, extraindo os seus dados, a partir de documentos que não disponibilizem informação semântica. Esta tarefa, realizada somente sobre os documentos previamente seleccionados no processo de filtragem, visa a identificação de padrões que permitam assumir a presença dos conceitos a catalogar. No exemplo da identificação de produtos, visa identificar a presença de um produto e a sua informação respectiva.

A estratégia adoptada para a obtenção desta capacidade, foi a extracção de informação em texto com o recurso a um motor de inferência de regras «se-então» que permitam descrever o formato de apresentação dos dados. Desta forma, cabe ao utilizador descrever a forma mais comum de apresentação dos dados que pretende armazenar no catálogo, permitindo ao sistema o seu reconhecimento posterior. O resultado desta tarefa são conjuntos de informação, que se espera correlacionados com o conceito que se pesquisa.

A classificação dos conceitos é o obstáculo final para a realização do processamento dos dados. Após o reconhecimento dos conceitos e a recolha dos seus dados é necessário fazer a sua classificação, o que permitirá o seu correcto armazenamento.

A abordagem a este problema baseia-se na utilização combinada de duas aproximações, com a utilização de palavras-chave, que permitem identificar conceitos, e no processamento dos URI das páginas. No primeiro caso, a classificação dos conceitos é baseada na informação extraída, fazendo-se por referência inversa, i. e., a informação recolhida é utilizada para identificar o conceito mais provável, permitindo assim a sua classificação. Naturalmente, que esta aproximação obriga à descrição dos conceitos com base nas palavras-chave mais comuns. No segundo caso, o processamento dos URI, visa explorar o facto de grande parte dos sítios Internet de disponibilização de informação serem cada vez mais de construção dinâmica, i. e., os dados estão armazenados em bases de dados, e as

páginas são construídas em consequência das solicitações dos utilizadores. Nestes casos, os URI contêm usualmente informação crucial, sobre a categoria dos produtos apresentados. Uma vez mais, esta solução obriga ao processamento manual da informação, mas, neste caso, de forma mais genérica, tendo em conta que se está a trabalhar ao nível das categorias de produto, e não nos produtos específicos, e. g., a descrição é realizada para máquinas fotográficas digitais e não para a máquina X modelo Z.

Apresenta-se de seguida um exemplo, assumindo uma vez mais o interesse do utilizador na construção de um catálogo de produtos. A Figura 15 apresenta casos ilustrativos de potenciais páginas Internet. A imagem à esquerda, apresenta informação genérica de apresentação da empresa, pelo que não é relevante para a construção do catálogo de produtos, devendo ser assim eliminada, durante a tarefa de filtragem. À semelhança desse documento, todos os documentos que não apresentem informação sobre produtos devem ser eliminados. As restantes páginas da Figura 15, já apresentam informações sobre os produtos disponíveis, todavia só a página à direita apresenta os custos associados, pelo que deveria ser a única seleccionada. Neste exemplo, eliminar páginas que não são relevantes para o utilizador, significa eliminar páginas que não apresentem produtos especificamente para venda.

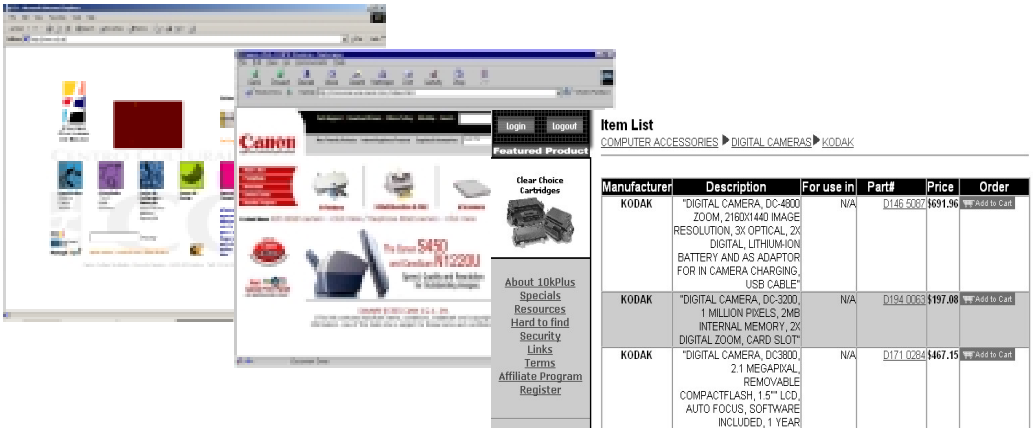


Figura 15 – Páginas da internet que permitem ilustrar as diferenças entre as duas primeiras tarefas de captura de conhecimento

A tarefa de reconhecimento de conceitos, neste exemplo, procura identificar os produtos apresentados nas páginas previamente seleccionadas. No exemplo apresentado na Figura 15 procura-se identificar a existência dos produtos apresentados em tabelas, com uma referência, descrição e preço. Como se pode constatar por comparação das duas figuras a apresentação dos produtos varia entre lojas virtuais, e mesmo no caso em que o formato de apresentação é semelhante (em forma de tabela), a sua aparência pode ser muito distinta. Neste sentido, a construção de regras genéricas é determinante para assegurar a abrangência do sistema.

Finalmente, a última tarefa, que é a classificação dos conceitos, neste caso dos produtos. Após a recolha da informação associada a um conceito com as regras pré-definidas é necessário fazer a sua classificação para posterior arquivo. Neste caso, é necessário conseguir identificar que os produtos a serem vendidos são máquinas fotográficas.

3.1.4 Sistema de Multiagentes

A natureza do problema em causa obriga à construção de um sistema complexo que tem que apresentar as seguintes propriedades:

- **modular,** tendo em conta que é naturalmente divisível em diversos módulos, a título de exemplo: pesquisa de dados na Internet, de recuperação de dados, de exploração de dados, de armazenamento de dados, entre outros. Esta facilidade de identificar, diversas tarefas, que devem ser realizadas por módulos distintos preconiza a adopção de uma solução distribuída;
- **descentralizado,** devido à distribuição geográfica dos dados que estimula a procura de uma solução de execução de tarefas perto das fontes de dados e pelas entidades mais competentes. Desta forma é essencial que cada entidade seja capaz de levar a cabo a sua tarefa, de forma, independente das restantes, limitando-se a recolher informação inicial, em tese enviada pela entidade a «montante» na cadeia de decisões, e a comunicar os resultados obtidos às entidades seguintes.
- **adaptável** a novos contextos, devido à constante adaptação a novos comportamentos definitivos em cada componente. As tarefas de alteração, remoção ou adição de novas características, são constantes e devem ficar confinadas ao tipo de módulo em causa, permitindo um controlo elevado;
- **a reduzida estruturação do sistema** é consequência da indefinição inicial das soluções propostas, o que levanta sérios problemas a uma fase de análise detalhada. A reduzida estruturação dos problemas está relacionada com a dificuldade de definir de forma precisa, na fase de análise do projecto todos os módulos da arquitectura, assim como todas as suas interacções. Os problemas que apresentam esta característica levantam sérias dificuldades às metodologias tradicionais, tendo em conta que se espera, na fase de análise, a sua completa e total descrição por forma a permitir uma implementação estruturada e sem surpresas.

Apesar de não ser uma solução mágica, as propriedades do paradigma de multiagentes adequam-se às dificuldades enunciadas tendo em consideração [135]:

- a elevada modularidade, oferecida pela natureza dos agentes, tendo em conta que cada agente é em si um módulo aplicacional, que interage com o meio e outros

agentes de modo formal. Não existe partilha de variáveis com as entidades que interagem com o agente (agentes, meio ambiente). Os agentes podem ser vistos como objectos pró-activos que possuem a modularidade que permitiu a ampla divulgação, e adopção, das linguagem orientadas por objecto.

- a elevada descentralização que reside na pró-actividade dos agentes, característica essencial na distinção face à tecnologia orientada por objectos. Tendo em consideração que cada agente é capaz de desencadear acções, e reacções a acontecimentos, estamos na presença de um sistema verdadeiramente descentralizado, tanto mais quanto estivermos em presença de implementações que tiram partido de recursos físicos distintos. Esta característica faz com que seja extremamente adaptado a sistemas que necessitem de um processo de tomada de decisão local sem a necessidade de um conhecimento do estado global. Esta capacidade é de elevada relevância uma vez que se procura uma solução que possa operar num sistema físico geograficamente distribuído, e no qual cada entidade seja capaz de levar a cabo a sua tarefa de forma independente das restantes;
- a elevada adaptabilidade devido ao facto de cada tipo de agente ser programado à custa de um código-fonte, que o caracteriza e distingue dos demais. Esta característica garante que a alteração do comportamento de um agente fica confinada à alteração desse código e não produz efeitos colaterais no comportamento de outros agentes. Esta característica, permite inclusive a remoção ou adição de agentes de forma controlada;
- a flexibilidade oferecida pela capacidade de definição do comportamento de um agente sem ter uma visão global do sistema e dos seus componentes, assegura a adequação a problemas pouco estruturados. Por outras palavras, é possível definir o comportamento de um agente sem se possuir a descrição total do meio ambiente e do comportamento dos outros agentes.

O paradigma de multiagentes foi utilizado para suportar a modelação dos sistemas, e sua implementação. Genericamente, cada tarefa principal é executada por um tipo específico de agente. O número específico de cada tipo de agentes activado é determinado em função das necessidades particulares de cada tarefa.

3.1.5 Interface única

Tendo em consideração o objectivo de evitar a utilização de múltiplas interfaces optou-se pela utilização de uma interface única (o catálogo dinâmico) que armazena os dados de todos os sítios monitorizados. O catálogo é actualizado pelo sistema de multiagentes e pelas pesquisas prévias, sendo a organização da informação baseada numa ontologia.

Desta forma, evita-se a interacção com um conjunto alargado de sítios Internet dispersos, e organizados segundo filosofias e interfaces próprias. A existência de uma interface única garante a facilidade de utilização e diminui a necessidade de adaptação constante às alterações impostas pela dinâmica própria dos diversos sítios Internet. O catálogo funciona, assim, à semelhança dos portais que disponibilizam directórios de acesso à informação na Web. A utilização do catálogo permite personalização e assegura um conforto adicional ao utilizador permitindo acelerar os processos repetitivos.

3.2 Arquitectura de referência

A arquitectura de referência sintetiza as opções metodológicas adoptadas, estabelecendo um modelo abstracto que é passível de ser particularizado para casos concretos. Genericamente, os documentos são armazenados em formato neutro, e todos os módulos acedem aos dados através de uma interface única. A formalização do conhecimento é realizada através de uma ontologia, estando descrita em OWL, assegurando uma terminologia comum. O processamento de dados é suportado em técnicas de aprendizagem de recuperação e extracção de informação em texto. A unidade principal de abstracção em toda a arquitectura é o agente, e a interface com o utilizador efectua-se através de um catálogo dinâmico.

A arquitectura está organizada em três módulos-base, que apesar de serem interdependentes, têm funções distintas e bem definidas. Neste contexto foram definidos os seguintes módulos:

- **O Catálogo dinâmico**, «Dynamic Catalogue (DC)»**,** que tem como função armazenar e disponibilizar, através de uma interface intuitiva, o acesso a toda a informação coligida;
- **Sistema de pesquisa directa**, implementado através do subsistema de agentes de Portal, «Portal Interface Agent System (PIA)», que permite a integração de agentes dedicados nos portais-alvo, responsáveis por disponibilizar uma porta de acesso privilegiada;
- **O sistema de pesquisa autónoma**, «Multi-Agent System (MAS)»**,** responsável por pesquisar a informação nos sítios Internet seleccionados, identificar as páginas que contêm temas e conteúdos relevantes para o utilizador, e fazer o carregamento semiautomático dos mesmos no catálogo.

A Figura 16 apresenta os módulos lógicos da arquitectura, as suas inter-relações, e a interacção com a envolvente, sendo possível identificar os dois mecanismos-base de carregamento de informação. **O sistema de pesquisa directa,** com o recurso aos agentes remotos de interface (PIA), instalados nos sítios internet, com os quais se pretende uma interface privilegiada, e o agente agregador, responsável pela interface entre o catálogo e os

agentes remotos. **O sistema de pesquisa autónoma** identificado pelo circuito de agentes Navegador, Explorador, Catalogador.

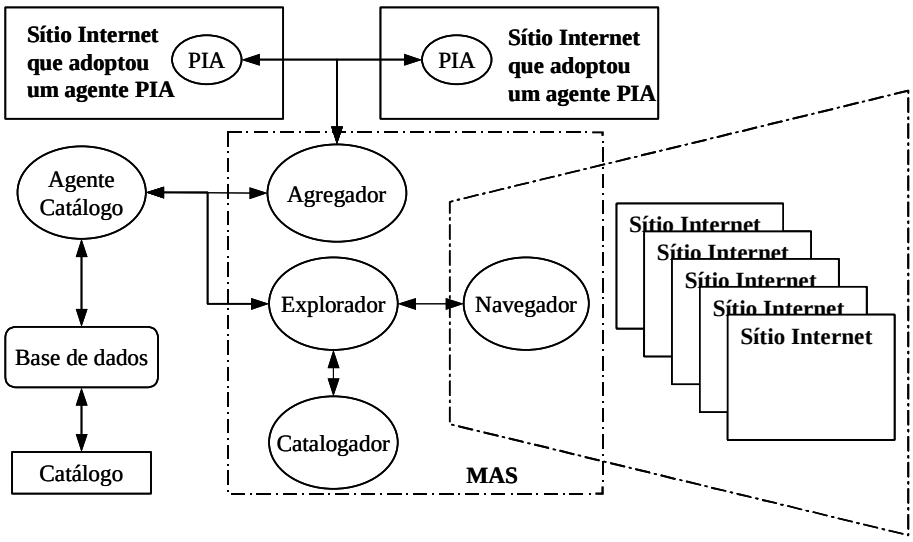


Figura 16 – Apresentação abstracta da arquitectura de referência

Independentemente da via de pesquisa de informação, os resultados são armazenados na base de dados do catálogo, assegurando uma utilização e interface únicas.

3.2.1 O catálogo dinâmico

O catálogo dinâmico, «Dynamic Catalogue (DC)», tem como função armazenar a informação coligida e disponibilizar uma interface para o utilizador tendo em conta as suas preferências. O catálogo recebe os dados provenientes dos sistemas de multiagentes (pesquisa directa e pesquisa autónoma), armazenando toda a informação em base de dados. A informação armazenada está catalogada segundo os conceitos da ontologia, previamente adoptada pelo utilizador. A organização numa ontologia permite tirar partido de todas as vantagens oferecidas por esta disciplina de representação de conhecimento, essencialmente na organização interna e no processo de criação das pesquisas.

A nível da organização interna, os conceitos estão previamente definidos e não existem ambiguidades, permitindo o acesso eficiente e eficaz aos dados. Contudo, as vantagens mais visíveis residem ao nível da interacção com o utilizador, por permitirem:

- eliminar pesquisas ambíguas, com inevitáveis resultados improdutivos, uma vez que a construção das perguntas é realizada por navegação no grafo de conceitos;
- efectuar pesquisas com diversos graus de abstracção, tendo em conta que o utilizador pode seleccionar os conceitos com o detalhe que considera adequado;
- construção de pesquisas auxiliada por detalhe progressivo dos conceitos, e numa fase mais avançada por sugestões baseadas na experiência adquirida;

- reduzir a obtenção de respostas inúteis uma vez que os resultados apresentados foram previamente catalogados;
- possibilitar a disponibilização de uma interface gráfica de representação do grafo de conceitos, defendida por diversos estudos como uma forma intuitiva e ergonómica, uma vez que tudo indica ser este o processo natural de organização mental utilizado para a representação do conhecimento [214].

Para além de todos os dados coligidos na Web, o catálogo armazena um histórico de acções, a ontologia pré-seleccionada, e informação sobre as preferências do utilizador.

3.2.2 O sistema de pesquisa directa

O sistema de pesquisa directa tem por objectivo a criação de uma interface privilegiada com sítios Internet que se disponibilizaram à execução de perguntas directas. A intermediação entre o sistema de multiagentes e o sítio, é realizada pela instalação no sítio destino, de um agente remoto de interface, «Portal Interface Agent (PIA)». No fundo, o PIA é responsável por disponibilizar uma porta de acesso privilegiada, permitindo um mecanismo de pesquisa directa sobre as bases de dados dos portais. Cabe ao PIA a tarefa de fazer a conversão entre a terminologia adoptada pelo sistema e a taxinomia de produtos existente em cada sítio. Desta forma, no processo de instalação do PIA é necessário estabelecer as relações entre as duas terminologias com vista à boa execução das pesquisas.

Este módulo é opcional, uma vez que a informação pode, potencialmente, ser coligida através do sistema de multiagentes. Todavia a sua inclusão disponibiliza ao fornecedor de informação, sítio Internet, um canal fiável, flexível e integrado com o catálogo, através de perguntas directas à sua base de dados.

A função do agente de interface é, assim, receber as perguntas enviadas pelo sistema, fazer a tradução das mesmas para o sistema local, e devolver a resposta. A comunicação com o sistema de multiagentes é mediada pelo agente agregador, que para além de conhecer todos os agentes instalados, tem permissão para comunicar com o agente interface do catálogo.

3.2.3 O sistema autónomo de pesquisa

O sistema de pesquisa autónoma é realizado pelo MAS. O MAS efectua a pesquisa autónoma, filtra a informação, identifica e cataloga os conceitos relevantes e actualiza semiautomaticamente o catálogo. A Figura 17 permite identificar o grupo de agentes mais relevantes no MAS, agrupados por funcionalidades, pelo que a cada elipse da ilustração corresponde a um, ou a um grupo de agentes. O objectivo de apresentar os agentes desta forma, visa a criação de uma figura mais simplificada que permite uma introdução progressiva do subsistema, assegurando uma visão gradual. Os restantes agentes serão

introduzidos, no momento em que forem estritamente necessários. Genericamente, cabe ao Navegador pesquisar os sítios na Internet recomendados pelo utilizador e seleccionar as páginas que contém temas relevantes; em seguida essas páginas são transferidas para o Explorador que tem que executar a tarefa de reconhecimento de conceitos, e posterior classificação em colaboração com o agente catalogador, sendo finalmente transferidos os dados para o catálogo.

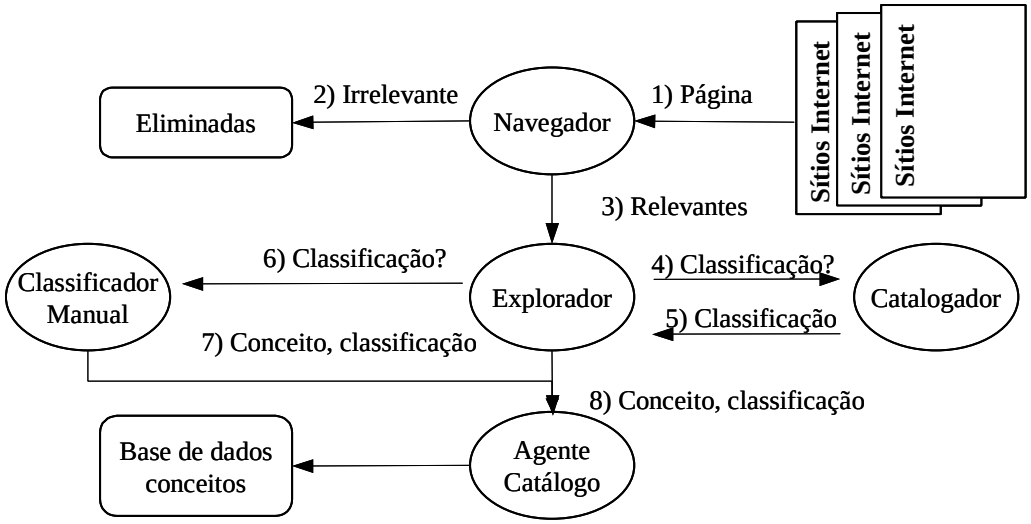


Figura 17 – Os meta-agentes existentes na arquitectura

O agente Navegador, *o* «Crawler», é responsável por realizar a actividade de pesquisa de todas as páginas pertencentes ao conjunto de sítios Internet que lhe são fornecidos pelo utilizador, seleccionando somente as páginas que possuem temas relevantes para o utilizador. Cabe assim ao Navegador, para além da actividade de navegação nos sítios, a execução da primeira tarefa do processamento dos dados, a filtragem.

O Agente Explorador, o «Miner», tem a seu cargo a tarefa de reconhecer os conceitos presentes nas páginas previamente seleccionadas pelo Navegador; fundamentalmente cabe-lhe a execução da segunda tarefa do processamento dos dados.

A classificação dos conceitos identificados pelo Explorador é realizada em colaboração com o agente Catalogador, que é responsável pela manutenção da ontologia e por catalogar os conceitos encontrados tendo em conta a informação recolhida (informação do conceito e do URI da página).

No caso da classificação atribuída pelo Catalogador ser inconclusiva, o que acontece sempre que é impossível identificar univocamente um conceito na ontologia, o conceito em causa é enviado para a interface de classificação manual que permite ao utilizador não só escolher uma classificação, como fornecer dados que permitam evitar o mesmo caso no futuro.

3.3 Metodologia específica de suporte à derivação de sistemas particulares

O carácter abstracto da arquitectura geral obriga à sua derivação para casos particulares, possibilitando pesquisas de informação específicas e adequadas às necessidades do utilizador. Apesar do sistema ser iniciado com um conjunto base de regras, (conhecimento inicial), responsável por um comportamento-base genérico, só o conhecimento crescente sobre o utilizador permite a criação de novas regras assegurando a sua personalização.

A derivação do sistema é realizada pela particularização do sistema de pesquisa autónoma.

A metodologia de derivação assenta na execução das seguintes fases: *i)* definição da ontologia de representação de domínio; *ii)* indução do SAD para o Navegador; *iii)* definição do SAD para o Explorador; e *iv)* Personalização da ontologia para o Catalogador.

O agente Tutor auxilia a realização de todas as fases permitindo libertar os restantes agentes destas tarefas mantendo a sua estrutura leve, desprovida de capacidades de inferência de conhecimento ou de descrição de novas regras. O Tutor fica assim encarregue de sintetizar os novos comportamentos, num processo paralelo ao funcionamento normal, e de enviar os resultados aos agentes destino, nomeadamente aos Navegadores, aos Exploradores e ao Catalogador. O condicionamento do comportamento dos agentes é efectuado pelo envio de mensagens que descrevem a informação a transferir.

***i)* Definição da ontologia de representação do domínio**

A primeira fase de derivação do sistema particular visa a representação do conhecimento reconhecido em cada sistema particular e é efectuado pela definição da ontologia de domínio. A arquitectura de referência prevê que a ontologia represente dois domínios de conhecimento: *i)* conhecimento sobre os assuntos considerados relevantes; *ii)* conhecimento sobre os conceitos catalogáveis. A sua especificação é encarada como uma acção de importação de duas ontologias de domínio.

O domínio sobre assuntos relevantes é utilizado essencialmente no âmbito do agente Navegador, como base de classificação do *corpus*, e posteriormente de classificação dos documentos analisados.

O domínio de conceitos catalogáveis, é empregue em todo o sistema, estando presente desde a interface de consulta, até aos agentes que têm como função identificar e reconhecer os conceitos catalogáveis.

Somente após a definição consistente da ontologia é possível iniciar as acções seguintes de derivação, que podem ser realizadas em paralelo.

***ii)* Indução do SAD para o Navegador**

Esta fase consiste em assegurar ao Navegador a capacidade de classificar os documentos que analisa como pertencendo a assuntos considerados relevantes ou irrelevantes. Neste sentido, é necessário executar a sequência de acções tradicionalmente associadas às tarefas de recuperação de informação em texto conducentes à indução do SAD. O recurso às técnicas utilizadas, assegura ao sistema a capacidade de adequação às necessidades e requisitos dos utilizadores, garantindo uma adaptabilidade incrementalmente às suas necessidades particulares.

As acções a executar, com o auxílio do agente Tutor, são:

i) A criação da base de dados, o *corpus*, de exemplos classificados segundo a ontologia de domínio de assuntos relevantes, utilizado pelos algoritmos de aprendizagem;

ii) A selecção das características mais relevantes através do conjunto de algoritmos disponibilizados, permitindo assim a eliminação de características redundantes, desnecessárias e indutoras por ruído;

iii) a indução de classificadores, através do conjunto de algoritmos de indução de classificadores que podem ser utilizados na criação de sistemas de tomada de decisão;

iv) a indução de SAD, pela determinação de qual o melhor método de combinação dos classificadores criados;

v) a exportação dos resultados para os agentes activos.

Após a execução desta tarefa o sistema fica capacitado para reconhecer os assuntos considerados relevantes, passando o Navegador a poder efectuar a filtragem dos documentos que encontra na Web.

***iii)* Definição do SAD para o Explorador**

Esta fase consiste em assegurar ao Explorador a capacidade de extrair a informação sobre os conceitos descritos na ontologia de domínio dos conceitos catalogáveis.

O reconhecimento dos conceitos é realizado através da utilização de um motor de inferência com utilização de regras *forward* «se-então», que descrevem as formas típicas de apresentação dos dados, e pela análise do elo de referência do documento.

A execução desta fase passa, assim, pela execução das seguintes tarefas:

i) identificar as formas típicas de apresentação de informação e fazer a sua descrição com a utilização de regras (regras de apresentação), e. g., uma forma típica de apresentação de informação é a utilização de tabelas;

ii) descrever a relação entre os descritivos do conceito e os elementos de dados extraídos com as regras de apresentação, i. e., atribuir significado semântico aos dados extraídos para cada componente;

iii) descrever as palavras-chave apresentadas no elo de evocação da página para permitir a extracção de informação semântica associada aos elos.

O Tutor disponibiliza uma interface de utilizador que permite a definição de regras de inferência «se-então» que permitem a identificação dos conceitos e da descrição dos elos.

***iv)* Personalização da ontologia para o Catalogador**

Esta fase consiste em assegurar ao agente Catalogador a capacidade de reconhecer conceitos segundo a ontologia de domínio. O reconhecimento dos conceitos é realizado por comparação das palavras-chave compostas, com as palavras atribuídas a cada conceito da ontologia. Desta forma, cada conceito da ontologia tem que ser descrito com um conjunto de palavras que permitem o seu reconhecimento posterior.

Neste sentido, esta fase passa pela definição para cada conceito, que se pretende reconhecer, de um conjunto de palavras-chave que permitam o seu reconhecimento futuro pelo sistema. O conjunto de palavras utilizadas para a descrição de cada conceito deve ser o mais rico possível, contudo os conjuntos utilizados devem ser disjuntos para evitar ambiguidades.

3.4 Implementação do protótipo

A prototipagem do sistema teve início, após a definição da metodologia geral e da definição da arquitectura de referência, e acompanhou em paralelo a definição da metodologia de derivação de sistemas particulares. Foi, assim, possível testar conceitos e métodos encurtando o ciclo de desenvolvimento.

Apesar de ser possível a implementação do sistema proposto numa perspectiva de aplicação independente, optou-se por uma solução em ASP[36], essencialmente devido:

- a garantir a monitoração da utilização do protótipo;
- à facilidade de demonstração progressiva de capacidades de forma controlada, sem necessidade de fazer actualizações do sistema;
- à simplicidade de actualização da solução de forma incremental pela alteração de componentes aplicacionais e inclusão de novas regras de conhecimento que influenciam o comportamento dos agentes, sem ser necessário implementar mecanismos complexos de disseminação de informação;

[36] ASP –*Application Service Provider*

- a evitar o difícil processo de instalação, em especial a criação de conhecimento de base por parte dos utilizadores da solução.

Desta forma assegura-se a disponibilização das funcionalidades de forma controlada através da internet.

A primeira versão do protótipo foi desenvolvida utilizando a linguagem de programação JAVA, no ambiente de desenvolvimento FORTE 2.0, recorrendo ao pacote Swing para melhorar as capacidades gráficas da linguagem e simplificar o trabalho do programador, devido à vasta colecção de primitivas gráficas. Para o suporte ao desenvolvimento do sistema de multiagentes recorreu-se à plataforma JatLite 0.4 Beta, tirando partido dos benefícios naturais de uma infra-estrutura já desenvolvida, com o encapsulamento de um conjunto de problemas meramente tecnológicos, e de construção de Agentes. Com o objectivo de simplificar o desenvolvimento do agente navegador adoptou-se o pacote WebSPHINX, uma biblioteca de classes JAVA que disponibiliza primitivas de navegação na Web.

A base de dados relacional utilizada foi o Microsoft SQLServer, com acesso via ODBC, e para a manutenção da ontologia o ambiente Protegé.

O sistema operativo-base de suporte ao desenvolvimento foi o Windows, todavia a implementação assenta numa filosofia Web em que os servidores de base de dados, de Web e segurança estão instalados em máquinas Linux. Esta aproximação deveu-se, principalmente, a questões de disponibilidade material.

Foi neste ambiente que o primeiro protótipo foi desenvolvido permitindo a sua primeira validação, contudo por questões de compatibilidade com a comunidade de multiagentes decidiu-se a migração da sua implementação.

O segundo protótipo foi implementado no ambiente de desenvolvimento Sun One Studio 5, com o recurso à plataforma de agentes JADE. O JADE assegura a conformidade com as normas FIPA, e permite a utilização de um conjunto de recursos que facilitam a implementação de soluções, contribuindo para a identificação de soluções elegantes que ultrapassaram limitações anteriores.

No decurso da adaptação ao JADE optou-se por migrar para uma base de dados de código livre, o MySQL. Esta decisão permitiu a criação de uma arquitectura exclusivamente em código-fonte aberto, reduzindo custos de licenciamento.

O ambiente de desenvolvimento final foi, assim, Sun One Studio 5, com o recurso à linguagem de programação JAVA, com os pacotes JADE 3.0, WebSPHINX 0.5 e Swing 1.3, com suporte em base de dados MySQL 4.0 e Protegé 2000 para a manutenção da

ontologia. Foi com este ambiente que se materializou o actual protótipo que permitiu a validação da arquitectura, por obtenção de resultados experimentais.

3.4.1 As ferramentas de desenvolvimento utilizadas

Esta secção realiza uma breve apresentação das ferramentas aplicacionais utilizadas no desenvolvimento do protótipo actual. Tendo em conta que existe uma vasta informação disponível na Internet esta apresentação visa, simplesmente, permitir uma caracterização genérica sobre as capacidades oferecidas.

A linguagem de programação seleccionada foi o JAVA, em particular pelo seu paradigma de programação orientada por objectos em ambientes heterogéneos distribuídos.

A linguagem JAVA, criada pela SUN Microsystems em Junho de 1995, gerou uma onda de aceitação, em virtude de ser orientada por objectos, e ter sido desenhada para aplicações Internet, permitindo a sua execução de forma independente da plataforma física.

Uma das vantagens do Java passa pelo facto de ser uma linguagem pensada de raiz sem qualquer preocupação de retrocompatibilidade com linguagens imperativas (como foram o caso do C++ ou Delphi), o que torna o Java uma linguagem simples e bem estruturada, fácil de compreender, na qual o tempo perdido em depuração e desenvolvimento de aplicações, se torna significativamente menor em relação a outras linguagens. O facto de basear a sua sintaxe na linguagem C, (uma das mais usadas até hoje), torna o tempo de aprendizagem dos processos básicos de programação muito menor relativamente a outras linguagens. Outro ponto forte do Java reside no facto do código desenvolvido ser independente da plataforma onde é executado. Deste modo todos os pormenores inerentes à plataforma ficam encapsulados, tornando as aplicações facilmente portáveis entre plataformas. A perda de desempenho associada à interpretação do código, tem sido compensada pelos avanços nos últimos anos em relação às máquinas virtuais de Java. O facto do JAVA ter uma política de código-fonte livre, assegura uma enorme quantidade e qualidade de aplicações e pacotes disponíveis para reutilização permitindo, consequentemente, a aceleração do processo de desenvolvimento. O próprio Java inclui uma extensa biblioteca de objectos que encapsulam interfaces e estruturas de dados, e. g., Streams de dados, Vectores, Tabelas de Hash, Sockets.

Numa perspectiva mais técnica, a restrição à utilização de herança simples, torna o desenvolvimento mais rígido, contudo, reduz a complexidade em termos de compreensão da aplicação. O uso de interfaces facilita a implementação de novas funcionalidades sem haver necessidade de modificar código antigo. A passagem de parâmetros exclusivamente por referência simplifica, uma vez mais, o desenvolvimento de aplicações. A existência do objecto, «Object», do qual todos os objectos são herdados directa ou indirectamente, torna toda a linguagem altamente estruturada, tornando fácil a documentação das aplicações.

A existência de métodos privados, públicos e protegidos ajudam grandemente no encapsulamento do comportamento de objectos. Um excelente tratamento de erros, através de excepções, ajuda o programador na recuperação de erros em tempo de execução e na na fase de depuração da aplicação.

Resumindo, o Java é uma linguagem de programação orientada por objectos, com grande aceitação na comunidade informática, que concilia a simplicidade de desenvolvimento com uma rigidez e disciplina necessárias ao bom desenvolvimento de aplicações eficientes e intuitivas em termos de compreensão do código implementado.

O ambiente de desenvolvimento utilizado, foi o Sun One Studio, que resulta da evolução do Forte for Java, e o seu surgimento representou um marco de sucesso na melhoria da utilização de recursos e no aumento do desempenho comparativo, tendo permitido a sua afirmação.

O Sun One Studio alia o alto desempenho, a uma série de ferramentas que simplificam o desenvolvimento em Java. À semelhança da maioria dos ambientes de desenvolvimento actuais suporta «sintax highlighting», auto-complete e identação inteligente, disponibilizando um ambiente versátil de depuração de código, com «breakpoints» e «watches». Possui um servidor de base de dados embebido, «pointbase», e um gerador de «javadocs» e ficheiros de tipo «jar». Suporta CVS para controle de versões e todas as classes podem ser compiladas através do ambiente de trabalho, devido a estreita relação que o Sun One Studio mantém com a Máquina Virtual existente localmente. Disponibiliza uma construção intuítiva de Interfaces Gráficas para Utilizadores (GUI) onde os componentes são inseridos através do método «Drag & Drop», muito utilizado por quem usa sistemas de operação gráficos. O Sun One Studio permite, igualmente, a criação de «Java Beans» e «Servlets», proporcionando todas as ferramentas necessárias para a criação e execução dos mesmos.

O desenvolvimento das interfaces gráficas foi conseguido, essencialmente, com o recurso ao pacote «Swing», (o sistema gráfico do Java) que disponibiliza um conjunto de funcionalidades que permitem a construção de interfaces intuitivas.

Os sistemas gráficos do Java são baseados em camadas permitindo posicionar os componentes gráficos, (como botões e campos de texto), em formulários. O «Swing» disponibiliza novas camadas de abstracção muito mais flexíveis, de modo a tornar o aspecto de aplicações com interfaces visuais mais apelativas e funcionais.

É possível controlar a largura, comprimento, transparência, cor, tipo de letra de qualquer componente «Swing», tornando mais fácil a construção de interfaces gráficas.

A criação e o refrescamento de componentes usa menos recursos do computador, tornando as aplicações mais «leves» e eficientes.

O «Swing» contém, também, novos componentes com funcionalidades que facilitam a visualização e manipulação de informação em termos visuais, e. g., barras de progresso; «Sliders»; árvores de ícones; paleta de Cores; campos para palavras-chave.

A ferramenta de representação de conhecimento utilizada foi o Protégé-2000 desenvolvido por Stanford Medical Informatics na Stanford University School of Medicine com o suporte de diversas agências dos EUA [215]. O Protégé é um editor de conhecimento representado através de ontologias, tendo sido desenvolvido em código-fonte livre, com a linguagem JAVA, o que permite uma fácil integração com aplicações que necessitem de uma base de conhecimento por extensão. A ferramenta é muito versátil e intuitiva permitindo, essencialmente, a construção de ontologias de domínio e a manutenção da informação.

O **sistema de gestão de base de dados** utilizado foi o MySQL devido, essencialmente, a ser uma ferramenta de código livre e de utilização sem licenciamento para projectos sem fins lucrativos, aliada ao elevado desempenho em termos de inserções e pesquisas de dados, em especial na versão 4, pela introdução de uma *cache* de tabelas. O MySQL suporta ANSI SQL e as principais funções de aritmética e grupo, o que permite uma versátil construção de perguntas. O MySQL é, principalmente, direccionado para aplicações com reduzido número de utilizadores em simultâneo, cenário no qual este servidor está a ser usado actualmente neste projecto. As versões actuais suportam, entre outras: «outer joins»; tabelas temporárias; campos de tamanho variável e fixo. Este SGBD, permite acessos ODBC, e tem sido utilizado como base de dados com elevada quantidade de informação (existem casos de bases de dados com 50 milhões de registos ou 60 mil tabelas). O elevado desempenho está directamente relacionado com a inexistência de mecanismos de controlo de integridade referencial entre tabelas (deixados à responsabilidade utilizador).

A utilização do MySQL como suporte ao armazenamento de dados provou a sua adequabilidade, tendo ultrapassado com sucesso a elevada quantidade de informação armazenada, e a necessidade de acessos expeditos, um dos requisitos mais determinantes na implementação.

A **plataforma de desenvolvimento de agentes** utilizada foi o JADE[37] (Java Agent Development Framework), o que se deveu principalmente: *i)* a simplificar a implementação de sistemas de multiagentes pela disponibilização de uma camada aplicacional intermédia; *ii)* respeitar as especificações impostas pela FIPA; e *iii)* pelo conjunto de ferramentas de suporte ao desenvolvimento e correcção de erros [216]. A plataforma oferece ao programador:

[37] JADE – é uma marca registada pelo CSELT, estando a utilização da plataforma autorizada segundo as regras de licenciamento para código livre. Mais detalhes podem ser encontrados em http://jade.cselt.it.

- conformidade com as especificações FIPA conseguida pelo seguimento das normas impostas pela FIPA, incluindo a existência de agentes-base, facilitadores distribuídos, implementação do protocolo IIOP de comunicação com outras plataformas, serviço de nomes e diversas bibliotecas FIPA;
- um sistema de mensagens com um conjunto alargado de mecanismos de envio, transporte e recepção de mensagens ACL, optimizados ao meio e à localização dos agentes intervenientes;
- uma plataforma distribuída em múltiplos servidores, pela utilização dos mecanismos de processos e pelas máquinas virtuais de JAVA, o que permite execução de tarefas paralelas, em processos, no mesmo servidor ou em servidores distintos;
- capacidades pré-definidas para os Agentes, fornecendo um modelo conceptual genérico de Agente que permite ao programador o enfoque da sua atenção no desenvolvimento do comportamento específico, tendo por base o conjunto de funcionalidades pré-definidas, e sendo possível a implementação de soluções de filosofia reactivas ou BDI;
- uma interface de gestão e correcção de erros que disponibiliza uma interface intuitiva para a gestão da plataforma, monitorização dos agentes e correcção de erros o que acelera o processo de desenvolvimento dos sistemas de multiagentes.

Estas características não comprometeram o desempenho da plataforma, essencialmente devido à adopção de uma filosofia de «utilizador-pagador» presente no desenvolvimento, com o objectivo de manter os custos de «overhead», usualmente associados a este tipo de arquitectura, a níveis muito baixos. Esta preocupação está presente nas soluções adoptadas, e a sua existência não penaliza o desempenho global da plataforma de Agentes. Alguns exemplos de opções tomadas, descritos em [217], são:

- os diversos meios de comunicação possíveis, implementados com protocolos de comunicação alternativos, para transportar as mensagens ACL o que assegura que somente o meio de comunicação seleccionado apresenta custos de «overhead»;
- o escalonamento cooperativo para os diversos comportamentos dos agentes, em detrimento da utilização de processos independentes. Esta solução reduz drasticamente os custos de mudança de contexto e de escalonamento (os métodos de sincronização são por vezes cem vezes mais lentos, devido aos custos associados à política de gestão de «locks» [218]). Esta opção é tanto mais relevante, quanto o facto da utilização de diversos processos, para cada comportamento, no caso específico dos agentes, não contribuir para uma solução mais interessante, tendo em conta que o espaço de memória dos agentes é usualmente comum a todos os comportamentos;

- a reutilização de recursos em diversos contextos, ao contrário da sua destruição e criação, reduzindo o esforço de gestão de memória dinâmica. (É necessário manter presente que uma chamada à função «new» desencadeia em média, cerca de 150 evocações de métodos [218]).

A utilização do JADE revelou-se interessante tendo em conta que a arquitectura proposta, o sistema de comunicações e o modelo de execução dos agentes é extremamente flexível, fácil de utilizar e assegura, de imediato, a conformidade com as especificações FIPA. A flexibilidade do JADE advém, em grande medida, pela conformidade com as especificações FIPA que determinam que somente os comportamentos externos aos Agentes podem ser definidos, pelo que não existem restrições ao desenvolvimento.

No anexo A.4 é apresentada a arquitectura da plataforma de Agentes, os mecanismos de comunicação, os modelos de execução dos agentes e algumas ferramentas de gestão.

4 Operacionalização da arquitectura de referência

A operacionalização da arquitectura de referência origina a construção de um protótipo composto por dois sistemas-base: **o Sistema de Catalogação** e o **Sistema de Apoio à Derivação de Sistemas Particulares**.

O **Sistema de Catalogação** é o sistema que permite o armazenamento e a consulta dos dados recolhidos através das pesquisas previamente efectuadas. Da operacionalização da arquitectura de referência, resultam as seguintes funcionalidades-base: o sistema de pesquisa directa, o sistema de pesquisa autónoma e as interfaces (onde está incluído o catálogo dinâmico).

O **Sistema de Apoio à Derivação de Sistemas Particulares**, consubstanciado no agente Tutor, permite a personalização do Sistema de Catalogação, pela definição de regras e SAD específicos, dedicados a cada caso concreto. Consequentemente, o comportamento global do sistema de catalogação é condicionado pelo Tutor através da sua capacidade de síntese de conhecimento.

O protótipo foi integralmente desenvolvido em tecnologia Web.

4.1 Sistema de Catalogação

O **Sistema de Catalogação** é composto pelas seguintes funcionalidades-base: **o sistema de pesquisa directa, o sistema de pesquisa autónoma e as interfaces de sistema** (onde está incluído o catálogo dinâmico).

Como já foi previamente referido, o **sistema de pesquisa directa** tem por objectivo a criação de uma interface privilegiada com sítios Internet que permite a execução de perguntas directas, **o sistema de pesquisa autónoma** permite a recolha de dados de forma semiautomática e **as interfaces de sistema** apresentam os resultados das pesquisas e permitem consultas locais.

4.1.1 O sistema de pesquisa directa

O objectivo desta pesquisa, como já foi descrito, é permitir um acesso directo aos sítios que decidiram ter uma relação privilegiada com o sistema. Pretende-se assim obter uma interface que permita fazer perguntas directas (e. g., solicitar informações sobre um conceito particular).

Este sistema é composto por dois tipos de agentes: os **agregadores** e os de **interface de portal (PIA)**. O **agente agregador** encarrega-se de fazer a gestão autónoma de um conjunto de agentes PIA que estão instalados remotamente nos sítios Internet que optaram pela sua instalação. O **PIA** concentra o cerne do trabalho, funcionando o agente agregador com um agente facilitador.

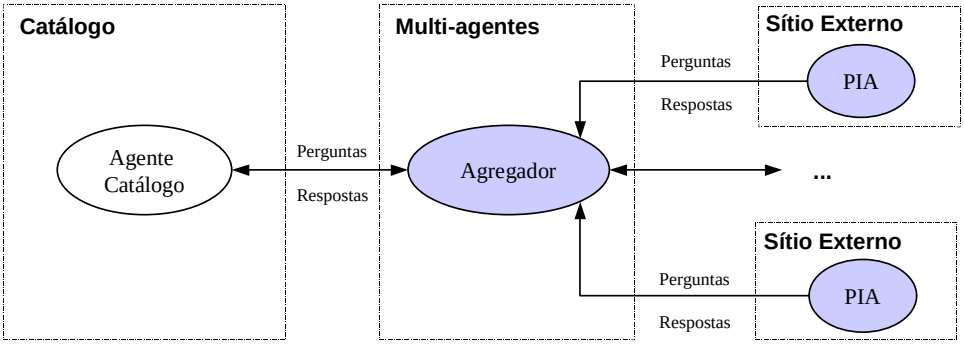


Figura 18 – Representação lógica dos agentes de uma pesquisa directa

A instalação de aplicações informáticas, em especial de agentes, com acesso directo aos dados reais, em plataformas terceiras, não é uma matéria pacífica, pela resistência frequente imposta pelas entidades envolvidas. Nestes casos é comum que as dificuldades técnicas sejam as mais simples de ultrapassar, por comparação com as barreiras sociais e culturais impostas. Todavia, neste trabalho ignoram-se deliberadamente estes problemas potenciais, assumindo nesta dissertação, uma predisposição à adopção do PIA. Assim, os problemas potenciais de natureza cultural não são despicientes nem devem ser esquecidos, com o risco de não ser possível levar à operacionalização dos agentes.

O agente agregador é um agente facilitador, encarregue da interface entre o Agente Catálogo e os PIA activos.

Uma nova pesquisa directa no catálogo, desencadeia uma mensagem do agente Catálogo para o agente agregador, notificando-o da necessidade de interpelar de novo os PIA activos. Por cada mensagem, o agregador executa as seguintes acções:

i) processa a mensagem enviada pelo agente Catálogo, que encapsula a pesquisa directa;

ii) verifica quais os agentes PIAs activos e disponíveis para receberem novas perguntas;

iii) encaminha a mensagem para os agentes PIA activos;

iv) recebe as respostas de cada agente, fazendo o seu envio para o agente Catálogo.

Com o objectivo de permitir a activação, em simultâneo, de diversas interfaces de catálogo, o agente permite o processamento em paralelo de diversas pesquisas. Por cada mensagem recebida, e após verificar a disponibilidade de, pelo menos, um agente activo, o agente agregador cria um novo comportamento de processamento de mensagem.

O agente passa, a partir desse momento, a aguardar mensagens de resposta de cada um dos PIA contactados. As mensagens recebidas são imediatamente redireccionadas para o respectivo agente Catálogo permitindo assim ao utilizador a recepção de resultados parciais.

Este comportamento termina quando todos os PIAs contactados enviarem uma mensagem de fim de resposta, ou por excesso de tempo.

O agente PIA é um agente de interface, entre o MAS e os sítios que optaram por estabelecer uma interface privilegiada. A funcionalidade-base deste agente é permitir o acesso à informação armazenada em cada sítio aderente, pelo que têm que conseguir lidar com a existência de fontes de informação muito heterogéneas. O agente é instalado, no cliente numa máquina que permita o acesso às bases de dados que contêm as informações que serão disponibilizadas pelos sítios. Após a correcta instalação e personalização do PIA, o agente comunica a sua activação passando a ser automaticamente identificado pelo MAS, em especial pelo agente agregador. Não existem limitações ao número de PIA a instalar, tendo em conta que existe um sistema de registo por nome único no MAS.

A boa execução da pesquisa de informação, obriga a que o PIA seja capaz de realizar as seguintes actividades:

i) recepção de mensagens de agente e interpretação do seu conteúdo;

ii) reconhecimento do conceito pesquisado por consulta à ontologia do MAS;

iii) identificação do conceito equivalente na terminologia do fornecedor, o que obriga à existência de uma tabela de correlação entre as duas terminologias;

iv) o conhecimento dos meta-dados da BD do fornecedor, i. e., tabelas e campos que armazenam os dados solicitados;

v) construção e execução da pesquisa na BD e subsequente tratamento e envio dos resultados, em formato de mensagem de agente.

A natureza heterogénea dos sistemas a integrar, conduziu à criação de um Agente com uma arquitectura interna pesada, mas que oferece a capacidade de adaptação às necessidades específicas dos ambientes-destino. **Este assunto tem sido vastamente estudado e insere-se num problema típico de integração aplicacional. As soluções adoptadas estão relacionadas com os trabalhos de Mestrado do autor [213].**

O PIA foi, assim, desenhado numa arquitectura multicamada de encapsulamento das diversas funcionalidades, apresentado na Figura 19, visando essencialmente a sua fácil adaptação a sistemas heterogéneos.

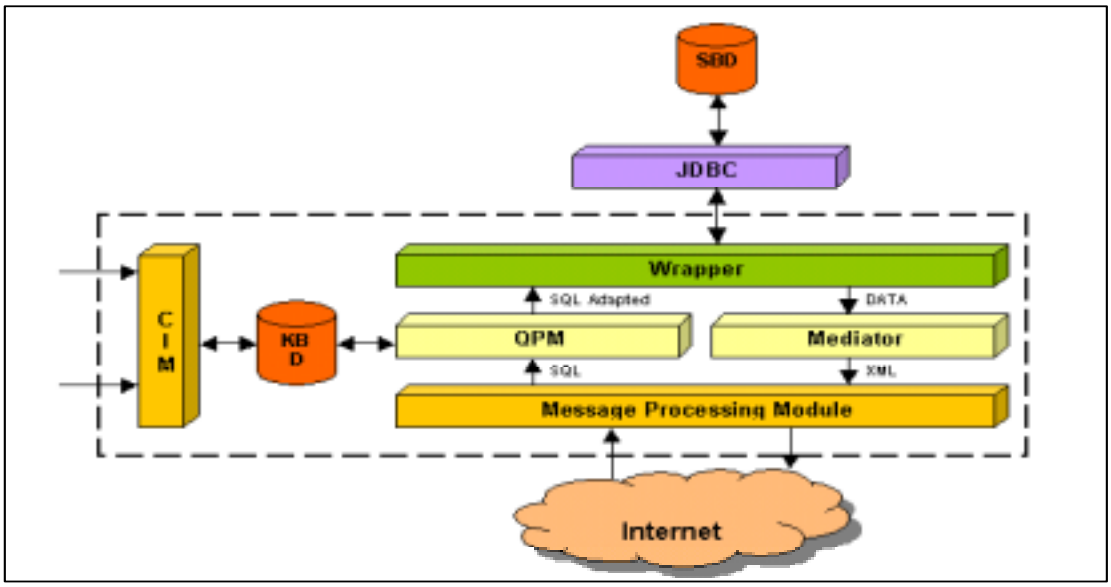


Figura 19 – Arquitectura multicamada definida para o Agente de interface com os sítios Internet. Imagem original do projecto DEEPSIA criada pela empresas Indra

O *MPM – Message Processing Module* é a janela de comunicação com o MAS, e tem como função-base a recepção e envio de mensagens ACL através da Internet. Cabe igualmente a este módulo a interpretação das mensagens ACL recebidas e conversão para uma linguagem meta-SQL, que é passada ao módulo seguinte, e em sentido inverso, a recepção da informação coligida, em formato XML, e a sua conversão em mensagens ACL. As funcionalidades mínimas exigidas a este módulo são a interrogação da existência de conceitos, quais os conceitos existentes e a descrição sobre o portal.

O *QPM – Query Processing Module* é responsável pelo controlo do Agente, cabendo-lhe a coordenação dos restantes módulos e tomadas de decisão sobre casos em que não existe uma resposta possível. Após a recepção das instruções meta-SQL enviadas pelo MPM, cabe ao QPM determinar se o conceito solicitado existe no Portal. Para este efeito recorre à KDB, onde estão armazenadas as descrições do Sítio.

A pesquisa é abortada sempre que o QPM determinar como impossível a execução da pesquisa solicitada, o que pode acontecer por:

- não identificar um conceito equivalente na BD do sítio. Neste caso, é devolvida uma mensagem que assinala a inexistência do conceito;
- imposição de restrições impossíveis de verificar, por exemplo sobre atributos inexistentes. Neste caso, é devolvida uma mensagem que assinala a impossibilidade de execução da pesquisa.

No caso, de ser possível executar a pesquisa, transforma a instrução abstracta numa instrução SQL a executar sobre a BD local, (respeitando o servidor) e transfere a mesma para o *Wrapper*, que se encarrega de executar a pesquisa.

O *Wrapper* direcciona a comunicação para a BD do portal. Este módulo é responsável por encapsular a BD, permitindo a execução das instruções SQL e a devolução da informação consultada. Cabe a este módulo impedir a execução de instruções de escrita sobre a BD do portal, garantindo um nível de segurança contra possíveis ataques.

O Mediator realiza a tarefa de análise dos resultados da pesquisa e de devolução dos mesmos para o MPM. Os resultados são enviados, organizados por conceitos, o que conduz ao envio de uma nova mensagem para o MAS por cada conceito identificado.

O *CIM – Configuration Interface Module* assegura os mecanismos que permitem personalizar o PIA às particularidades de cada sítio Internet receptor. A configuração é realizada através da utilização de uma interface gráfica que permite a funcionalidade de configuração de protocolos e aplicações. As funcionalidades oferecidas permitem, entre outras, a definição da sua localização no PIA, dos interlocutores no MAS e no Sítio, do tipo de BD e quais os *drivers* que permitem o correcto acesso (e. g., MySQL, Oracle, SQLServer); definição de utilizadores e palavras-chave; descrição da estrutura das bases de dados e das relações entre a ontologia do MAS e a taxinomia local.

O *KDB – Knowledge Data Base* armazena a informação referente à descrição dos meta-dados, das terminologias e das suas relações, e permite a consulta em tempo real pelo PIA durante o processamento das perguntas. Este módulo possui uma interface que assegura a pesquisa dos dados de forma eficiente por parte do QPM.

A configuração do sistema de pesquisa directa assenta, naturalmente, na configuração do agregador e dos PIAs. A configuração do agregador limita-se à definição dos seus parâmetros de contextualização que lhe permitem comunicar com o agente interface de catálogo e com o agente servidor de nomes. Por oposição, a tarefa de configuração dos agentes PIA às particularidades de cada sítio Internet receptor, é complexa e obriga a uma elevada intervenção humana. A operacionalização em cada novo ambiente obriga à definição dos protocolos, à localização das aplicações e à descrição e definição de inter-relações dos meta-dados dos dois sistemas envolvidos (o MAS e o sistema receptor). Desta forma, para a integração de cada PIA é necessário realizar as seguintes tarefas:

- definição da localização do agente e quais os seus interlocutores no MAS e no Sítio receptor;
- definição do tipo de base de dados e quais as interfaces que permitem o seu correcto acesso (e. g., MySQL, Oracle, SQLServer);

- definição da localização e identificação da base de dados do sítio;
- definição de um utilizador, permissões e palavras-chave para acesso à BD;
- descrição da estrutura de dados utilizada na BD do Sítio através da utilização de uma ferramenta de definição dos meta-dados, editor DTDs de XML, que permite identificar quais as tabelas existentes, o seu nome e os seus campos (tipo e nome). A automação deste processo passa pela leitura dos meta-dados de forma automática, por exemplo através de Microsoft Document Object Model (DOM);
- definição das interrelações entre a ontologia utilizada no MAS e a estrutura de dados descrita nos meta-dados; O utilizador tem que ser capaz de tomar decisões que ultrapassem a inexistência de relações directas, e. g., o nível de detalhe dos conceitos entre representações raramente é equivalente.

A interface da pesquisa directa é acedida através do catálogo, e está representada na Figura 20. Como se pode observar, a interface apresenta no canto superior direito o conjunto de sítios que estão disponíveis para interagirem com o catálogo. Naturalmente, só apareceram os portais que adoptaram o agente PIA e que estiverem activos. A interface permite a identificação e caracterização do portal que possui PIA instalado. Esta capacidade permite ao utilizador decidir sobre a utilidade da consulta ao portal pelo perfil apresentado pelo mesmo. A verificação do estado dos agentes PIA é realizada sempre que a janela da pesquisa directa é activada.

Figura 20 – Interface do catálogo dedicada à pesquisa directa

A selecção do conceito é realizada por navegação na ontologia do catálogo até identificação do nó mais adequado. Não é exigido ao utilizador a selecção de uma folha da árvore, o que permite a selecção de conceitos a diversos níveis de abstracção. Esta liberdade de selecção conduz a que não exista somente uma resposta para a pesquisa efectuada. Neste caso serão devolvidos ao catálogo, tantas mensagens quantos os conceitos correctamente identificados pelos PIAs. É disponibilizado ao utilizador informação sobre o estado das pesquisa efectuada, cabendo ao catálogo e ao agente Interface de catálogo a sua actualização num dos seguintes estados: curso, terminada ou cancelada.

4.1.2 O sistema autónomo de pesquisa

O sistema autónomo de pesquisa permite a monitoração semiautomática da informação existente num subconjunto de sítios Internet previamente seleccionados. A Figura 21 apresenta uma representação dos agentes que estão envolvidos na execução da pesquisa.

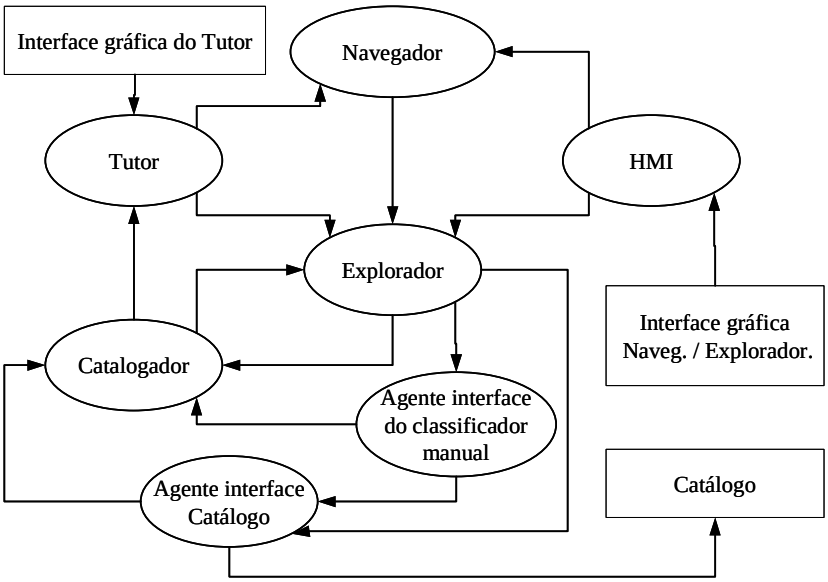


Figura 21 – Representação dos agentes do sistema autónomo de pesquisa (MAS)

Cada agente ilustrado, e. g., Explorador, Navegador, representa na verdade o conjunto de agentes do mesmo tipo, i.e. um sistema pode existir com *n* Exploradores, *k* Navegadores, etc. Os agentes são apresentados em detalhe nas secções seguintes.

4.1.2.1 O agente Navegador (Crawler)

A principal função de um navegador (Crawler) é, como o nome indica, navegar na Web, seguindo os elos existentes entre páginas. Genericamente, este tipo de agentes é utilizado pelos motores de pesquisa, ou pelos sistemas de espelho de sítios Internet[38], para

[38] Espelho de sítios Internet – Tradução para «*Mirrors*».

efectuarem a recolha sistemática de informação. A Figura 22 apresenta o Navegador, os agentes com que interage e as mensagens trocadas.

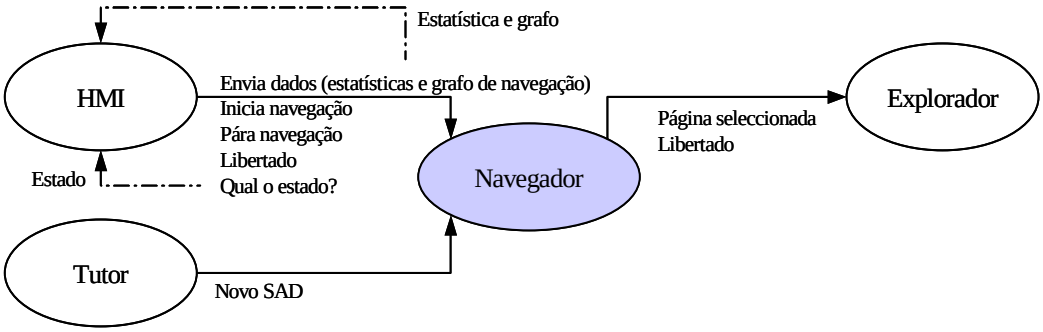


Figura 22 – Representação lógica do Navegador, dos agentes com que interage e respectivas mensagens

O **comportamento genérico do agente**, caracteriza-se por aguardar a sua configuração através de mensagens, passando em seguida ao estado de «apto para pesquisa», onde se mantém até receber a indicação do conjunto de sítios a pesquisar. A pesquisa é desencadeada pela recepção de uma mensagem, que contém a lista dos sítios-alvo e seus endereços. Por cada sítio, a página inicial é descarregada e processada, assim como todas as páginas que nela são referenciadas e assim sucessivamente. A pesquisa do sítio termina com a inexistência de novas referências ou com a recepção de uma mensagem de interrupção.

Com vista à efectivação do comportamento genérico o agente exibe o seguinte conjunto de comportamentos-base: *i)* Pesquisa; *ii)* Filtragem de informação; *iii)* Resposta à interface gráfica.

O **comportamento de pesquisa** pode alterar significativamente a eficiência do agente. Adequando a estratégia de navegação, é possível evitar pesquisas exaustivas e pouco produtivas, por condicionamento do percurso do agente, tendo em conta a informação encontrada. A título de exemplo, a pesquisa em profundidade é desadequada no caso da informação estar apresentada numa árvore de temas (i. e., cada ramo um tema, sem elos de ligação). Neste sentido, a existência de mecanismos que permitam alterar o comportamento de navegação do agente é determinante. O comportamento do Navegador pode ser condicionado pelo envio de mensagens que conduzem à alteração dos parâmetros apresentados na Tabela 4.

Parâmetro	Valor de omissão	Descrição
Navigation method	Depth	O Navegador pode realizar pesquisas em profundidade, (i. e., explorando em profundidade cada ramo da árvore) ou em extensão (i. e., pesquisando todos as páginas de cada nó).
Crawling method	Server	Modo de selecção dos elos identificadas numa página com vista à sua inclusão na lista de análise. Server: selecciona elos exclusivamente para o próprio sítio; Sub-tree: selecciona todos os elos.
Depth	100	Profundidade máxima que o Navegador percorre em cada sítio Internet, (o valor 100 corresponde na prática a infinito).
Page Timeout	60 Seg.	Tempo que o Navegador espera pelo carregamento de uma página; após esse tempo considera que a página não está acessível.
Page Size	100 Kbytes	Dimensão máxima permitida para uma página; acima deste valor não é descarregada.
Crawl timeout	-1	Máximo tempo de navegação (com o valor -1 este parâmetro é ignorado).
Use browser cache	Yes	Recurso a páginas armazenadas em memória.
Threads	1	Número de processos independentes criados para realizarem análises de páginas em simultâneo.
Obey robot exclusion	Yes	Definição do comportamento quanto ao protocolo de execução de *Robot Exclusion Protocol*[39]
Ask user for passwords	Yes	O Navegador solicita a palavra-chave sempre que for necessária.

Tabela 4 – Parâmetros de configuração do comportamento do Navegador

Os valores assumidos por omissão para os parâmetros definem o comportamento-base de pesquisa do agente no momento da sua activação. Os primeiros parâmetros condicionam o comportamento de navegação pela alteração do método de navegação, de selecção dos elos e da profundidade de pesquisa máxima admitida. O segundo grupo condiciona o processo de execução da navegação pela limitação de tempos de espera e de dimensão de páginas. Os restantes parâmetros estão relacionados com detalhes de implementação, (utilização de memória e processos) e de relacionamento do agente com o ambiente.

Para além do condicionamento do tipo de pesquisa, cabe ao Navegador, através de outro comportamento, a tarefa de **filtragem e transferência de documentos relevantes** para o agente Explorador.

A determinação da relevância dos documentos analisados é realizada através do seu sistema de apoio à decisão (SAD), previamente enviado pelo Agente Tutor. O SAD é armazenado num ficheiro XML que assegura um comportamento coerente na activação dos agentes, sem que seja necessária a interacção com o Tutor. Todavia, a possibilidade de envio por mensagem de um novo SAD permite alterar o seu comportamento assim que

[39] *Robot Exclusion Protocol* – A inclusão de uma meta-marca de exclusão de robôs de pesquisa numa página de Internet, permite informar que não são permitidas análises automáticas a partir daquela página.

desejado. A análise de relevância de um documento não é interrompida, no caso de ser recebido um novo SAD, pelo que a alteração só produzirá reflexos no documento seguinte.

O navegador não possui uma interface gráfica, todavia permite a avaliação do seu desempenho através **do comportamento de resposta a mensagens de interfaces gráficas.** É, assim, da responsabilidade de quem quer monitorar o seu comportamento, a iniciativa de consultar a actividade do agente, o que é conseguido através do envio de mensagens. Em resposta o navegador devolve toda a informação sobre as páginas visitadas, que pode ser apresentada no formato mais adequado. O tipo de informação consultado está relacionada com o número de páginas percorridas e consideradas relevantes, assim como uma listagem exaustiva de todas as páginas consultadas (endereço, classificação atribuída e localização).

A interface desenvolvida para o Navegador, apresentada na Figura 23, permite controlar os parâmetros de execução e monitorar a evolução de uma pesquisa autónoma. Existem duas zonas principais: *i)* comandos genéricos; e *ii)* relatório de pesquisa.

Na zona superior da janela encontram-se **os comandos genéricos**: alteração da lista de endereços para pesquisa, início de pesquisa, libertação de Navegador e suspensão das pesquisas.

Na zona inferior da janela são apresentados **os relatórios de pesquisa** do Navegador em dois formatos distintos: em árvore e em directoria. A Figura 24 apresenta um relatório de progresso, capturando da esquerda para a direita a evolução de uma pesquisa, desde o inicio, somente com duas páginas analisados, até ao final. Como se pode observar os documentos analisados são representadas pelo símbolo de página, sendo marcados a cinzento todos os documentos considerados relevantes.

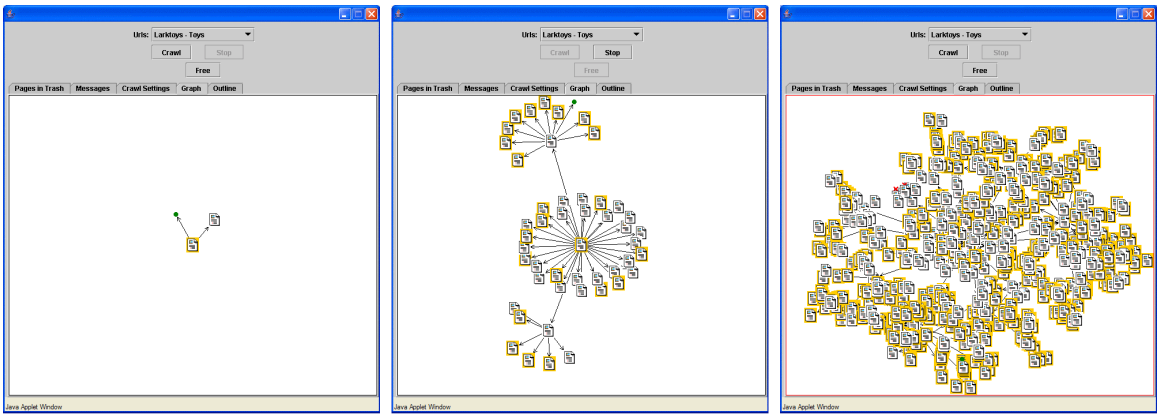


Figura 23 – Apresenta um exemplo da evolução de uma pesquisa, na versão de directoria de páginas. A cinzento estão as páginas que foram classificadas como relevantes

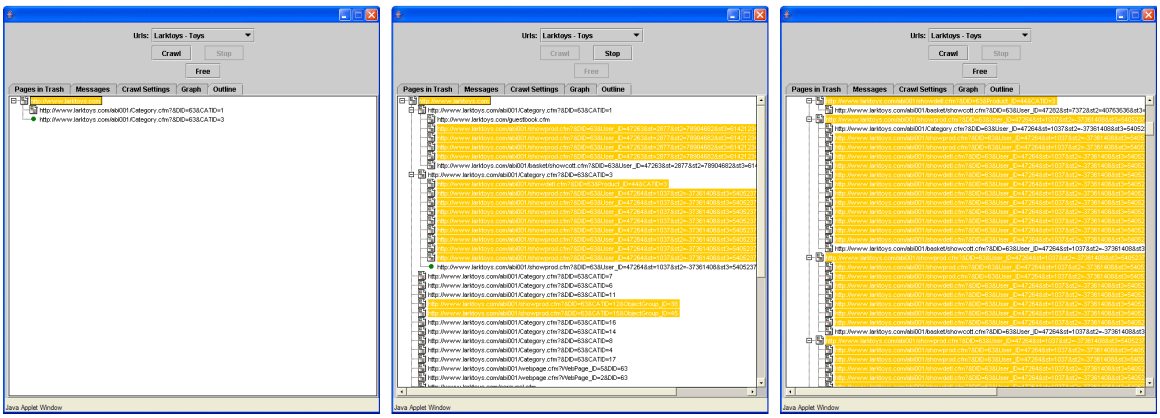


Figura 24 – Apresenta um exemplo da evolução de uma pesquisa, na versão de árvore de páginas. A cinzento estão as páginas que foram classificadas como relevantes

A Figura 24, apresenta o mesmo exemplo, agora em formato de directoria. Neste caso, os documentos analisados são apresentados numa estrutura hierárquica, estando marcados a cinzento, uma vez mais, os documentos considerados com relevantes pelo Navegador.

O pacote WebSPHINX foi utilizado no desenvolvimento do Navegador por disponibilizar um navegador genérico, circunscrevendo assim o desenvolvimento à programação dos comportamentos do agente que condicionam o processo de selecção de documentos relevantes.

4.1.2.2 O agente Explorador

O Explorador tem como função coordenar o processamento dos documentos seleccionados pelo Navegador e enviar a informação coligida para o catálogo. O processamento de cada documento, divide-se na identificação dos conceitos presentes e sua respectiva classificação. Compete ao Navegador identificar os conceitos e solicitar a sua classificação ao agente catalogador, responsável pela estimativa, em função da informação recolhida. Dependendo do sucesso da classificação, cabe ao Explorador enviar a informação recolhida directamente aos agentes interface de catálogo ou para interface de classificação manual. A Figura 25 apresenta o Explorador, os agentes com que interage e as mensagens trocadas.

O grande desafio deste agente reside assim na identificação dos conceitos relevantes para o utilizador, uma vez que as páginas na Web estão fundamentalmente preparadas para apresentarem informação a humanos, dificultando o processamento automático. Na verdade, são ainda raras as páginas que utilizam protocolos que permitam a automatização de processos (e. g., XML), pelo que a Web está repleta de informação de apresentação, que tem que ser evitada com vista a ser possível a recuperação dos dados.

O comportamento genérico do Explorador, caracteriza-se assim por aguardar uma mensagem que sinalize a existência e localização de um novo documento. Cada novo

documento é carregado localmente e o seu conteúdo é analisado em busca dos conceitos armazenados. Os conceitos reconhecidos são enviados para classificação ao agente catalogador que responde com a sua estimativa. Em função da resposta, os dados são enviados para o catálogo ou para a interface de classificação manual.

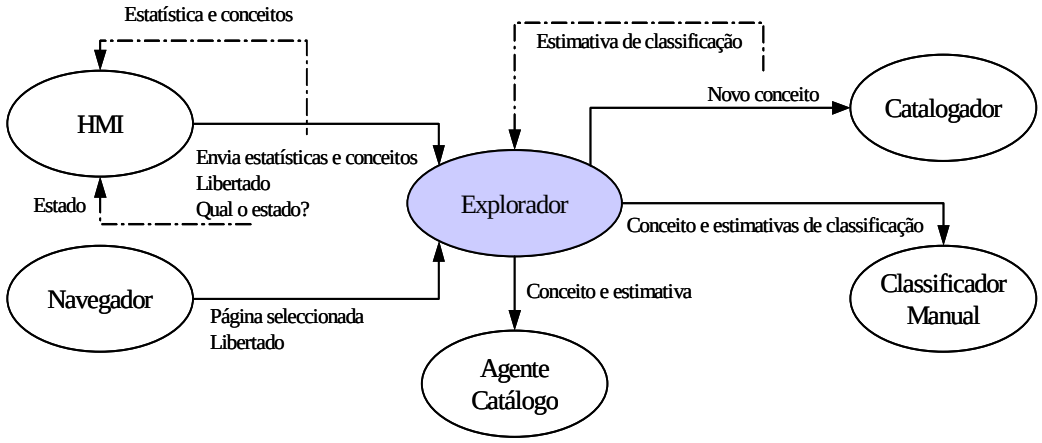


Figura 25 – Representação lógica do Explorador, dos agentes com que interage e respectivas mensagens. O tracejado representa mensagem de resposta, às mensagem a partir da qual têm origem

Com vista à efectivação do comportamento genérico, o agente exibe o seguinte conjunto de comportamentos-base: *i)* Identificação de conceitos; *ii)* Solicitação da classificação dos conceitos; *iii)* Envio da informação para os sistemas a montante.

O comportamento de identificação de conceitos é realizado com o recurso a um SAD de regras, que é armazenado numa base de dados de conhecimento (BDC) local. As regras de inferência são baseadas no mecanismo «se-então», e são definidas manualmente, i. e., não existe aprendizagem automática.

As regras são construídas por análise crítica e são essencialmente baseadas nos processos comuns de apresentação dos conceitos. Uma análise detalhada dos documentos armazenados no *corpus* de exemplos permite a identificação das regras que facultam ao Explorador reconhecer a localização dos conceitos.

Um exemplo ilustrativo de um regra para identificação, no caso em que os conceitos são produtos, é a existência de uma tabela com um cabeçalho, contendo determinadas palavras que classificam as colunas, e. g., descrição, preço, referência ou sinónimos.

As regras são descritas com o auxílio do agente Tutor, que as transfere por mensagem para os Exploradores, afectando o seu comportamento.

O comportamento de solicitação de classificação dos conceitos, permite por cada conceito reconhecido fazer uma tentativa de classificação por consulta ao agente catalogador, detentor de todas as regras de classificação dos conceitos referentes à ontologia utilizada. A pergunta enviada ao catalogador contém uma palavra-chave

composta, calculada tendo em consideração o URL da página, e as informações recolhidas sobre o conceito. As regras de definição da palavra-chave são definidas uma vez mais com o recurso às regras «se-então» (e. g., se for identificada a referência de produto junto à palavra-chave). Desta forma, assim que a informação sobre um conceito fica definida, a palavra-chave composta é construída, com o recurso às regras «se-então» armazenadas na base de conhecimento do Explorador. A palavra-chave composta é utilizada pelo catalogador na pesquisa da ontologia para identificação do conceito.

O comportamento de envio de informação para os sistemas a montante permite o processamento das respostas enviadas pelo Catalogador. Por cada solicitação de classificação de conceito, o Explorador recebe do catalogador um dos seguintes tipos de resposta:

- o conceito é identificado como pertencente a uma única classe, o que desencadeia o envio imediato do conceito e toda a informação recolhida para o catálogo;
- o conceito é identificado como pertencente a um conjunto de classes, não permitindo a decisão automática, o que obriga ao recurso da interface de classificação manual;
- o conceito não é reconhecido, o que uma vez mais força o envio para a interface de classificação manual, sem qualquer sugestão.

O Explorador ignora as limitações visuais definidas pelo utilizador à ontologia, reportando ao catálogo todos os conceitos identificados; desta forma, o Explorador não elimina conceitos que por definição do utilizador não são num dado momento relevantes, permitindo a existência de dados sempre que o utilizador altera a configuração da ontologia visível.

A interface desenvolvida para o Explorador, apresentada na Figura 26, disponibiliza duas zonas principais: *i)* listagem dos documentos analisados; e ii) listagem dos conceitos identificados e classificação atribuída por documento analisado.

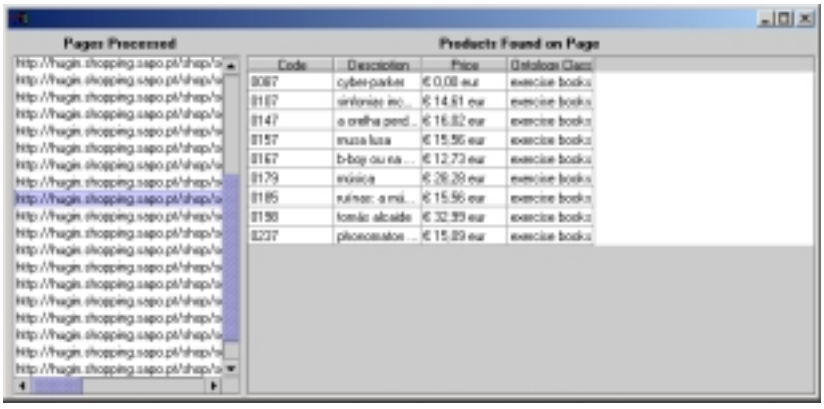


Figura 26 – Interface do Explorador, permite a listagem de todos os documentos analisados e a listagem por documento, dos conceitos identificados e da classificação atribuída

A zona esquerda da janela apresenta a listagem dos documentos analisados, sendo possível a sua consulta directa por selecção da linha através da evocação do navegador Internet instalado por omissão no posto de trabalho. Desta forma, o utilizador pode consultar a validade da análise realizada, por comparação com os dados do documento e pelos dados coligidos. Os dados extraídos para cada documento são listados na zona direita da janela.

4.1.2.3 Agente Catalogador

O catalogador, apresentado na Figura 25, tem como função estimar a classificação dos conceitos identificados pelo Agente Explorador.

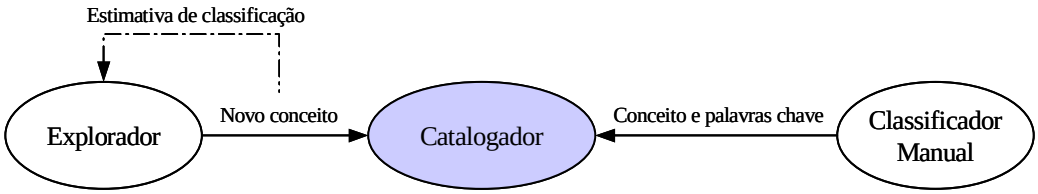


Figura 27 – Representação lógica do Catalogador, dos agentes com que interage e respectivas mensagens.

O seu **comportamento genérico** caracteriza-se por aguardar mensagens de agentes Exploradores, que indagam sobre a classificação de conceitos reconhecidos. A mensagem de pedido de classificação enviada pelos Exploradores contém uma palavra-chave composta que é utilizada para pesquisar a ontologia com vista à classificação do conceito. Os resultados obtidos são consequentemente comunicados ao Explorador.

O agente catalogador exibe um único comportamento-base: classificação de conceitos. A capacidade de reconhecimento dos conceitos reside na sua descrição prévia na ontologia, pela introdução de um conjunto de palavras-chave, para comparação com a chave composta. A descrição dos conceitos é extremamente relevante pois define a qualidade das futuras respostas do agente. Quanto mais completas e disjuntas forem as listas das palavras identificadoras do conceito, mais exacta será a resposta fornecida pelo agente. A base de dados de conhecimento do agente é enriquecida com a utilização do sistema, através da interface de classificação manual, uma vez que as correcções, ou novas palavras-chave identificadas são enviadas para os catalogadores, o que permite o aumento do seu desempenho.

A estimativa da classificação está relacionada com o número de conceitos identificados, através da palavra-chave composta, sendo possível a atribuição de nenhuma, uma, ou várias classificações. Assume-se que podem existir diversos conceitos com a mesma palavra-chave, o que desencadeia obrigatoriamente, uma classificação múltipla do conceito, conduzindo inevitavelmente ao envio do conceito para a interface de classificação por intervenção humana.

4.1.2.4 Agente interface do classificador manual

Este agente é responsável por realizar a interface entre o MAS e a aplicação de classificação manual. **O comportamento genérico** deste agente caracteriza-se pela recepção de mensagens que contêm conceitos ambíguos ou não reconhecidos. Compete a este agente enviar a informação recolhida para a interface manual e receber, em resposta, os conceitos devidamente classificados por operadores. A informação recebida é, consequentemente, transmitida para o catálogo e para o agente Catalogador, permitindo o futuro reconhecimento de conceitos semelhantes. A Figura 28 realiza o enquadramento do agente.

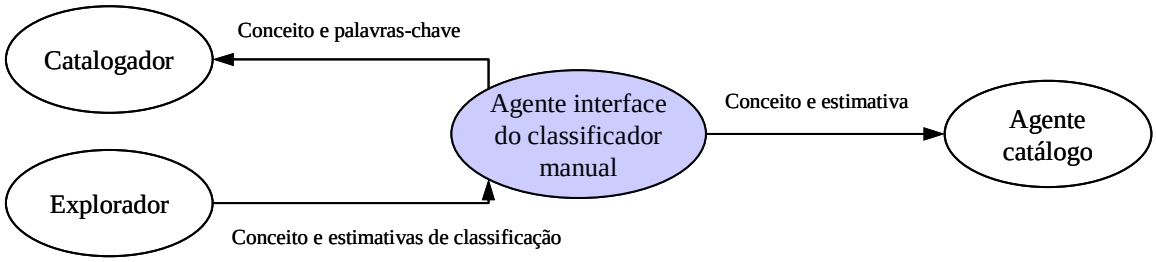


Figura 28 – Representação lógica do agente interface do Classificador, dos agentes com que interage e respectivas mensagens.

A interacção deste agente com o MAS realiza-se, assim, por recepção de mensagens do Explorador, com os conceitos que não foram automaticamente catalogados e por envio de mensagens ao agente interface de catálogo e agente Catalogador. As mensagens para o agente Catálogo enriquecem o catálogo com novos conceitos classificados manualmente, enquanto que as mensagens para o agente catalogador enriquecem o conhecimento do catalogador sobre o contexto, pelo envio de novas palavras-chave de conceitos. Fica assim assegurado que os conceitos não reconhecidos pelo sistema não só são classificados, como contribuem para melhorar o desempenho global do sistema, uma vez que a adição de novas palavras-chave permitem o automático reconhecimento dos conceitos em causa.

A Figura 29 apresenta a interface de classificação manual que é apresentada por cada conceito desconhecido pelo MAS.

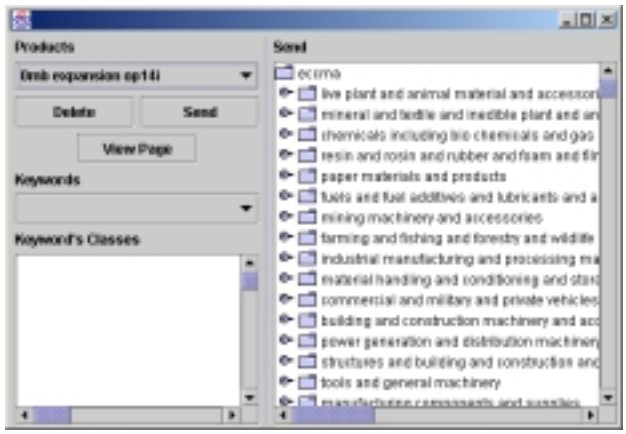


Figura 29 – Interface de classificação manual

As operações possíveis são a consulta do documento onde o conceito foi identificado, a eliminação do conceito, ou a confirmação da descrição do conceito que desencadeia o envio da informação de novo para o MAS. A descrição do conceito é produzida pela selecção das palavras enviadas pelo MAS, (as palavras associadas pelo Explorador ao conceito), pela adição opcional de novas, e pela escolha do conceito da ontologia. As operações descritas estão disponíveis numa única janela, o que reduz o esforço do utilizador.

4.1.2.5 O Centro de Mensagens (CM)

Este agente não foi referido anteriormente para simplificar a descrição da arquitectura e por ser encarado como um agente de sistema, competindo-lhe assegurar a troca de mensagens através de um conjunto de mecanismos de nível de abstracção superior aos disponibilizados pelas plataformas de agentes. A arquitectura de referência prevê um conjunto de necessidades de comunicação que ultrapassam as funcionalidades oferecidas pelas plataformas conformes com as normas FIPA, que se limitam ao registo e localização de agentes e ao envio directo de mensagens. A inexistência de mecanismos de envio de mensagens para grupos de agentes, (e. g., para um tipo de agentes) ou para agentes inactivos obrigaram à criação deste agente intermediário.

O CM tem assim, como responsabilidade, a manutenção do registo dos tipos de agentes, dos agentes activos, da intermediação das mensagens e da comunicação com outros CM instalados em plataformas remotas. Consequentemente, toda a comunicação entre agentes é efectuada via CM.

Por cada CM existe ainda um servidor de documentos que tem como função receber os documentos seleccionados pelos Navegadores para os Exploradores. Estes servidores, criados por questões relacionadas com a implementação, diminuem a dimensão das mensagens, evitando o congestionamento do CM. Deste modo, os Navegadores, em vez de enviarem os documentos seleccionados e respectivas estimativas atribuídas por

mensagens, realizam as seguintes operações: *i)* carregamento do documento no servidor; e *ii)* envio de uma mensagem com a localização do documento e a estimativa atribuída.

No momento de activação de cada plataforma, um e só um CM é automaticamente activado tendo como parâmetros de configuração mais relevantes, os tipos de agentes, de serviços e de mensagens existentes no sistema. O tipo de agentes permite a criação de grupo; o tipo de classe de mensagens a declaração da intenção da sua recepção e. g., os navegadores podem informar do seu interesse de receberem mensagem sobre páginas de arte contemporânea; e o tipo de serviços o registo de capacidades.

Sempre que um agente é activado regista-se no CM, declarando o seu tipo, os serviços que presta e a intenção de recepção de um ou vários tipos de mensagens. Compete igualmente a cada agente informar o CM do abandono do sistema. O CM mantém-se, desta forma, informado do seu universo de agentes, reunindo um registo de endereços e informações que cada agente entende publicitar.

Para além das mensagens de registo e sua manutenção, os agentes enviam ao CM todas as mensagens cujo destinatário são comunidades de agentes. As mensagens podem ser enviadas por difusão, sendo entregues a todos os agentes activos da comunidade; ou para um agente da comunidade cabendo ao CM a decisão do destinatário.

As mensagens de difusão são úteis para processos de actualização globais, por exemplo actualização de SAD, enquanto, as mensagens para agentes de uma comunidade, permitem ao remetente não ter que conhecer à partida o(s) destinatário(s) final(is), (permitindo por exemplo o envio de mensagens para especialistas), evitando todavia um excessivo tráfego de mensagem de controlo entre agentes. Desta forma, a título de exemplo, o navegador não tem que saber qual será ou serão os Exploradores encarregues de processar as páginas que classificou como relevantes.

A selecção dos destinatários pode ser realizada tendo em conta um processo de optimização de recursos, por entrega de mensagem somente a agentes livres. Todavia, foi implementada uma solução mais simples, baseada numa filosofia circular. As mensagens são distribuídas de forma sistemática e sequencial, entre os agentes que declararam a intenção de recepção do tipo mensagem. Cada mensagem recebida é entregue a um e só um agente que é posto no fim da fila.

4.1.2.6 Reinterpretação do fluxo de informação

Uma vez concluída a apresentação dos agentes é agora possível descrever todo o fluxo de informação desencadeado por uma pesquisa autónoma, apresentado na Figura 30, em que as linhas contínuas identificam mensagens entre os agentes (ACL), enquanto as linhas a tracejado identificam a transferência de documentos.

Por cada nova solicitação de monitoração o Navegador escolhido inicia a análise dos documentos, transferindo os relevantes para os concentradores de páginas activos e comunicando o facto à comunidade de Exploradores (que declararam capacidade de interpretação do tema estimado), pelo envio de uma mensagem. A mensagem inclui a localização e classificação atribuída para o documento em causa.

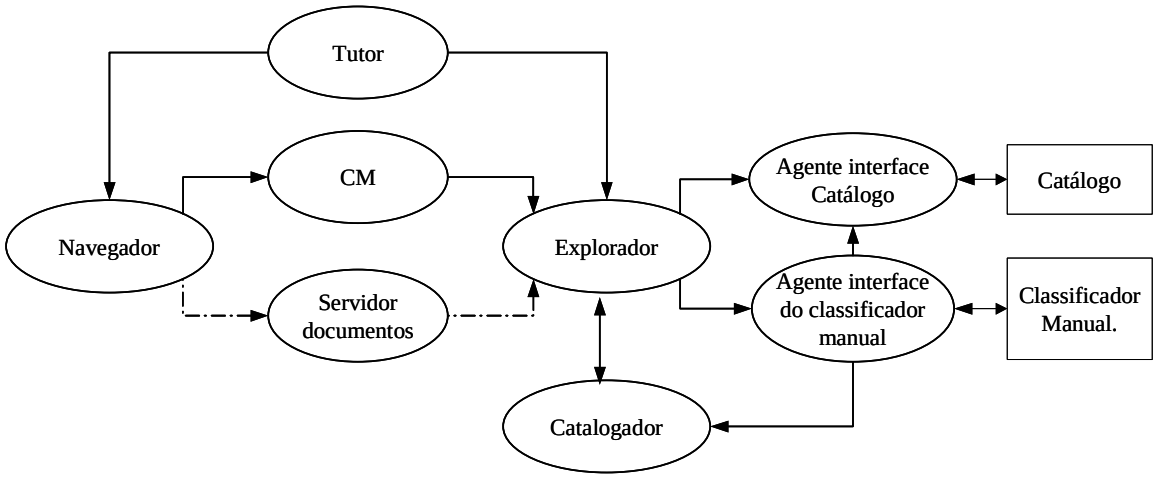


Figura 30 – Representação do fluxo de informação no MAS

A recepção de uma mensagem por parte do Explorador desencadeia o processo de carregamento da página, do servidor mais próximo, e do início da extracção de informação pela aplicação das regras existentes no seu SAD. O reconhecimento de cada conceito, desencadeia uma interacção com a comunidade de Catalogadores numa tentativa de classificação, pelo envio de uma mensagem com a palavra-chave composta determinada.

O Catalogador que recebe a mensagem utiliza a palavra-chave composta para pesquisar a ontologia e seleccionar a(s) classe(s) mais adequada(s), enviando em resposta ao Explorador o conjunto de classes seleccionadas.

No caso da identificação dos conceitos ser unívoca, a informação recolhida é enviada directamente para a comunidade de agentes interface de Catálogo, caso contrário, a informação e o conjunto de classes seleccionado são enviados para um agente interface de classificador manual.

Após o reconhecimento e correcta classificação do conceito, com a consequente verificação das palavras-chave, a informação é transferida para a comunidade de catálogos e as novas palavras-chave e conceitos associados são enviados por difusão para a comunidade de Catalogadores.

Todas as mensagens enviadas para comunidades são encaminhadas para o CM, que se encarrega do seu reencaminhamento ou eventual armazenamento até que existam agentes destinatários.

Em paralelo, o agente Tutor, que concentra todas as funcionalidades de criação de DSSs, pode a qualquer momento actualizar as funcionalidades de um conjunto de agentes pelo envio de mensagens ACL que transportam a informação. Desta forma foi possível construir um sistema que apresenta um comportamento global de recuperação e extracção de informação, que pode ser alterado sempre que necessário de forma simples e flexível.

4.1.3 As interfaces do sistema de catalogação

O sistema de catalogação dispõe de duas interfaces humano-máquina (IHM) principais: *i)* o Catálogo dinâmico; e *ii)* a Interface avançada, associadas, respectivamente, ao perfil de utilizador regular (*Regular User*) e ao perfil de super-utilizador *(Power User)*. Ver Figura 31.

O catálogo dinâmico disponibiliza o acesso a todos os métodos de pesquisas e consultas através de uma interface intuitiva, facilmente utilizável por utilizadores com conhecimentos básicos em informática.

A Interface avançada visa permitir o controlo e a monitorização dos agentes do subsistema de pesquisa autónoma permitindo a observação do seu progresso. Esta interface é dedicada a utilizadores com capacidades avançadas de Tecnologias de Informação, e foi desenvolvida com dois objectivos: *i)* permitir uma fácil monitorização do sistema; *ii)* demonstrar em tempo real o fluxo de informação do sistema autónomo de pesquisa.

Figura 31 – As interfaces do sistema de catalogação.

A entrada do sistema é única, (ver janela da esquerda da Figura 31) e permite, no momento da solicitação de acesso ao sistema, seleccionar o perfil de utilizador e consequentemente o tipo de interface (para além de introduzir o nome de utilizador e a palavra-chave é igualmente solicitada na definição do perfil desejado).

A selecção do perfil regular vai, directamente para **o catálogo dinâmico**, enquanto a selecção do perfil de super-utilizador conduz para a **interface avançada,** de modo a permitir acompanhar a dinâmica do sistema de pesquisa autónoma.

A possibilidade de criar diversas interfaces para o mesmo sistema aplicacional foi facilitada devido à separação entre os módulos desenvolvidos e as suas interfaces. Nenhum módulo no sistema catalogador foi desenvolvido contendo a sua interface incorporada. Optou-se pela definição de um protocolo de troca de mensagem entre as interfaces e os diferentes componentes e agentes que compõem o sistema, permitindo assim a utilização dos diversos módulos em diferentes contextos.

Cada módulo ou agente disponibiliza um conjunto de mensagens que permite a interrogação sobre o seu estado, comportamento e desempenho. Desta forma, cabe às interfaces de monitoração a responsabilidade de apresentar o estado dos agentes apresentando a informação no formato mais adequado.

Esta solução permite que a construção de novas ferramentas de monitoração, por exemplo, para um novo dispositivo, esteja confinada ao desenvolvimento da interface de comunicação e de apresentação de informação (e. g., dispositivo móvel, uma ferramenta legada).

4.1.3.1 O Catálogo dinâmico

O catálogo dinâmico é a interface-base para o utilizador final que permite o acesso, de forma intuitiva, às capacidades de pesquisa (directa, autónoma), consultas locais, assim como, às funcionalidades-base de personalização, de histórico e de configuração geral, representados na Figura 32.

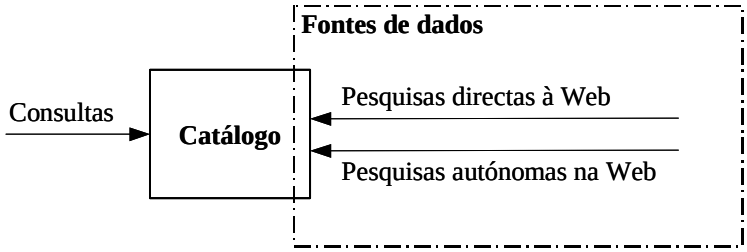


Figura 32 – Operações-base sobre o catálogo

Tendo em consideração que as pesquisas são efectuadas através de sistemas de agentes, o catálogo possui um agente de interface, que permite garantir o encapsulamento do sistema de multiagentes, assim como assegurar que para o MAS, o catálogo não é mais do que um agente com o qual é necessário interagir.

Esta opção permite a reutilização de catálogos existentes no mercado, (numa perspectiva de integração de sistemas legados), bastando para tal a incorporação do agente de interface.

Este agente de interface, apresentado na Figura 33, é assim responsável por:

- receber as mensagens enviadas pelo MAS que têm como destinatário o catálogo e realizar as actualizações na base de dados de conceitos do catálogo;

- aceitar todos os pedidos de interacção realizados pelos utilizadores via catálogo, realizar a sua tradução em mensagens e enviar as mesmas para o destinatário final. Os pedidos dos utilizadores são comunicados via base de dados operacionais.

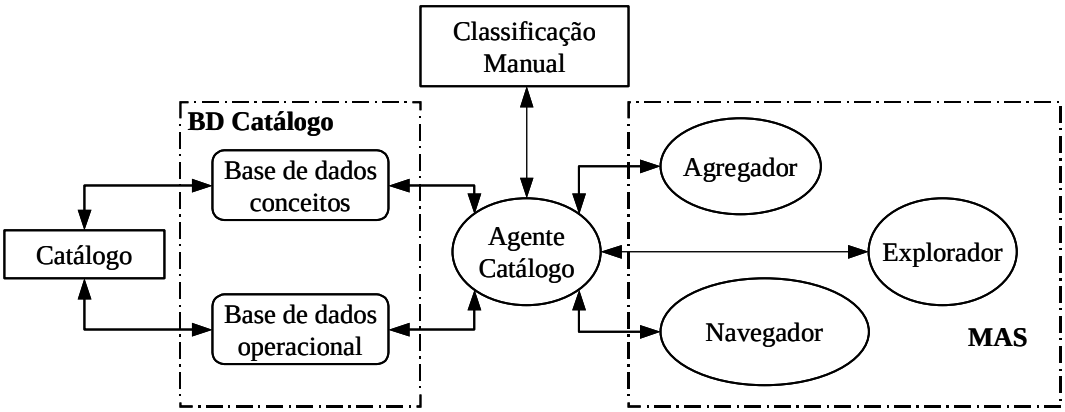


Figura 33 – Apresentação lógica do agente interface de catálogo

Os pedidos solicitados ao sistema autónomo de pesquisa são conduzidos para os agentes Navegadores, sendo as respostas recebidas através dos agentes Exploradores. Por sua vez, os pedidos solicitados ao sistema de pesquisa directa, são conduzidos ao agente agregador que é responsável, igualmente, por retornar as respostas, à excepção dos conceitos que são classificados através da interface de classificação manual, que são devolvidos directamente pelo agente Catálogo. Finalmente, os pedidos de consulta local são tratados no catálogo não implicando interacção com o sistema de agentes.

A repetição de pesquisas autónomas e directas pode conduzir à recolha de dados redundantes sobre o mesmo conceito. No caso da informação ser idêntica, por exemplo resultante da repetição de uma pesquisa autónoma sobre o mesmo sítio, permanece o registo mais actualizado na base de dados. No caso da informação corresponder à alteração da descrição do conceito, e. g., alteração de atributos, é gerado um novo registo. Desta forma, visa-se evitar a redundância de dados, e ao mesmo tempo a construção de um histórico de produtos, o que abre possibilidades futuras de exploração de dados.

As pesquisas e consultas estão baseadas na navegação da ontologia, tendo contudo a implementação ficado limitada a uma navegação na relação de herança, o que elimina muitas das vantagens potenciais das ontologias, e. g., navegação entre relações de semelhança, dependência, sinónimo.

A interface de consulta é a interface primária de todo o catálogo, uma vez que o conceito-base da arquitectura visa a substituição das pesquisas na Internet, por pesquisas locais que permitem, de forma expedita e organizada, disponibilizar todos os dados necessários ao utilizador. O catálogo permite a consulta directa à BD de conceitos que é enriquecida pela execução de pesquisas autónomas e directas. Ao contrário das pesquisas

autónoma e directa, a pesquisa local, tal como o nome indica, está limitada à BD de conceitos do catálogo, evitando interacção com o exterior. A pesqusia dos dados classificados e armazenados na BD é realizada por navegação na ontologia. Uma vez mais, cabe ao utilizador definir a especificidade da pesquisa pelo nível de profundidade do grafo de conceitos (e. g., Serviços, Serviços de Segurança, Serviços de Segurança Permanente). Quanto mais específica for a questão menor será, naturalmente, o conjunto de conceitos apresentados ao utilizador para comparação.

O catálogo permite ainda o **acesso à Interface para a pesquisa directa** onde o utilizador pode seleccionar o conceito e os PIAs a consultar. A cada nova pesquisa, corresponde uma mensagem de controlo na BD operacional do catálogo, que é posteriormente processada pelo agente Catálogo. É este agente que envia uma mensagem para o agente Agregador que faz a sua disseminação por todos os agentes PIA activos, desencadeando as respectivas pesquisas directas às DB locais de cada sítio.

Por cada conceito identificado é enviada uma mensagem ao agente agregador, que se encarrega de comunicar com o agente Catálogo para carregamento na BD de conceitos do catálogo.

As mensagens de resposta são pós-processadas, pelo agente Catálogo, sendo transformadas em registos na base de dados local, no nó da ontologia correspondente, para futuras consultas locais ao catálogo, sendo, igualmente, apresentadas na zona de resultados da pesquisa directa.

O catálogo permite, ainda, o **acesso à interface de pesquisa autónoma** sobre um conjunto de sítios Internet a ocorrer de forma cíclica num intervalo de tempo. Nesta interface, cabe ao utilizador, identificar o conjunto de sítios que quer ver monitorizados pelo MAS e desencadear a pesquisa. O pedido é registado na BD operacional do catálogo. É da responsabilidade do agente interface de catálogo a identificação de um novo pedido, processamento da informação e envio de uma mensagem para o Navegador do utilizador (reservado previamente).

A sequência natural dos acontecimentos conduz ao desencadear de uma nova operação de pesquisa, respeitando o intervalo de tempo solicitado pelo utilizador. A cada nova pesquisa, corresponde uma nova mensagem e a cada nova mensagem corresponde um novo ciclo de análise do conjunto de sítios solicitados e recepção como resposta, de um conjunto de mensagem com os conceitos identificados. As mensagens recebidas descrevem os conceitos identificados e classificados pelo MAS e são pós-processadas com o objectivo de se transformarem em registos de BD de conceitos do catálogo. Em consequência a uma ordem de pesquisa autónoma o agente Catálogo recebe um conjunto, normalmente

massivo, de mensagens, correspondente aos conceitos encontrados, que permitem enriquecer o catálogo.

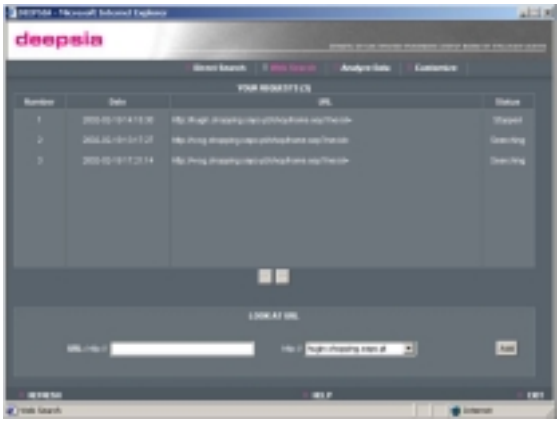


Figura 34 – Interface do catálogo dedicada à pesquisa automática

4.1.3.2 Interface Avançada

Como já foi referido a outra interface do sistema de Catalogação é a **Interface Avançada** que permite o controlo e monitoração de um subconjunto de agentes responsáveis por uma pesquisa autónoma específica. Esta interface permite acompanhar todo o processo, disponibilizando ao utilizador informações sobre o estado da pesquisa, as mensagens trocadas entre os agentes, e o tipo e a qualidade da informação recolhida.

Com o objectivo de evitar a criação de diversos agentes interface dedicados, optou-se pela criação de um único agente que fornece interfaces para todos os agentes do MAS. Cabe a este agente construir as mensagens específicas para cada agente-alvo, receber as respostas e criar a interface adequada. A Figura 35 apresenta a arquitectura lógica adoptada.

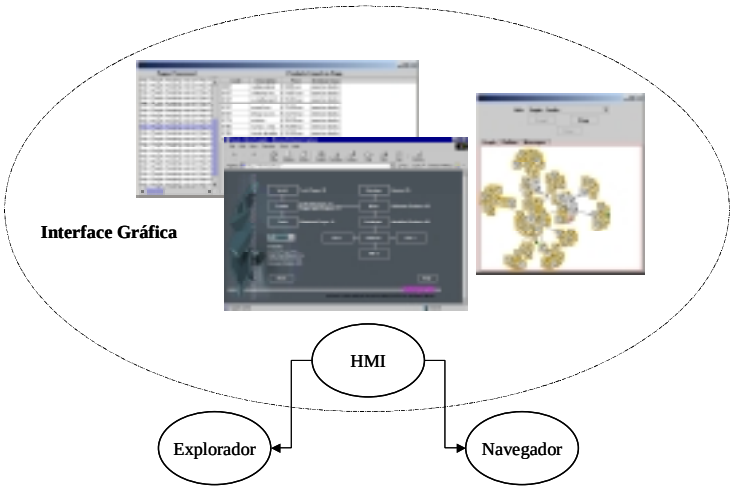


Figura 35 – Representação lógica da Interface gráfica do sistema

A interface, representada na Figura 36, apresenta, de forma esquemática, os principais agentes intervenientes numa pesquisa autónoma específica, e valores sobre as principais trocas de informação, o que permite analisar a dinâmica do sistema em tempo real.

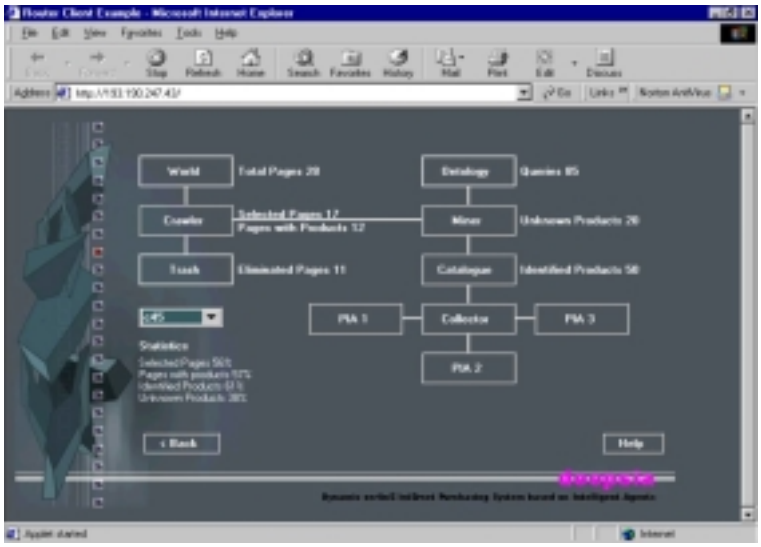


Figura 36 – Interface principal do agente HMI

Os valores apresentados quantificam:

- o número total de páginas processadas pelo sistema desde a sua activação;
- o número de páginas que foram classificadas como não interessantes pelo Navegador, não tendo sido consideradas para análise posterior pelo Explorador;
- o número páginas que foram classificadas como interessantes pelo Navegador, possuíam de facto conceitos relevantes, pela análise do Explorador;
- o número de perguntas efectuadas pelo Explorador ao Catalogador para tentativa de classificação dos conceitos identificados;
- o número de conceitos não reconhecidos pelo Catalogador e enviados para a interface de classificação manual;
- o número de conceitos reconhecidos e classificados pelo Agente Catalogador;

Estes valores são produzidos pelo Navegador e Explorador, que se encarregam de enviar, através de mensagens, os valores para a interface principal. O Navegador produz o número total de páginas, as páginas eliminadas e as páginas seleccionadas. O Explorador produz o número de páginas com conceitos, questões efectuadas ao Catalogador e conceitos reconhecidos e não reconhecidos.

Os seguintes valores estatísticos básicos são calculados tendo por base a informação recebida:

- a percentagem de páginas seleccionadas pelo Navegador para análise posterior;

- a percentagem de páginas seleccionadas que foram identificadas como contendo conceitos pesquisados;
- a percentagem de conceitos classificados pelo Catalogador.

A interface principal permite ainda, o acesso às interfaces gráficas do Navegador e do Explorador, pela selecção da área que representa o respectivo agente.

4.2 Sistema de Apoio à derivação de sistemas particulares

O sistema de apoio à derivação de sistemas particulares, consubstanciado no agente Tutor, auxilia à realização das tarefas da metodologia específica de suporte à derivação de sistemas particulares.

As principais tarefas são: *i)* a definição da ontologia de domínio; *ii)* a preparação dos dados e a aquisição de conhecimento (por inferência ou descrição de regras); e *iii)* a transferência dos resultados para os agentes destinatários (Navegador, Explorador e Catalogador).

A estratégia de concentrar todas as actividades referidas no agente Tutor, permitiu manter os restantes agentes ágeis e focados nas suas actividades principais.

As tarefas são executadas de forma autónoma e independente do sistema de Catalogação, o que permite a sua operação, mesmo quando o Tutor está activo. O novo conhecimento é transferido para os respectivos agentes por mensagem, assegurando o esperado aumento do seu desempenho. A Figura 37 permite identificar a transferência de informação do Tutor para os agentes Navegador, Explorador e Catalogador.

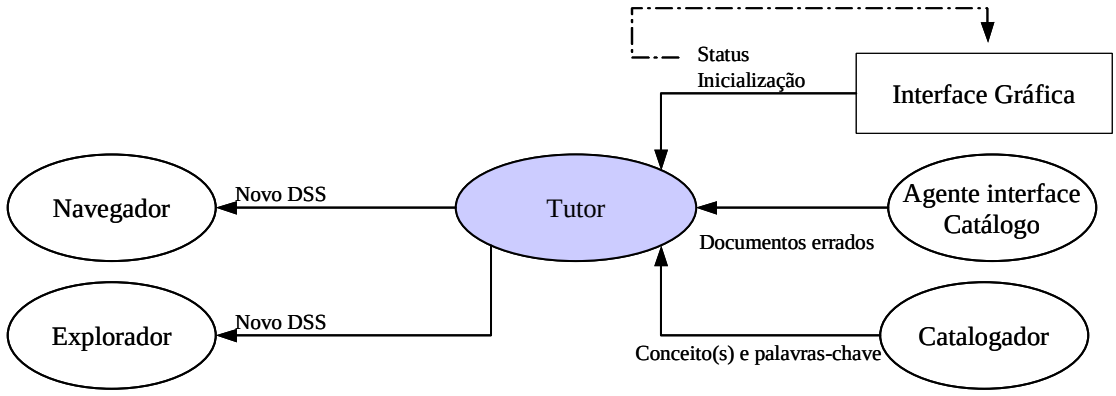


Figura 37 – Representação lógica do mecanismo de actualização do conhecimento ao Navegador, Explorador e Catalogador.

As mensagens enviadas contêm as alterações aos SAD em formato XML, o que permite o seu armazenamento directo em ficheiros locais. Uma vez mais, a interface gráfica é uma aplicação externa o que permite a sua criação para diversas plataformas tendo, no entanto, sido desenvolvida somente uma interface para Web.

4.2.1 Definição da ontologia de representação de domínio

A primeira tarefa a realizar, de forma a derivar um sistema particular, está relacionada com a representação do conhecimento de cada sistema particular e é efectuada pela definição da ontologia de domínio. É necessário descrever dois domínios de conhecimento: *i)* conhecimento sobre os assuntos considerados relevantes; *ii)* conhecimento sobre os conceitos catalogáveis. A sua especificação é encarada como uma acção de importação de duas ontologias de domínio. O domínio sobre assuntos relevantes é utilizado, essencialmente, no âmbito do agente Navegador, na sua fase de treino, como base de classificação do *corpus* e, na fase de produção, para classificação. O domínio de conceitos catalogáveis, é empregue em todo o sistema, estando presente desde a interface de consulta, até aos agentes que têm como função identificar e reconhecer os conceitos catalogáveis. O Protégé é utilizado como ferramenta de construção e manutenção da ontologia. A importação da ontologia para o sistema é realizada por acesso ao Protégé, sendo, posteriormente, exportada para uma base de dados relacional, permitindo, assim, acessos mais eficientes.

Somente após a definição consistente da ontologia é possível iniciar as seguintes acções de derivação, que podem ser efectuadas em paralelo.

4.2.2 Indução do SAD para os Navegadores

A criação do SAD para os Navegadores é realizada com o recurso a técnicas de aprendizagem supervisionada em texto, apresentadas na Figura 38, cabendo, assim, auxiliar a execução das seguintes tarefas:

- **Criação da base de dados:** pela disponibilização de uma ferramenta que permite a navegação, selecção e classificação de páginas, assegurando o seu armazenamento numa base de dados, posteriormente utilizada pelos algoritmos de aprendizagem;
- **Selecção de características:** através de algoritmos que permitem fazer a selecção das características mais relevantes para o problema em estudo, permitindo assim a eliminação de características redundantes, desnecessárias e responsáveis por ruído;
- **Indução de classificadores:** pela oferta de um conjunto de algoritmos de indução de classificadores que podem ser utilizados na criação de sistemas de decisão;
- **Indução de SAD:** pela disponibilização de ferramentas para criar e determinar qual o SAD que melhor combina os classificadores criados;

O método proposto utiliza diversas técnicas e está construído para poder incorporar novas capacidades.

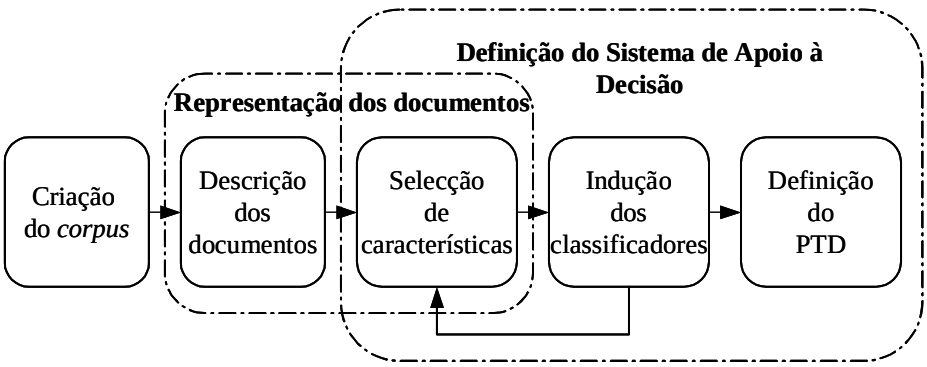


Figura 38 – Etapas do modelo conceptual do processo de aprendizagem

O Tutor utiliza em todos os processos de aprendizagem, a mesma metodologia de avaliação de desempenho definida por um conjunto de **regras de manipulação do *corpus***, de um **conjunto de métricas único** e de um **processo de generalização das estimativas**.

O processo de avaliação de desempenho dos algoritmos de aprendizagem desempenha um papel crucial, tanto para quem desenvolve o sistema, como para os utilizadores. Neste sentido, adoptou-se uma metodologia que permite extrapolar valores indicativos do desempenho das soluções propostas. Esta avaliação é importante para prever o futuro desempenho na aplicação dos métodos em estudo, e para auxiliar a selecção do modelo mais adequado [40]. Todavia, é necessário manter presente que não foi possível identificar um método isento de falha [219, 220]. A metodologia adoptada é usualmente utilizada para a avaliação de sistemas de recuperação de informação, usufruindo assim de um período de maturação que ultrapassa os trinta anos. A sua consolidação tendo sido especialmente realizada durante as diversas conferências Text Retrieval Conferences (TREC) [221], onde contribuições substanciais têm sido realizadas no desenvolvimento de metodologias, métricas, e na criação de *corpa* representativos.

Regras de manipulação do *corpus*

As regras adoptadas para a manipulação do *corpus* de dados procuram uma separação clara entre dados utilizados no processo de inferência e dados para determinação do desempenho. Tendo em consideração este desiderato, sendo J o conjunto que contém todas as observações do *corpus*, os dados são divididos em duas partes: Conjunto de Desenho D e Conjunto de Teste T, tal como está representado na Figura 39. O conjunto de treino é, então, usado exclusivamente para avaliação do desempenho do modelo construído com D, não podendo ser utilizado, em caso algum, no processo de indução. A divisão em conjunto de desenho e conjunto de teste, potencia a obtenção de um modelo com elevada capacidade de generalização e assegura a confiança na generalização determinada [222].

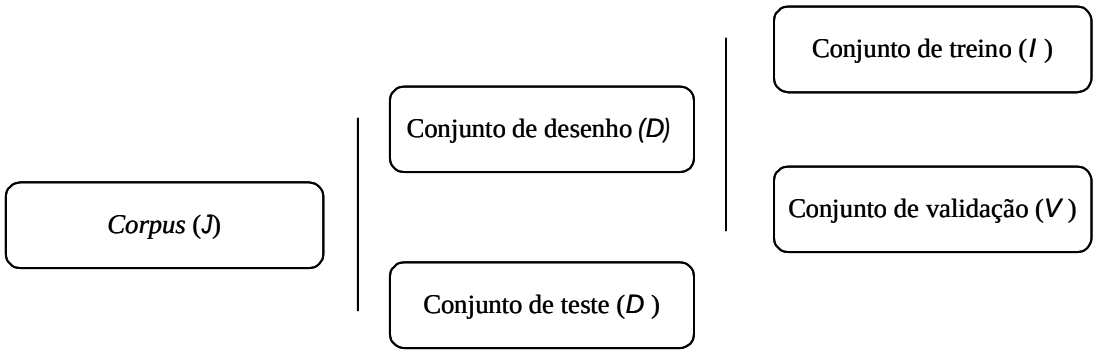


Figura 39 – Representação gráfica da divisão do *corpus* para utilização nos processos de aprendizagem

Nos casos em que os algoritmos de indução necessitam de realizar avaliações intercalares do desempenho para correcção de parâmetros ou selecção de modelos intermédios, realiza-se a subdivisão do conjunto *D* em duas partes: conjunto de Treino *I* e conjunto de validação *V* . O conjunto *I* é, então, utilizado para o treino dos parâmetros do modelo, sendo o conjunto *V* utilizado para estimar a generalização do desempenho dos modelos intermédios, sempre que necessário.

Conjunto de métricas de desempenho

As métricas de desempenho utilizadas com o objectivo de avaliar o sistema na perspectiva dos utilizadores são a precisão, equação (6), e rechamada, equação (7), que permitem, respectivamente, determinar a capacidade de apresentar somente os resultados apropriados e a capacidade de identificar todos os resultados relevantes.

A métrica F_1, equação (9), foi adoptada como métrica de combinação da precisão e rechamada atribuindo, desta forma, igual relevância às duas medidas.

Processo de generalização das estimativas

Para a estimativa do erro adoptou-se o princípio-base de amostragens sucessivas, genericamente apelidado por validação cruzada, como método para a generalização de estimativas. O método de validação cruzada, implementado por omissão, realiza o seccionamento em 10 subconjuntos.

Tendo em consideração que as métricas apresentadas resultam do cálculo médio dos valores verificados experimentalmente, sempre que necessário, optou-se por acompanhar os valores médios do correspondente desvio-padrão, permitindo uma avaliação da dispersão dos valores em relação ao valor médio. O desvio-padrão é calculado utilizando o método (n-1) ou não desviado[40], i. e., através da seguinte equação:

[40] Não desviado - Tradução do autor para «*nonbiased*»

$$desvio - padrão = \sigma(x) = \sqrt{\frac{n\sum x^2 - \left(\sum x\right)^2}{n\,(n-1)}} \qquad (57)$$

A equação (57) assume que os parâmetros provêm de uma amostra da população, sendo n a dimensão da amostra e x cada um dos valores. Esta aproximação pode parecer desadequada uma vez que se utilizam todos os valores disponíveis para o cálculo da métrica, contudo é necessário manter presente que os valores são uma amostra da população total.

4.2.2.1 Criação do Corpus

As técnicas e métodos de aprendizagem supervisionada inferem conhecimentos baseados no pressuposto que a base de dados de exemplos, o *corpus*, é representativo do universo em estudo. Este pressuposto, pode ser assegurado com graus de confiança consideráveis, no caso do universo em estudo ser finito, de dimensão limitada e amostrável. Todavia, sempre que se está na presença de um universo de infinito, assegurar a representatividade de um *corpus* passa a ser um problema delicado e a sua não representatividade conduzirá à inferência de regras de conhecimento não adequadas, redundando num fraco desempenho final do sistema. Neste sentido, é imperativo assegurar que o *corpus* representa tanto quanto possível, o universo em estudo, sendo necessária a maior atenção na sua criação, baseada numa metodologia que assegure um processo de representatividade efectivo.

A prova final da validade de um *corpus* termina, inevitavelmente, com a utilização do sistema em casos reais. Só o bom desempenho do sistema em ambiente real, permite assegurar que o *corpus* foi correctamente construído. No caso contrário, em especial se esta avaliação não for coincidente com as estimativas, são fortes os indicadores de um *corpus* não representativo, pondo em causa o interesse da realização de novos estudos e técnicas, uma vez que o problema pode residir não no processo, mas, efectivamente, na matéria em estudo.

Esta preocupação conduziu à necessidade de estabelecer um conjunto de regras gerais que permitam diminuir o risco da criação de um *corpus* que não seja representativo do universo em estudo. A aplicação das regras, não permite assegurar a representatividade do universo, todavia são um auxílio à justificação da validade do *corpus*.

Genericamente, a criação do *corpus* rege-se por capturar correctamente os dados, com uma definição sintáctica exacta, enriquecida, sempre que possível, com informação semântica e de organização.

O processo sugerido baseia-se no respeito das seguintes regras:

- **Definição sintáctica inequívoca dos dados –** que permita assegurar que existe um reconhecimento exacto do conteúdo sintáctico dos dados. Este assunto é trivial,

todavia é necessário assegurar uma política de identificação das unidades-base dos dados, no caso de dados de texto na Internet, palavras, marcas, etc.;

- **Definição semântica –** que permita assegurar o reconhecimento de conteúdos semânticos-base que permitam codificar, de forma equivalente, dados que tenham o mesmo significado semântico. Esta actividade requer sensibilidade para os dados em questão e é essencial para permitir o reconhecimento dos conceitos inerente aos dados. Um exemplo de dados equivalentes é a existência de números numa página. O valor do número é provavelmente irrelevante para o reconhecimento do tema da página, todavia a existência de um número é potencialmente relevante. Neste sentido, é importante identificar a existência de um número, mais do que o seu valor;
- **Reconhecimento de estruturas e atributos –** que permitam agrupar e/ou valorizar dados pela sua localização. No caso de dados de texto, a sua presença num destaque, num título, ou por estar representado num tipo de letra diferente;
- **Definição inequívoca dos conteúdos que fazem parte de uma classe –** que permitam classificar de forma clara se um determinado conteúdo faz parte de uma classe. O estabelecimento de regras claras e concisas evita a marcação errónea de conteúdos, o que comprometerá o desempenho global do sistema;
- **Recolha aleatória de exemplos –** realizar a construção do *corpus* através da recolha de dados baseada em amostragem aleatória, o que diminuirá a existência de desvios nos dados.

O Tutor disponibiliza ao utilizador uma ferramenta que permite apresentar os documentos em estudo, fazer a sua classificação de acordo com a ontologia de relevâncias adoptada, e adicionar o documento ao *corpus*. O documento é armazenado no formato neutro adoptado para toda a plataforma permitindo o seu reconhecimento imediato.

4.2.2.2 Representação das páginas

Para a representação dos documentos adoptou-se uma representação vectorial. As palavras são assim descritas em função de:

$$[((W_1,S_1),(W_2,S_2),\ldots,(W_N,S_N)),C_i] \tag{58}$$

em que W_i é um termo quantitativo, palavra extraída do texto, S_i é um termo qualitativo que caracteriza a relevância da palavra. A relevância da palavra pode entrar em conta com a posição da palavra, com ocorrência da palavra entre marcas HTML assim como a sua frequência.

$$S_i = C_1 P_{freq}(i) + C_2 P_{marcas}(i) + C_3 P_{posição}(i) \tag{59}$$

em que

$$P_{freq}(i) = \frac{n_f(i)}{N_{TOT}} \tag{60}$$

$$P_{marcas}(i) = \frac{\sum_{T=1}^{T=M} C_T \frac{Marca_T(i)}{N_{Marca_T}}}{N_{Marca}} \tag{61}$$

$$P_{posição}(i) = \frac{\sum_{P=1}^{P=N} C_P Posição_p(i)}{N_{Total}} \tag{62}$$

A página passa, assim, a ser representada por um vector

$$\vec{p} = (S_1,...,S_N) \tag{63}$$

em que os elementos S_i representam a palavra W_i.

4.2.2.3 Selecção de características para Catalogação de texto

Com vista a determinar as melhores características a utilizar na representação dos documentos o Tutor disponibiliza: *i)* a possibilidade de filtragem através de listas de paragem; *ii)* a capacidade de selecção por limiares de frequência superiores e inferiores; *iii)* dois métodos de ordenação e, finalmente; *iv)* um método de optimização das listas obtidas para determinação dos melhores candidatos. Compete ao utilizador decidir qual a combinação das acções a realizar que permitam obter os melhores resultados face aos dados que possui. A Figura 40 representa os diversos métodos disponibilizados.

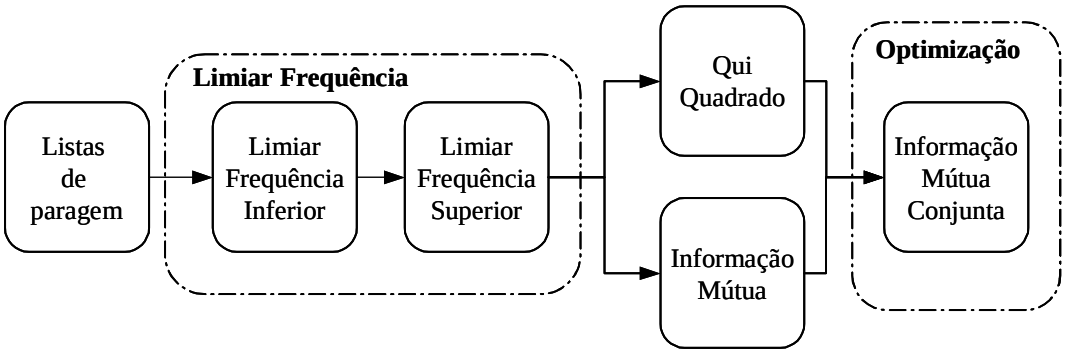


Figura 40 – Diagrama de blocos das acções possíveis no processo de selecção de características

i) Filtragem através de listas de paragem

A primeira etapa na selecção de características é baseada na eliminação pura e simples de palavras que constem de uma lista de paragem. As listas de paragem são constituídas pelas palavras mais comuns de cada língua. A manutenção destas palavras no vector de representação das páginas não contribui, naturalmente, para a realização de processos de classificação, devido à sua proliferação nos textos, independentemente do assunto versado.

Exemplos ilustrativos de palavras que constem nas listas de paragem são os artigos, preposições, etc.

***ii)* Limiar de frequência por documento[41] (LF)**

O método Limiar de frequência por documento calcula a ocorrência de cada característica e elimina as que estão fora de dois limites (máximo e mínimo) pré-definidos. Este método baseia-se no pressuposto, de que: *i)* características que apresentam baixa ocorrência não são relevantes pelo carácter avulso do seu acontecimento; ou *ii)* que características que têm elevada frequência, não são discriminativas. O LF é um dos métodos cegos implementados no Tutor, permitindo a eliminação de características de forma expedita e sem grande esforço computacional. A técnica é facilmente escalável a domínios de análise de elevada dimensão. Todavia, é considerada uma técnica *ad-hoc* principalmente para melhorar a eficiência e não um critério de princípio para realizar o pré-processamento dos dados, com vista à selecção de características, para métodos de aprendizagem.

A escolha dos limiares é determinante, pois pode conduzir a uma eliminação muito agressiva de características conduzindo à perda de eficácia se forem eliminadas características determinantes na determinação de uma variável de classe.

iii) Critérios de ordenação de características

O Tutor possui dois métodos de ordenação de características que utilizam duas aproximações-base distintas: teoria de informação e estatística.

O primeiro método implementado pertence a classe de métodos de cálculo de informação, muito utilizados em aprendizagem automática [72] [20], e visa medir a quantidade de informação que se obtém sobre a classificação de um documento, quando se conhece, *a priori,* a existência ou ausência de uma característica.

O método seleccionado, para avaliar o ganho de informação, como o critério de medida de discriminação das características de C é a Informação Mútua, apresentado na equação (20).

Reescrevendo as probabilidades, da equação (24), em função de contadores obtém-se

$$p\left(c_k, y_{i,j}\right) = p\left(c_k\right).p\left(y_{i,j} \mid c_k\right) = \frac{nc_k}{n} \quad \frac{n_{c_k, y_{i,j}}}{nc_k} = \frac{n_{c_k, y_{i,j}}}{n} \tag{64}$$

isto é,

$$p\left(c_k \mid y_{i,j}\right) = \frac{n_{c_k, y_{i,j}}}{n_{y_{i,j}}} \tag{65}$$

resultando,

[41] Limiar de frequência por documento – Tradução de «*Document frequency thresholding*»

$$H(C \mid Y_i) = -\sum_{k=1}^{|C|}\sum_{j=1}^{|Y_i|}\frac{n_{c_k,y_{i,j}}}{n} log \frac{n_{c_k,y_{i,j}}}{n_{y_{i,j}}} \quad (66)$$

onde $n_{c_k,y_{i,j}}$ é o número de ocorrências simultâneas de $c_k \in C$ e $y_{i,j} \in y_i$, n é o número total de exemplos do conjunto de treino e $n_{y_{i,j}}$ é o número de ocorrência de $y_{i,j}$.

O segundo método de ordenação disponibilizado é de natureza estatística e baseado na família do χ^2. Este método visa determinar o grau de independência entre os termos e as classes.

Fazendo a reescrita da equação (26) recorrendo, de novo, à utilização dos contadores obtém-se:

$$\chi^2(Y_i,c_j) = \frac{n_Y \times (n_{c_j=1,y_i=1} \times n_{c_j=0,y_i=0} - n_{c_j=0,y_i=1} \times n_{c_j=1,y_i=0})^2}{(n_{c_j=1,y_i=1} + n_{c_j=0,y_i=1}) \times (n_{c_j=1,y_i=0} + n_{c_j=0,y_i=0}) \times (n_{c_j=1,y_i=1} + n_{c_j=1,y_i=0}) \times (n_{c_j=0,y_i=1} + n_{c_j=0,y_i=0})} \quad (67)$$

Para além do $\chi^2_{média}(Y_i)$ o Tutor disponibiliza as seguintes variações $\chi^2_{min}(Y_i)$ e $\chi^2_{max}(Y_i)$ respectivamente para (28) e (29).

Optimização das listas ordenadas

Os critérios de ordenação anteriores, assumem a independência das variáveis, premissa que não é assegurada em documentos de textos. Após verificada esta limitação, dotou-se o Tutor de um método de optimização da ordenação de características que entra em consideração com as correlações conjuntas. Tendo por base as listas de ordenação obtidas pelos métodos anteriores, o Tutor reavalia a ordenação, eliminando as variáveis correlacionadas através da informação mútua conjunta.

Por isso, redefine-se o critério para seleccionar-se o conjunto de variáveis S de tal modo que

$$I(C;S_1,\cdots,S_K) \geq I(C;Z_1,\cdots,Z_K) \quad para\ todo\ o\ Z_i\ e\ S_j \in \Delta. \quad (68)$$

Pelas razões já apontadas este critério é equivalente a

$$H(C \mid S_1,\cdots,S_K) \leq H(C \mid Z_1,\cdots,Z_K) \qquad para\ todo\ o\ Z_i\ e\ S_j \in \Delta. \quad (69)$$

Contudo, a complexidade C_K^N deste algoritmo inviabiliza a sua aplicação directa. Uma aproximação possível para viabilizar o cálculo da informação mútua condicionada, seria a determinação de uma solução por método de pesquisa ávida, o que apesar de não considerar todas as dependências entre as variáveis seleccionadas, seria um passo intermédio.

A implementação adoptada recorre à utilização de algoritmos genéticos, permitindo a determinação de soluções admissíveis, potencialmente superiores às identificadas pelos

métodos anteriores. Todavia, é necessário ter presente que as soluções admissíveis, podem estar longe da solução óptima ou, melhor dizendo, do conjunto de soluções óptimas, uma vez que no caso da Informação Mútua Conjunta (IMC), existe um conjunto de soluções equivalentes. Este conjunto é composto por todas as combinações do melhor conjunto de variáveis, uma vez que para a IMC a ordem das variáveis é irrelevante.

Os algoritmos genéticos são, frequentemente, descritos como métodos de pesquisa global que não utilizam informação de gradiente, o que permite a sua aplicação a funções não diferenciáveis assim como a funções com diversos máximos locais. São estas características que conduziram à sua aplicação para o cálculo da IMC [223].

Numa apresentação muito genérica, os algoritmos genéticos pertencem à família de modelos computacionais inspirados pela evolução da vida na terra e codificam as soluções potenciais de um problema específico em cromossomas que, pela aplicação de operadores de re-combinação, (que preservam a informação crítica), se transformam em novas soluções admissíveis (que nalguns casos serão melhores).

A aplicação dos algoritmos genéticos inicia-se pela criação de uma população original de cromossomas, gerada usualmente de forma aleatória. Os cromossomas obtidos são então avaliados, sendo reservadas capacidades de reprodução dependentes da qualidade da solução codificada, i. e., são dadas maiores probabilidades de «multiplicação» aos cromossomas que codificam melhores soluções, em detrimento dos restantes cromossomas.

As componentes intrinsecamente dependentes do problema em estudo são: *i)* a codificação das soluções; *ii)* a função objectivo; e *iii)* a função de adaptação.

No caso concreto, **a codificação** visa a representação dos documentos tendo-se optado por representar o cromossoma como um vector de *bits*, em que a cada posição corresponde a uma característica. Assumindo uma dimensão L para os cromossomas, existem 2^L cromossomas possíveis. Quando o *bit* apresenta o valor 1, a característica está seleccionada para pertencer ao vector de representação do documento (existem somente K características nesta situação). Neste sentido, a utilização de todas as características candidatas não é admissível, uma vez que mesmo após a selecção inicial, baseada nas listas de paragem e nas técnicas de frequência, o seu número é ainda da ordem das dezenas de milhares, o que torna a dimensão do cromossoma e o espaço de pesquisa demasiado elevado para utilização. Optou-se, assim, pela utilização das melhores características, ordenadas através dos métodos da IM e QQ, o que reduz a dimensão do cromossoma e, consequentemente, ao número de soluções válidas. O facto de só serem avaliadas as melhores características, conduz a que este processo funcione como uma aproximação para eliminação variáveis correlacionadas.

O algoritmo de implementação permite definir o valor de L e de K, permitindo a realização de estudos comparativos com vista à identificação da melhor combinação para cada caso específico.

A Figura 41 apresenta um exemplo da codificação utilizada, em que a descrição dos cromossomas é efectuada pelo primeiro vector, que descreve, para cada posição, qual o índice da característica que o ocupa.

Posição	1	2	3	4	...	1023	1024
Característica	20	5	3015	548	...	478	35

Cromo. 1	1	0	0	0	...	1	0
Cromo. 2	0	1	1	0	...	1	0
			...				
Cromo. N-1	1	1	1	0	...	0	0
Cromo. N	0	0	0	0	...	0	0

Figura 41 – Exemplo da codificação utilizada com a apresentação da estrutura de dados de descrição e de um cromossoma.

No exemplo apresentado, as primeiras quatro posições são ocupadas, respectivamente, pelas características 20, 5, 3015 e 548. Os cromossomas descrevem quais as características que estão a ser consideradas para utilização no vector de representação. O cromossoma 2 utiliza as características 5, 3015 e 478, enquanto que o cromossoma N não utiliza nenhuma das características apresentadas, uma vez que estão todas a 0.

As noções de **função objectivo e de função de adaptação,** são muitas vezes utilizadas indistintamente. Nesta dissertação adoptou-se a seguinte convenção: **a função objectivo** é uma métrica absoluta de desempenho, tendo em conta os parâmetros admitidos; a **função de adaptação** é uma métrica relativa, permitindo realizar a ordenação dos cromossomas tendo, consequentemente, reflexos directos nas suas oportunidades de multiplicação. Por outras palavras, a **função de adaptação** transforma a **função objectivo**, em possibilidades de reprodução dos cromossomas, premiando os que apresentam melhor desempenho. A função objectivo de um cromossoma é, assim, independente de outros cromossomas, todavia a função de adaptação é sempre definida em função de uma população.

No caso presente, a função objectivo é directamente o valor da IMC, i. e., para cada cromossoma o seu valor de desempenho está directamente relacionado com o valor intrínseco da IMC das variáveis que estão seleccionadas para pertencerem ao vector de representação das páginas. A função de adaptação é definida pela ordenação dos

cromossomas pelo valor mais elevado do valor objectivo, i. e., são seleccionados os cromossomas que apresentem maior IMC.

O cálculo da função objectivo é o passo computacional mais demorado pois é necessário calcular o desempenho do vector codificado tendo em conta a base de dados reais. Sendo assim, e para cada cromossoma, é necessário determinar a representação da base de dados em função do vector codificado e determinar o valor da IMC.

Por questões de optimização, no início do processo realiza-se a representação da base de dados através da matriz $L \times N$, que descreve todos os exemplos em função de todas as características presentes no cromossoma. Desta forma, o cálculo do valor da IMC para cada cromossoma consiste, num primeiro passo, em realizar a compactação da matriz por eliminação das colunas não utilizadas na representação e, finalmente, no cálculo do valor por contagem dos exemplos.

O algoritmo implementado segue os passos tradicionais definidos e estudados por John Holand e seus estudantes, desde os anos sessenta, que permitem fazer evoluir uma população inicial, baseada em mecanismos de selecção, recombinação e mutação.

A população inicial é gerada aleatoriamente, tendo como parâmetros iniciais a dimensão do cromossoma, L , o número de características K que se pretende seleccionar e a dimensão da população P . Após a geração da população inicial, cada cromossoma é avaliado para atribuição do valor da função objectivo, i. e., valor da IMC, e inicia-se o processo de evolução, visando a sua transformação na população seguinte.

A definição de etapas de evolução, em função da população corrente, auxilia a explicação do processo de evolução, devido à natureza cíclica do algoritmo. A Figura 42 ilustra um ciclo de evolução de uma população, sendo possível identificar o processo de selecção, seguido de recombinação, mutação e, finalmente, a aprovação dos elementos da população intermédia que se transformará na população seguinte, reiniciando-se o processo.

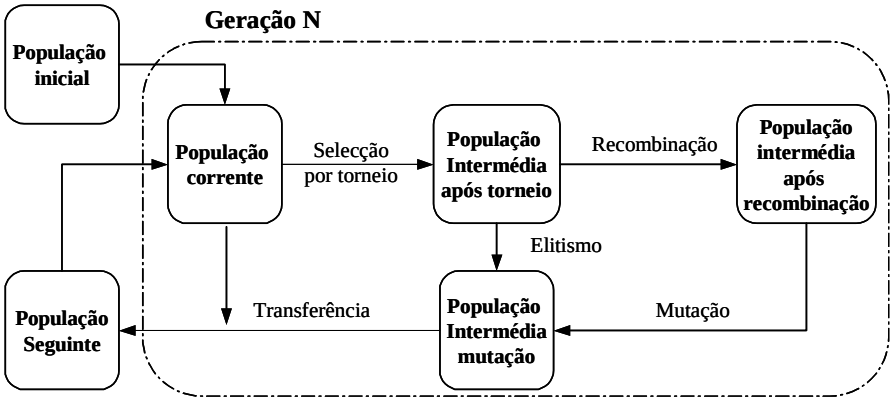


Figura 42 – Etapas do algoritmo genético implementado

O número de gerações G constitui outra variável do sistema, que pode ser definida em valor absoluto ou no número de gerações em que o valor da função objectivo do melhor cromossoma não apresenta alterações. O critério de paragem é definido através do parâmetro C onde 0 define o critério de paragem por valor absoluto de gerações, e 1 por número de gerações em que o melhor cromossoma apresenta um valor estável para a função objectivo.

A criação da população intermédia é baseada no método de amostragem, torneio. No torneio, a população corrente é amostrada por tiragem aleatória com reposição de um número T de cromossomas, que são de seguida avaliados pela função de adaptação, sendo seleccionado o melhor para fazer parte da geração intermédia. São realizados P torneios o que assegura a criação de uma população intermédia de dimensão igual à população corrente. Tendo em conta que o método de selecção é aleatório com reposição, é expectável que cada cromossoma seja seleccionado, em média, duas vezes para a população intermédia. Os melhores cromossomas vencem os dois torneios em que estão envolvidos assegurando duas cópias na geração intermédia. Os cromossomas médios vencem um, assegurando uma cópia enquanto que os cromossomas com baixo desempenho não se reproduzem sendo eliminados. Em expectativa, é assegurada uma ordenação linear, com um desvio de 2, para os melhores indivíduos, para o caso de T ser igual a 2. O desvio é tanto maior quanto maior for o valor de T. [224]

A etapa seguinte consiste na recombinação dos cromossomas presentes na população intermédia. De forma aleatória, são seleccionados pares de cromossomas que são recombinados com uma probabilidade P_c. No caso de não serem seleccionados para recombinação, os cromossomas originais são directamente copiados para a geração intermédia. No caso de ser definida a recombinação, os cromossomas são divididos através de um ponto de corte, sendo a posição de corte seleccionada aleatoriamente. Os cromossomas são reconstruídos pela troca dos fragmentos obtidos. A Figura 43 ilustra o processo, representando os valores do cromossomas de forma abstracta pelas letras $\alpha\beta\lambda\nu$, para simplificar a identificação da localização dos fragmentos.

Após a recombinação dos cromossomas é necessário assegurar que o número de *bits* presentes continua a representar uma solução admissível, i. e., que se mantêm, exactamente, K *bits* a um em cada cromossoma. Os cromossomas são avaliados e, até estar assegurada a representação de uma solução admissível, é seleccionado aleatoriamente um *bit* que esteja a um, (no cromossoma com uns a mais) e forçado a zero, (sendo assegurado o processo inverso no cromossoma complementar). A operação de recombinação é efectuada $P/2$ vezes, assegurando a manutenção da dimensão intermédia.

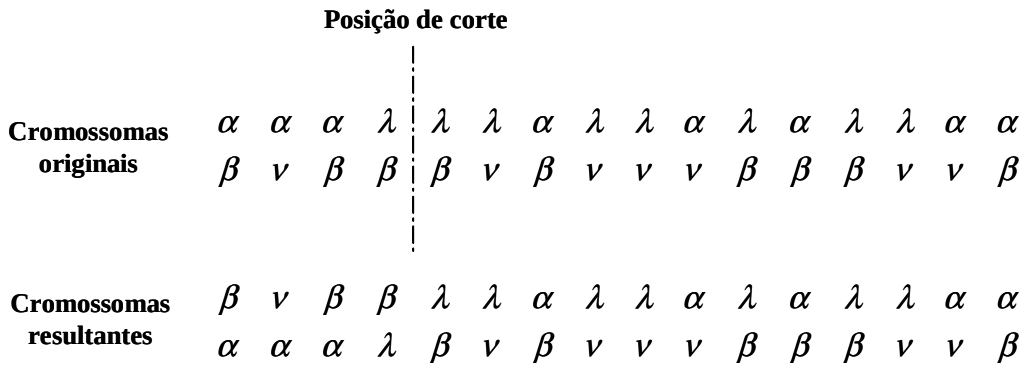


Figura 43 – Exemplo da recombinação de dois cromossomas, em função da posição de corte

A mutação é a etapa seguinte no processo de evolução da população intermédia. A mutação é efectuada com uma probabilidade P_m de realização aplicada a todos os *bits* existentes na população. O algoritmo de mutação implementado não assegura uma mutação efectiva, uma vez que a mutação de um *bit* é realizada por troca do seu valor com outro *bit* do cromossoma, permitindo assegurar a manutenção da codificação de soluções admissíveis.

Após a mutação, selecciona-se um indivíduo, aleatoriamente sem reposição, de cada uma das populações (inicial e intermédia). É transferido para a população seguinte, o indivíduo da população intermédia, com uma probabilidade P_t, sendo nos restantes casos seleccionado o indivíduo da população original. Esta operação é, naturalmente, realizada P vezes.

Para terminar, assegura-se o processo de elitismo. Antes de iniciar o processo de mutação, a população intermédia é avaliada e o melhor indivíduo é guardado, para se poder assegurar a sua transmissão à geração seguinte. A troca do melhor indivíduo efectua-se após a última etapa descrita, pela substituição de um indivíduo seleccionado aleatoriamente. Após esta troca chega-se finalmente à população seguinte, sendo possível realizar a sua avaliação.

A configuração do Tutor com vista à utilização dos algoritmos genéticos passa pela definição dos parâmetros que foram apresentados ao longo da descrição do algoritmo, e que estão resumidamente apresentados na Tabela 5 onde, para além do seu nome e da sua descrição, é possível verificar os valores assumidos por omissão.

Nome	Descrição	Valor por omissão
L	Dimensão do cromossoma (número de *bits*)	1024
K	Número de características a utilizar no vector	100
P	Dimensão da população	100
G	Número de gerações	1024
C	Critério de paragem (0 – número absoluto de gerações, 1 – número de gerações em que o melhor cromossoma apresenta um valor estável para a função objectivo)	0
T	Número de indivíduos utilizados em cada torneio	2
P_r	Probabilidade da recombinação se efectuar	0.90
P_m	Probabilidade da mutação se efectuar	0.01
P_t	Probabilidade de aceitação do indivíduo da geração intermédia para a geração seguinte	0.90

Tabela 5 – Apresentação dos parâmetros de configuração dos algoritmos genéticos com os valores por omissão.

4.2.2.4 Classificadores

Após a criação das listas de ordenação das características, o Tutor disponibiliza ferramentas que auxiliam a execução da próxima tarefa ou seja, a criação dos classificadores que permitem estimar a classificação dos documentos analisados. Foram integradas as seguintes classes de algoritmos: Vizinho mais próximo; Árvore de Decisão e Naive Bayes.

Figura 44 – Interface do Tutor para os módulos de indução de classificadores para o Navegador.

Vizinho mais próximo

O método dos k-vizinhos é um método de aprendizagem baseado em instâncias, estando o processo de aprendizagem reduzido à memorização de cada exemplo do conjunto de treino, podendo ser descrito da seguinte forma:

> ***Para cada*** *exemplo de treino (x, c(x))* ***adicionar*** *à lista de exemplos de treino.*
> ***Retorna:*** *lista de exemplos de treino.*

O processo de classificação das novas observações é descrito da seguinte forma:

K-Vizinho (y, exemplos de treino)
Sejam $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$, *pertencentes à lista de exemplos de treino, os k-vizinhos mais próximos de* $\boldsymbol{y}$.

Retorna: $\hat{c}(\boldsymbol{y}) = \arg\max_{v \in C} \sum_{i=1}^{K} \delta(v, c(\boldsymbol{z}_i))$

Para o cálculo da distância o Tutor disponibiliza duas opções: a distância euclidiana, uma das mais conhecidas, que no espaço bidimensional é equivalente ao Teorema de Pitágoras,

$$D_E(x_i, y_j) = \sqrt{\sum_{v=1}^{K} (x_{i_v}{}^2 - y_{j_v}{}^2)}, \tag{70}$$

e a distância de Hamming,

$$D_H(x_i, y_j) = \sum_{v=1}^{k} \left| x_{i_v} - y_{j_v} \right|, \tag{71}$$

O cálculo da distância de Hamming é muito eficiente e especialmente recomendado para representações vectoriais binárias, visto ser um caso particular da distância euclidiana $D_E(x_i, y_i) = \sqrt{D_H(x_i, y_i)}$.

Árvores de decisão

O Tutor disponibiliza um conjunto de algoritmos para a indução de árvores de decisão tendo por base o C4.5.

O processo utilizado pelo Tutor, para a indução de árvores de decisão, assume a existência do conjunto de atributos S, o conjunto de desenho *D* e o conjunto de classes C e é descrito pelo seguinte algoritmo:

Função *ID3 (S, D, C)* ***retorna*** árvore de decisão
Início
[A] Se D só contém observações da classe C_i retorna folha com C_i

[B] Se $S = \{\ \}$ retorna **folha com classe mais comum (indução de erro)**

[C] Seja $A_{sel} = \max_{A_j} \arg\left(Gi\left(A_j, D\right)\right)$

[D] Seja $D_l \subset D$ **tal que** D_l **contenha somente observações com** $A_{sel} = a_l$

[E] Cria **Arv com raiz** A_{sel}

[F] Para cada $A_{sel,i}$ **adicionar a Arv, arco** a_i **que articula com** $ID3\,(S - \{A_{sel}\}, D_i, C)$

Retorna*: Arv*
Fim

No caso particular de se assegurar um número de atributos que evite o passo B as árvores induzidas apresentam uma eficácia total sobre o conjunto de desenho. Todavia, as árvores

obtidas nesses casos são muitas vezes não balanceadas, longas e pouco eficazes sobre o conjunto de teste.

O processo de classificação das novas observações é descrito da seguinte forma:

> *Navegação na árvore até a uma folha e atribuição à observação, da classificação da folha. A navegação na árvore realiza-se seguindo, para cada nó, o arco correspondente ao valor do atributo da observação.*

Para além da disponibilização do C4.5 na sua versão original foi implementada uma adaptação, apelidada de C4.5 Iterativo que consiste na geração sucessiva de árvores de decisão, utilizando, em cada iteração, um novo conjunto de características S. O conjunto S é actualizado em cada iteração, pela inclusão de uma nova característica que é a melhor candidata da lista ordenada L de características. A cada iteração a nova árvore é avaliada e, no caso do seu desempenho ser inferior, a característica previamente inserida em S é removida, passando-se para a iteração seguinte.

Por outras palavras, o algoritmo C4.5 iterativo utilizado pelo Tutor assume a existência de uma lista ordenada de atributos L, de um conjunto de desenho *D* e de um conjunto de classes C, e é descrito pelo seguinte algoritmo:

> ***Função*** *C45_Iterativo (L, D, C)* ***retorna*** árvore de decisão
> *Início*
> *Seja A={ }*
> ***Seja*** *Iteração=0*
> ***Seja*** *erroPadrao=ErroMinimo=Infinito*
> ***Enquanto*** *CritérioParagem (L)*
> *Seja A=L(1)*
> *Seja S=S+A*
> ***Seja*** *arv= ID3 (S, D, C)*
> ***Se*** *Avaliação(arv)<erroPadrao ActualizaErros (erro, erroMinimo, erroPadrao)*
> *Senão*
> ***Seja*** *S=S-A*
> *ActualizaLista(L)*
> *Retorna: arv*
> *Fim*

A lista de características a utilizar é determinante, tendo em conta que a ordem passa a ter uma relevância superior por comparação ao C4.5 original, devido à remoção de características. A sua posição relativa é determinante para a inclusão ou exclusão da árvore final.

Existem dois critérios de paragem-base do algoritmo proposto, por iteração completa sem obtenção de uma árvore com melhor desempenho, ou por número máximo de iterações.

A escolha do critério de paragem está relacionada, naturalmente, com o tempo de indução. A limitação de um número de iterações assegura, à partida, um tempo de indução menor. Todavia, existe outro elemento determinante na escolha, o tratamento das características que foram removidas do conjunto de características S por não terem contribuído para a obtenção da melhoria do desempenho. No caso das características serem re-adicionadas no

final das listas de características candidatas, permite que sejam mais tarde seleccionadas, tornando o critério de ciclo completo mais atractivo.

Classificador Naive Bayes

O tutor possui, ainda, outro tipo de algoritmo de indução de classificadores fortemente relacionado com a noção de incerteza, o classificador Naive Bayes. Na sua implementação utiliza-se a estimativa do cálculo das probabilidades de um evento, tendo por base que

$$\hat{p} = \frac{conta(ocorrências)}{conta(oportunidades)}, \tag{72}$$

ou seja, a estimativa é efectuada pela verificação do número de ocorrências sobre o número total de oportunidades. A validade desta aproximação está directamente relacionada com a relevância estatística dos dados e a utilização de pequenos números compromete seriamente a validade das estimativas efectuadas, uma vez que pequenas alterações afectam significativamente os valores obtidos. Neste sentido, é necessário ter presente que o número de amostras deve ser significativo, com vista a não introduzir desvios relevantes no classificador estimado. Para o cálculo da probabilidade condicionada de A dado B recorreu-se a:

$$\hat{p}(A \,/\, B) = \frac{conta(A,B)}{conta(B)} \tag{73}$$

em que $conta(A,B)$ corresponde à contagem das ocorrências de A e B em simultâneo, e $conta(B)$ as ocorrências de B.

A equação (73), para além de apresentar o mesmo problema da relevância estatística, acresce um novo problema, específico da sua aplicação ao classificador Naive Bayes. Na eventualidade de uma característica apresentar ocorrência zero, o resultado do classificador será inevitavelmente zero, devido ao produto das probabilidades. Na prática, a existência de um atributo com ocorrência zero, elimina de imediato a hipótese de selecção da classe em estudo, o que é uma falha significativa. Com vista a contornar esta situação, sem se recorrer à atribuição de valores de arbitrários às probabilidades zero, recorreu-se à utilização do operador de Laplace. Assim, a probabilidade de A dada a classe B passa a ser:

$$p(A \,/\, B) = \frac{conta(A,B) + 1}{conta(B) + |\,S\,|}. \tag{74}$$

A aplicação da equação (74) a um conjunto de treino, resulta em:

$$p(a_i \,/\, c_i) = \frac{conta(a_i, c_i) + 1}{conta(c_i) + |\,S\,|}, \tag{75}$$

onde $conta(a_j, c_i)$ corresponde ao número de vezes que o atributo está presente no conjunto de observações classificadas como c_i, $|S|$ a cardinalidade do conjunto de características, e $conta(c_i)$ o total de características presentes nas observações classificadas como c_i.

Por aplicação da equação (72) e (74) à equação (45) resulta em

$$\underset{c_i \in C}{arg\ max} \quad \frac{conta(c_i)}{N} \prod_{j=1}^{k} \frac{conta(a_j, c_i) + 1}{conta(c_i) + |S|} \tag{76}$$

em que N corresponde ao total de observações.

O processo de aprendizagem deste algoritmo consiste no cálculo das probabilidades condicionadas e das probabilidades *a priori,* de cada classe, podendo ser descrito da seguinte forma:

N toma o valor do número de observações no conjunto de Desenho
Para cada *classe* $c_i \in C$

$conta(c_i)$ *toma o valor do número de documentos do tipo* c_i *, utilizado para a probabilidade, a priori, e número máximo possível de oportunidades de ocorrência de um atributo*

$$p(c_i) = \frac{conta(c_i)}{N}$$

Para cada *atributo* $a_i \in A$

$conta(j, c_i)$ *toma o valor das ocorrências de* a_i *nos documentos tipo* c_i

$$p(a_j / C_i) = \frac{conta(j, c_i) + 1}{conta(c_i) + |S|}$$

Guarda na Matriz de probabilidade
Retorna: *Matriz de probabilidades*

O processo de classificação de novas observações é descrito da seguinte forma:

Retorna: $\hat{c} = \underset{Ci \in C}{arg\ max} \quad p(c_i) \prod_{j=a_1}^{a_k} p(a_j / c_i)$

4.2.2.5 Sistemas de Apoio à Decisão

A etapa seguinte consiste na criação de SAD, recorrendo à combinação dos múltiplos modelos induzidos, através da elaboração de Processo de Tomada de Decisão (PTD) que permita agregar as suas estimativas individuais. A utilização de um PTD é especialmente interessante, devido à natureza distinta dos métodos de indução apresentados nas secções anteriores, e à inexistência de superioridade genérica de um algoritmo de indução de classificadores. A Figura 45 apresenta a interface construída, sendo visível a possibilidade de selecção dos classificadores, do processo de tomada de decisão e do universo de Navegadores a quem são comunicados os novos SAD.

Os modos disponíveis para elaboração do PTD são: *i)* a selecção de classificador; *ii)* a maioria; e *iii)* método *Flexible and Adaptive joining of Estimators* (Fajé).

A elaboração do PTD baseado na selecção de classificador consiste em abdicar de um processo de combinação de diversos classificadores. Este modo assume, para a estimativa do SAD, os valores determinados pelo classificador seleccionado.

Figura 45 – Interface do Tutor para a definição de SAD para o Navegador

O modo seguinte cria um PTD com o conjunto de classificadores seleccionados pelo utilizador, baseado na regra da maioria.

O último modo disponível, o modo Fajé, implementa um processo de generalização por empilhamento através da criação de um PTD por indução de uma árvore de decisão, tendo por base o comportamento dos classificadores em uso, apelidados de classificadores intermédios. O princípio-base do método consiste no enriquecimento da base de dados de exemplos do conjunto de desenho com os valores atribuídos pelos classificadores intermédios, permitindo, assim, que para além do vector de representação do documento e da classificação real, se passe a ter disponível a classificação atribuída por cada classificador intermédio. Em seguida, as classificações atribuídas por cada classificador intermédio passam a ser consideradas como características do documento e é utilizado um método de indução para inferir o PTD.

Por outras palavras, sendo

$$E = \{e_1,\ldots,e_{|E|}\}, \tag{77}$$

o conjunto de classificadores intermédios, e

$$P = \{P_1,\ldots P_{|E|}\}, \tag{78}$$

o conjunto de novos atributos, em que p_i toma o valor estimado por e_i, o conjunto de desenho passa a ser discriminado por

$$F = S \cup P \,, \tag{79}$$

e pretende-se , inferir um PTD, tal que,

$$PTD(I) : S_1 \times S_2 \ldots \times S_k \times P_1 \times \ldots \times P_{|E|} \rightarrow C \,, \tag{80}$$

seja uma relação em que I é o conjunto de pares ordenados de classificadores intermédios e seu desempenho, S_i a i-ésima característica original e P_i a estimativa atribuída pelo i-ésimo classificador intermédio que melhor descrimine C.

Desta forma PTD entrará em linha de conta não só com os valores do vector de representação como, igualmente, com as classificações estimadas pelo conjunto de classificadores em uso.

O método implementado é, assim, baseado num processo de indução a duas fases que consistem no enriquecimento dos exemplos com novas características de classificação e na indução do PTD.

A Figura 46 apresenta a transformação de um documento na etapa de enriquecimento, através da utilização dos classificadores intermédios, sendo o vector inicial enriquecido com as suas estimativas. Naturalmente, todo o conjunto de desenho tem que ser enriquecido, pelo que o algoritmo apresenta uma complexidade linear dependente da dimensão do conjunto.

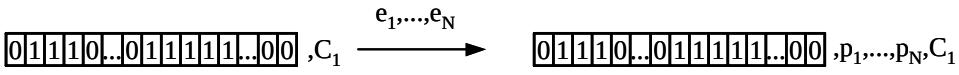


Figura 46 – Exemplo de transformação de um documento na etapa de enriquecimento do conjunto de desenho no método Fajé

Após o enriquecimento de todo o conjunto inicia-se a fase da indução do PTD, entrando em linha de conta com os novos atributos. Na indução do PTD é necessário ter em atenção que existem dois conjuntos de valores admissíveis, (os valores das características originais e os valores estimados), o que obriga a uma adequação dos algoritmos. A indução consiste na selecção de características e indução do classificador.

Tendo em conta que os classificadores intermédios podem utilizar vectores de representação distintos, o processo de selecção de características é realizado sobre o conjunto resultante da reunião de todas as características presentes nos diversos vectores, ao qual se reúnem as novas características. Após a selecção de características, passa-se à fase de indução do classificador.

No caso do classificar induzido não utilizar algumas características acrescentadas durante o processo de enriquecimento, os classificadores respectivos podem ser eliminados tendo em conta que a sua estimativa não será utilizada no processo de decisão. A potencialidade de remoção de classificadores que não acrescentam informação relevante ao processo de decisão é elevada, tendo em conta a diminuição de tempo necessária para a obtenção da estimativa final.

O PTD que se obtém funciona em tempo real, a jusante dos classificadores em uso, realizando a estimativa de classificação tendo em conta não só o valor dos atributos de representação do documento mas, igualmente, tendo em conta os valores estimados por cada um dos classificadores intermédios.

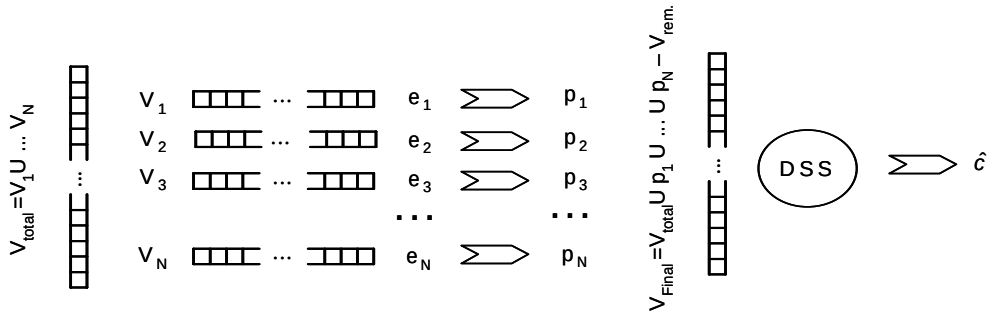


Figura 47 – O processo de tomada de decisão baseado em DSS construídos com o recurso ao método Fajé.

A Figura 47 ilustra o processo de tomada de decisão baseado no DSS criado com o recurso ao método Fajé. O documento é representado através do vector V_{total} , que contém a reunião das características necessárias para os classificadores intermédios. Cada classificador intermédio, baseado no vector que necessita realiza a sua estimativa, que é disponibilizada ao PTD que toma a decisão final baseada nas características seleccionadas, durante o seu processo de indução V_{Final} .

4.2.3 Definição das regras do SAD do Explorador

A definição do SAD do Explorador é realizada pela definição de regras de inferência *forward* utilizadas na identificação de informação relevante. As regras procuram capturar os processos comuns de apresentação de informação, e foram inferidas manualmente por análise crítica dos documentos existentes no *corpus*.

O processo de identificação de informação consiste na capacidade de **extracção de conceitos** e de criação de um **conjunto de palavras-chave**, (para cada conceito identificado), que é utilizado para a sua posterior classificação. Este processo não aplica técnicas de aprendizagem automática. As regras, que foram previamente definidas, são

operacionalizadas para cada caso concreto pelo utilizador, pela definição de um conjunto de parâmetros através do Tutor.

4.2.3.1 As regras de extracção de conceitos

As **regras de extracção de conceitos** são regras de inferência, previamente instaladas no sistema, que descrevem o processo de reconhecimento de conceitos (assim como dos seus atributos), cabendo ao utilizador, exclusivamente, a sua operacionalização por descrição da informação a utilizar. O primeiro passo consiste na definição dos processos de identificação da presença de conceitos, i. e., definição do momento de activação da regra, seguido, naturalmente, pela definição de regras de extracção da informação dos conceitos, i. e., o processo de identificação de qual a informação que faz parte do conceito.

O Tutor possui dois tipos de regras: *i)* **extracção de conceitos de tabelas** e de *ii)* **extracção de conceitos em folhas de texto**.

O primeiro tipo de regra permite a **extracção de conceitos de tabelas**. Assume-se assim, que a cada linha da tabela corresponde um e só um conceito, que é integralmente descrito através das suas diversas colunas, i. e., que cada coluna da tabela contém uma característica do conceito. Utilizando, uma vez mais, o exemplo do reconhecimento de produtos, a Figura 48, permite identificar um modo comum de apresentação de informação para venda e, neste contexto, a cada linha corresponde um produto e a cada coluna um atributo do produto.

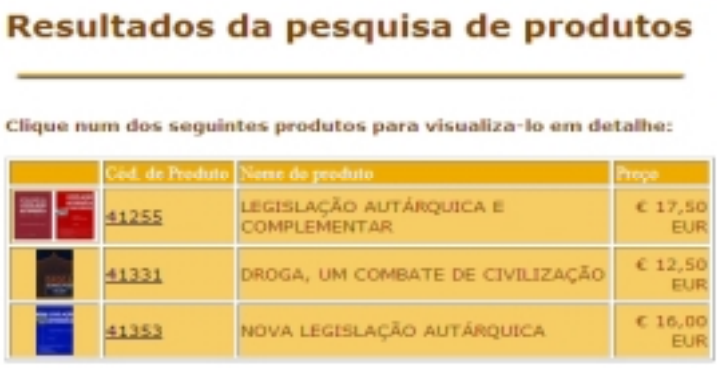


Figura 48 – Exemplo de documento HTML que apresenta produtos em formato tabela.

Esta **regra é activada** sempre que é identificada **a presença de uma tabela no documento** em análise.

A primeira acção, executada pela regra, visa **associar significado semântico** a cada coluna da tabela, de modo a identificar os diferentes atributos de conceito.

Para realizar o reconhecimento semântico dos atributos utilizou-se um processo de análise do conteúdo do cabeçalho das tabelas. Extrai-se o texto de cada elemento do cabeçalho e

faz-se a sua comparação com os conjuntos de palavras previamente definidos pelo utilizador, para cada atributo (palavras de cabeçalho). **Compete ao utilizador, definir o melhor conjunto de palavras de cabeçalho** (na prática as palavras mais comuns utilizadas nos cabeçalhos para descrever as colunas) que são posteriormente utilizadas para o reconhecimento dos atributos. Desta forma, a regra extrai o texto de cada coluna do cabeçalho e compara-o com a lista de palavras de cabeçalho, fazendo a associação, em caso de sucesso, da coluna ao respectivo tipo atributo.

O passo seguinte, ocorre, exclusivamente, se foi possível associar significado semântico às colunas, e consiste na extracção dos conceitos realizada por análise das sucessivas das linhas de tabela, assumindo um conceito por linha. No caso do reconhecimento de produtos, a primeira acção procura validar o cabeçalho da tabela, relacionando cada coluna da tabela com um atributo de produto, e a extracção é efectuada por iteração nas linhas, assumindo que, em cada linha, existe um produto que é descrito nas sucessivas colunas.

A título de exemplo, a segunda linha da tabela apresentada na Figura 48, permitiria extrair um produto com código de referência 41331, descrição «Droga, um combate de civilização», preço 12,50€ e URL (apresentado na coluna da esquerda).

Foram instanciadas 4 regras, seguindo o princípio descrito, assumindo que a posição do cabeçalho pode variar, encontrando-se no topo da tabela, (tal como foi apresentado), no fundo da tabela ou, ainda, nos lados da tabela, (o que obriga neste caso a assumir que os produtos são apresentados nas colunas e os seus atributos nas linhas).

O segundo tipo de regras, permite a **extracção de conceitos em folhas de texto** tirando partido da assumpção de que diferentes atributos na descrição de um conceito estão envolvidos em diferentes marcas HTML. A análise de um documento em HTML pode ser realizada assumindo a existência de uma árvore de marcas de formatação de texto. Um exemplo de um documento HTML, e da sua análise em árvore é apresentado na Figura 49.

O processo de extracção baseia-se na identificação de conjuntos de folhas de texto terminais que, potencialmente, correspondem a um único conceito. No exemplo da Figura 49 existem 6 folhas terminais que definem três conceitos. Cada folha está separada das outras através das marcas HTML que a envolvem.

Esta **regra é activada** para todos os documentos analisados pelo Explorador.

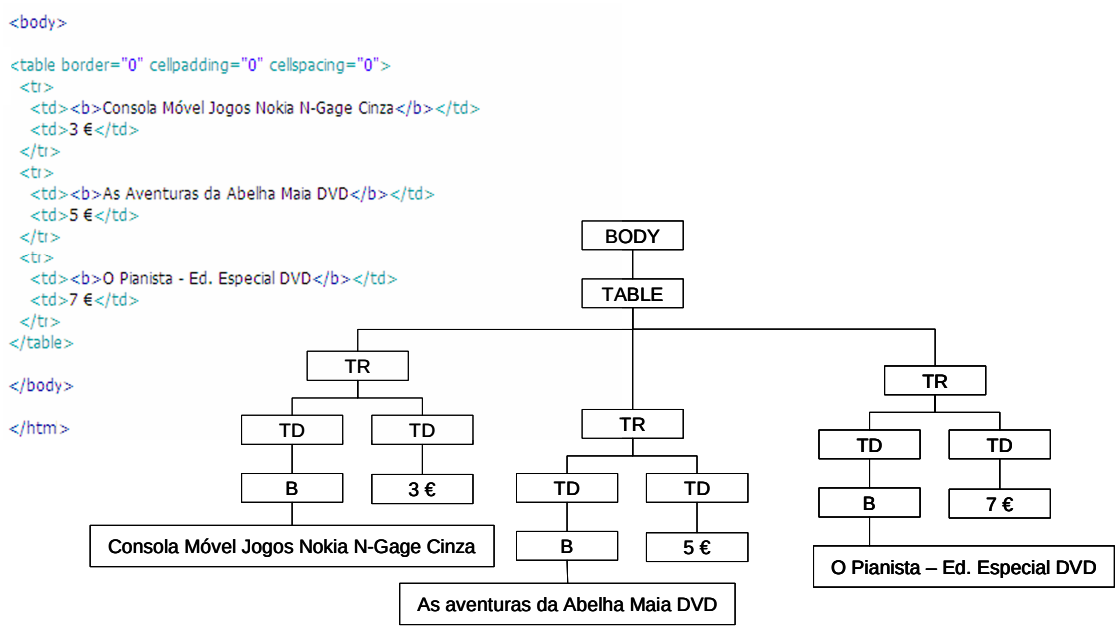


Figura 49 – Exemplo de um documento HTML e da sua representação numa estrutura de árvore de marcas HTML

A primeira acção visa identificar, recursivamente, todas as folhas de texto dependentes de uma marca de nível superior. O objectivo deste processo reside em isolar os nós da árvore que contém abaixo de si informação de um único conceito. O resultado deste passo é um conjunto de vectores de folhas de texto que agrupam todos os textos hierarquicamente abaixo dos nós processados. No exemplo, da Figura 49, o resultado seria:

[A] · Consola Móvel Jogos Nokia N-Gage Cinza, 3 €, As Aventuras da Abelha Maia DVD, 5 €, O Pianista – Ed. Especial DVD, 7 €

[B] · Consola Móvel Jogos Nokia N-Gage Cinza, 3 €

[C] · As Aventuras da Abelha Maia DVD, 5 €

[D] · O Pianista – Ed. Especial DVD, 7 €

[E] · Consola Móvel Jogos Nokia N-Gage Cinza

[F] · As Aventuras da Abelha Maia DVD

[G] · O Pianista – Ed. Especial DVD

[H] · 3 €

[I] · 5 €

[J] · 7 €

A acção seguinte consiste na **filtragem de vectores** que permite eliminar informação redundante e inconsistente. A realização deste passo obriga à supressão de vectores incompletos e ao reconhecimento de itens irrelevantes, correspondendo, respectivamente, à eliminação de vectores que não possuem conceitos e à eliminação de itens que não correspondem a informação relacionada com atributos do conceito. A operacionalização desta regra, realizada pelo utilizador, passa pela descrição das características específicas dos conceitos em análise, no âmbito do problema concreto.

A acção seguinte, **a eliminação de informação irrelevante**, não necessita da intervenção do utilizador e consiste em suprimir itens que sejam compostos por palavras usadas de forma repetida, (que não contêm informação), tais como, uma vez mais, no exemplo dos produtos, 'preço', 'comprar', 'informação' ou 'detalhe'. O conjunto de palavras a eliminar é construído, para cada página, em tempo de execução e contêm os termos que são repetidos mais do que uma vez por cada vector, (a múltipla ocorrência de uma palavra num vector é apenas contabilizada uma vez, e não são contabilizadas folhas que contenham preços). As folhas de texto que correspondem a itens deste conjunto são igualmente eliminadas dos vectores, processo que deixará intactas as folhas que contenham informação relevante. De seguida, são eliminados vectores incompletos, i. e., são eliminados vectores que contenham apenas uma folha, pois será necessário um mínimo de duas folhas por vector para conter um conceito.

O resultado das duas acções permite reduzir o conjunto original ao seguinte conjunto:

[B] · Consola Móvel Jogos Nokia N-Gage Cinza, 3 €
[C] · As Aventuras da Abelha Maia DVD, 5 €
[D] · O Pianista – Ed. Especial DVD, 7 €

Depois deste passo assume-se que o conjunto conterá somente vectores com conceitos e, dentro desses vectores, encontrar-se-á apenas informação relacionada a atributos.

O último passo, consiste em **atribuir significado semântico ao conteúdo** de cada vector, o que acontece, uma vez mais, pela definição das palavras-chave mais usuais e de características de conceito.

Ao contrário do que acontece na regra de **extracção de conteúdos de tabelas** em que a operacionalização depende, exclusivamente, da definição do conjunto de palavras-chave de cabeçalho, esta regra obriga a inclusão de linhas de código, não tendo sido possível a sua realização por simples configuração paramétrica.

4.2.3.2 As regras para extracção de palavras-chave

Para além de extrair a informação do conceito é necessário compor o conjunto de palavras-chave que permitem ao Catalogador a sua posterior classificação. Em tempo real, cabe ao Explorador seleccionar para cada conceito identificado, o conjunto de palavras-chave que fazem parte da palavra-chave composta, que é utilizada para seleccionar qual ou quais os conceitos correspondentes considerados para catalogação no catálogo.

A palavra-chave composta é criada tendo em consideração: *i)* um subconjunto de atributos recolhidos no processo de identificação de conceitos; *ii)* informação resultante da interpretação do elo de localização da página URL [53].

A identificação dos atributos a considerar é uma tarefa simples e rápida, uma vez que consiste na identificação dos atributos que melhor permitem discriminar os conceitos. No exemplo dos produtos o candidato natural é a descrição de produto. Esta actividade é genérica e aplica-se a todos os conceitos sendo independente dos sítios internet em análise.

A interpretação dos elos consiste na capacidade de extrair a informação armazenada nos endereços das páginas que estão a ser processadas. A maior parte das páginas geradas dinamicamente possuem endereços muito expressivos que contêm dados valiosos sobre a informação apresentada. Este tipo de regras é dependente dos sítios em análise sendo dificilmente aplicável a casos desconhecidos, contudo tem a vantagem de conduzir a desempenhos muito elevados. Este tipo de regra é valido, até que os meta-dados dos sítios sejam alterados, o que não é uma operação muito comum tendo em conta o esforço envolvido na tarefa. As classificações perduram muito para além das alterações dos dados específicos dos produtos. Esta técnica é tanto mais importante quanto mais vasta é a existência de sítios Internet construídos de forma dinâmica.

4.2.4 Personalização da ontologia para o Catalogador

A última tarefa consiste na personalização da ontologia de domínio. Para além da hierarquia de conceitos e suas relações, é necessário adicionar a cada conceito um conjunto de palavras-chave (palavra-chave composta) que o identificam.

O Tutor disponibiliza o acesso à interface de classificação manual, apresentada na Figura 29, que permite ao utilizador inserir as palavras-chave que são, posteriormente, utilizadas para o reconhecimento dos conceitos.

4.3 Detalhes de implementação dos agentes

Na implementação do sistema foram adoptadas as seguintes soluções partilhadas por todos os componentes: *i)* a hierarquia de classes, *ii)* a operacionalização dos comportamentos; *iii)* representação dos documentos; *iv)* política de gestão dos recursos físicos disponíveis.

i) A hierarquia de classes

Os agentes foram integrados, como é ilustrado na Figura 50, na hierarquia de classes de agente definida no JADE e na dependência directa da classe «GuiAgente», que acrescenta capacidades gráficas à classe genérica «Agent».

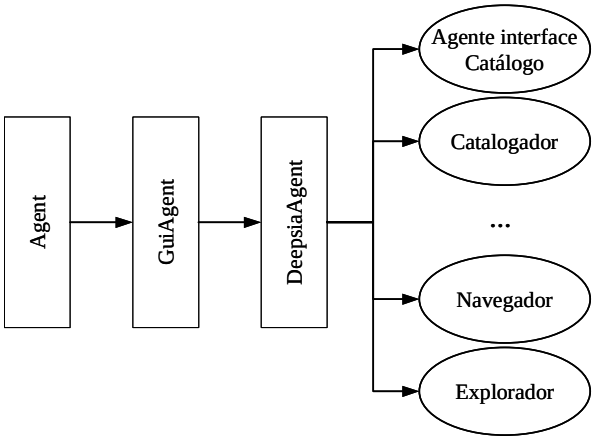


Figura 50 – Ilustração da hierarquia de classes adoptadas para a implementação dos agentes

ii) Operacionalização dos comportamentos

Os comportamentos foram implementados com o recurso à classe «Behaviou*r*» que permite o lançamento de processos paralelos com vista à optimização dos recursos e execução de tarefas concorrentes. Mais do que o paralelismo, (que não é real uma vez que as plataformas utilizadas são monoprocessador), este mecanismo permite uma abstracção conceptual e uma programação mais simples e elegante. A título de exemplo, esta aproximação permite ao Explorador a exibição, em simultâneo, dos seguintes comportamentos: recepção de mensagens, carregamento de documentos dos concentradores de páginas, extracção de conceitos, recepção de mensagens do catalogador e envio de mensagens. A possibilidade de programação independente destes comportamentos, através de «Behaviour», permite: *i)* isolar potenciais problemas; *ii)* auxiliar na obtenção de um código-fonte legível; e *iii)* explorar as potencialidades do paralelismo.

iii) Representação dos documentos

Para a representação do documento em formato original, optou-se pela utilização da classe Page que, apesar de suportar qualquer tipo MIME, foi utilizado, essencialmente, para HTML. Este objecto faz o processamento dos documentos, armazenando-os numa estrutura em árvore, que permite o acesso eficiente às marcas, às palavras ou aos elos.

iv) Gestão dos recursos físicos disponíveis

Os recursos físicos disponíveis não permitiam a disponibilização de um conjunto ilimitado de agentes e catálogos, pelo que foi definido um número máximo de utilizadores em simultâneo, aos quais é assegurado um ambiente de trabalho independente. Esta abordagem permite, igualmente, uma mais fácil monitoração dinâmica do sistema e um controlo mais apertado do protótipo através da Web, assegurando, ao mesmo tempo, uma qualidade de serviço mínima.

Neste sentido, a cada utilizador é reservada uma base de dados lógica, e um ambiente de trabalho para pesquisas autónomas, (a comunidade de agentes Exploradores e

Navegadores é reduzida ao mínimo), i. e., um único agente (Navegador e Explorador). Os restantes recursos são partilhados. O sistema liberta automaticamente os recursos reservados após um período pré-definido de não utilização (por omissão, 24 horas), permitindo a sua recuperação automática.

5 Estudo de Caso

Este capítulo apresenta o estudo de caso utilizado que serviu de base para validar o enquadramento global proposto nesta dissertação.

A primeira secção discute a evolução dos modelos de negócio e consequente desadequação das tecnologias de informação tradicionais e apresenta o conceito de «e-procurement». A segunda secção apresenta o projecto DEEPSIA e as restantes os resultados obtidos com a utilização do enquadramento global proposto.

5.1 Introdução

Os produtos e serviços são o desfecho da composição dos resultados parciais obtidos nas diversas etapas da sua construção, onde interagem diversos actores, entre eles os fornecedores de componentes, os fabricantes, e os clientes, suportados por uma infra-estrutura que viabiliza as necessárias relações. Em paralelo a esta cadeia, existem diversas entidades que disponibilizam serviços de manutenção, publicidade, gestão, etc., que contribuem de forma indirecta para a sua concretização. A maioria das organizações possui um conjunto alargado de relações, dependendo da natureza do produto ou do serviço que produz. No caso dos fabricantes PME o número de relações oscila entre as cem e as mil, dependendo da sua dimensão e internacionalização.

5.1.1 Os modelos de negócio

Neste contexto existem diversos modelos de negócio que pautam as interacções. As **cadeias de fornecimento** são a estrutura tradicional de organização das unidades fabris; caracterizadas por uma linearidade, comprador/vendedor, muitas vezes artificial, i. e., uma organização adquire dos seus fornecedores e vende aos seus clientes [225]. Este modelo afasta diversos fornecedores (assim como os fornecedores dos clientes finais) conduzindo a que eventos determinantes (ocorridos nos clientes ou no início da cadeia) levem semanas a ter reflexos, conduzindo a perdas importantes de competitividade.

Michael Porter [226], em 1980, descreve o enquadramento do **modelo de cadeia de valor acrescentado,** como uma artéria da economia em que circulam produtos e serviços. Neste modelo, as empresas estão localizadas numa cadeia de valor acrescentado em que compram produtos e serviços, acrescentam valor e vendem aos seus clientes. A análise não está assim limitada à natureza das ligações como acontece nas cadeias de fornecimento; neste modelo é acrescentada a análise da alteração dos recursos económicos ao longo da cadeia, pela sua crescente valorização.

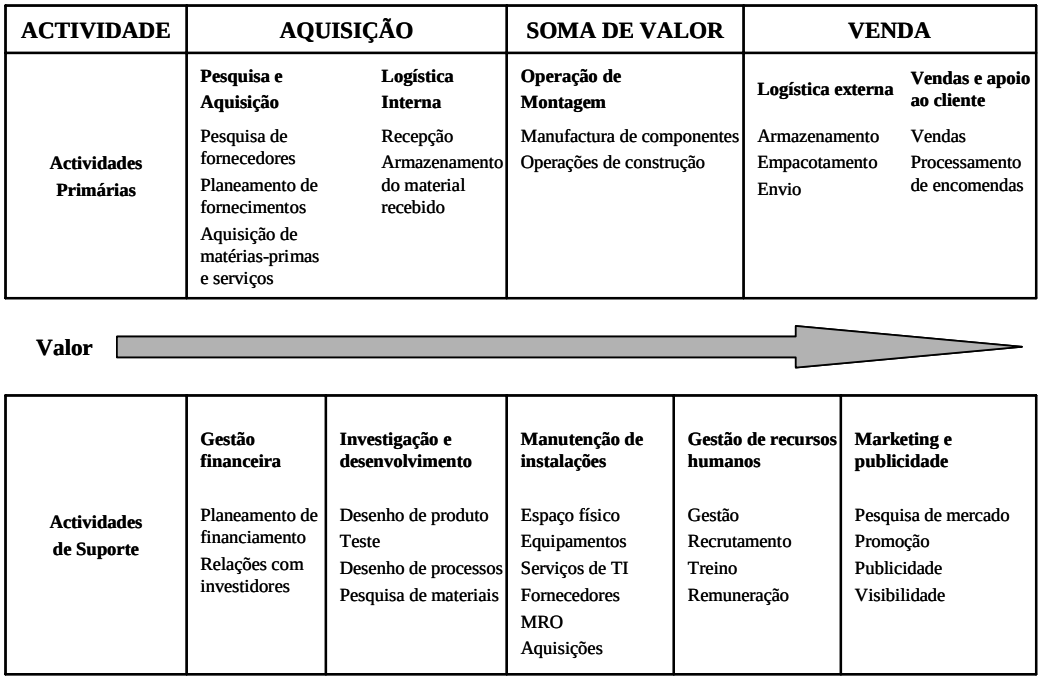


Figura 51 – Adaptação da figura original do enquadramento da Cadeia de valores de Michael Porter

As actividades primárias têm por objectivo a produção do produto ou serviço e a sua venda, incluindo as pesquisas de mercado e aquisições de matérias, a logística interna, as operações do processo de produção, a logística externa, a venda e o suporte ao cliente. As actividades de suporte visam o apoio à sua construção, incluindo, por sua vez, os recursos financeiros, a investigação e desenvolvimento, a gestão de instalações, os recursos humanos, o *marketing* e a publicidade.

Com a divulgação da Web os modelos em cadeia desenvolveram-se, estendendo a sua abrangência focada numa só empresa, para **modelos de cadeia ponto-a-ponto estendida,** que inclui todas as entidades que contribuem com recursos e serviços para a elaboração de um produto final. Consequentemente, as cadeias deixaram de ser lineares transformando-se em redes e todas as relações entre participantes são possíveis. Nesta configuração, as relações são mais próximas, i. e., existem, potencialmente, menos intermediários, o que permite diminuir a distância dos fornecedores ao cliente final combatendo a latência imposta pela propagação de informação ao longo das cadeias. Acresce a este aumento de eficácia estrutural, o facto das ligações baseadas na Internet permitirem relações extremamente

dinâmicas e flexíveis, permitindo gestão de informação colaborativa em tempo real entre os parceiros envolvidos. A utilização da Internet permitiu, inclusive, que uma empresa deixasse de realizar todas as actividades relacionadas com a sua actividade principal, permitindo a delegação de tarefa. A Dell Computer Corporation é um conhecido exemplo de construtor de computadores que deixou de construir componentes, utilizando a Internet para controlar a aquisição de componentes e serviços de logística. A actividade principal da empresa passou, assim, a ser a gestão, através da Internet, de ordens de construção de componentes e equipamentos desencadeadas directamente pelos seus clientes.

5.1.2 Tecnologias desadequadas aos novos modelos

Nas cadeias de fornecimento, as relações eram geridas por EDI ou protocolos proprietários, e caracterizavam-se por serem inflexíveis, estáticas e extremamente duradouras. Todo o modelo está assente em mecanismos de confiança, que são amadurecidos ao longo dos tempos, após complexos processos de conhecimento, em que um dos parceiros ocupa uma relação predominante, a exemplo do que acontece na indústria automóvel em que os construtores são determinantes e ditam as regras. Todavia, esta filosofia não é adequada a modelos em rede que, assentes na Web, tiram partido da democratização e da facilidade de comunicação. O estabelecimento de ligações passa a ser muito mais dinâmico e flexível, potenciando a criação de novas configurações e estimulando, inclusive, a criação de consórcios temporários para o fornecimento de matérias ou serviços. A competição deixa, assim, de ser entre empresas, passando a estar fortemente ligada a redes de valor acrescentado que permitem satisfazer os pedidos dos clientes. Um pedido de um cliente desencadeia um ajustamento na rede e determina a criação dinâmica de consórcios que permitem satisfazer o pedido em causa. Naturalmente que uma empresa pode pertencer a diversos consórcios e no limite, a todos se for fornecedora exclusiva de uma componente essencial. Estes consórcios são altamente voláteis, e só são criados com a efectivação do pedido do cliente, i. e., somente o consórcio vencedor é na realidade criado. Neste modelo de negócio o consórcio não tem uma existência legal, uma empresa agregadora realiza a oferta ao cliente e encarrega-se, caso vença o concurso, de coordenar a boa execução das tarefas. Neste ambiente extremamente dinâmico, é essencial a existência de ferramentas que permitam auxiliar as empresas a criar as soluções mais favoráveis, i. e., identificar quais as ligações mais favoráveis naquele momento particular para o negócio específico. Esta capacidade permite evitar a utilização de um conjunto restrito de fornecedores, o que acontece devido à impossibilidade de comparar e consultar um vasto conjunto de possibilidades por natural falta de tempo. Tendo em conta que estas ferramentas capturam a fiabilidade dos fornecedores, baseada num histórico e na credibilidade atribuída pelo mercado, permitem com grande facilidade testar diversas configurações capacitando uma ampla e eficaz pesquisa de mercado.

5.1.3 O «e-procurement»

O «e-procurement» é o processo de negócio de aquisição de matérias, equipamentos e serviços com o recurso ao suporte de meios digitais. O processo de negócio inclui a identificação das necessidades, a selecção das potenciais soluções, dos possíveis fornecedores, do processo de selecção, aquisição e, finalmente, da avaliação dos resultados obtidos. Os benefícios por comparação com o «procurement» tradicional, ultrapassam a mera redução de custos, uma vez que força a adopção de um novo modelo de negócio ágil e flexível alterando os processos de negócio tradicionais com ganhos de eficiência. A maior parte das soluções aplicacionais disponíveis no mercado estão focadas na área MRO (Manutenção, Reparação, Operação).

As despesas totais de aquisição de um bem ultrapassam, assim, os seus custos directos, incluindo todos os custos associados ao processo de aquisição, i. e., o tempo de pessoal gasto na identificação, comparação, negociação, fornecimento do bem em causa, assim como as autorizações, as interacções interdepartamentais (e. g. contabilidade e tesouraria) controlo de qualidade, inventários, entre outras [227]. Genericamente, a regras dos 20/80 aplica-se uma vez mais, 20 por cento dos processos consomem 80 por cento dos custos totais, ao mesmo tempo que 80 por cento do tempo do pessoal está associado somente a 20 por cento do valor dispendido.

O «e-procurement» de MRO nas PMEs é reconhecidamente uma área-chave, potencial para o desenvolvimento e o aumento da eficiência, em especial, tendo em conta a quase total inexistência de automatização de processos nesta área. Contudo, não tem sido adoptado, devido à complexidade e à resistência social, o que conduz a que a esmagadora maioria das actividades seja ainda realizada através de contactos pessoais, por telefone e fax. São ainda apresentadas como razões para a sua não adopção, a baixa presença de fornecedores na Web, a falta de demonstração clara do processo de retorno de investimento e dos benefícios associados aos processos digitais.

Esta posição conservadora conduziu a uma resistência à adopção das soluções propostas inviabilizando o sucesso comercial de diversas ofertas, entre elas as baseadas em plataformas Ariba e CommerceOne.

5.1.4 Os tipos de presença na Web

A existência de ofertas na Web é determinante para o desenvolvimento do «e-procurement». Sem um número alargado de fornecedores disponíveis não é possível imaginar as leis de mercado a funcionar, sem que se caia em situações de monopólio efectivo. O esforço de implantação pode estar centrado no fornecedor, no comprador ou em intermediário.

No caso de ser o fornecedor a tomar a iniciativa, estamos na presença da solução em que se disponibiliza um catálogo electrónico com informação rica e facilmente consultável pelo cliente através de uma ferramenta de navegação da Web. Todo o esforço de implementação e manutenção está centrado no fornecedor, sendo por este prisma, o cliente claramente beneficiado, uma vez que o processo de «purchasing» fica mais simplificado, rápido e eficaz. O fornecedor pode, desta forma, aceder a um mercado mais vasto, o mercado global, e reduzir os seus custos devido à automação dos processos, melhorando a sua relação com os clientes.

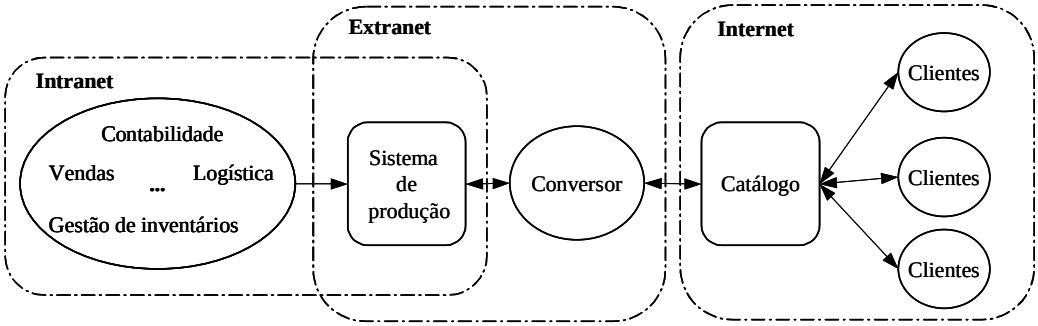


Figura 52 – Modelo de presença Web para compras em que a implementação está sob responsabilidade do fornecedor

No caso da iniciativa estar centrada no cliente, estamos na presença de um modelo em que fornecedores são convidados a descrever os seus produtos em catálogos electrónicos que são geridos pelo cliente. Neste caso, cabe ao cliente o processo de selecção dos fornecedores, sendo o responsável por manter e controlar o sistema de catálogo. O cliente disponibiliza uma infra-estrutura que permite aos seus fornecedores descreverem os produtos que oferecem. Esta solução é extremamente exigente uma vez que obriga, não só à implementação do sistema, como à criação de uma infra-estrutura que permita a sua disponibilização na Web. Apesar do maior esforço estar centrado nos clientes, esta abordagem obriga, igualmente, a um esforço considerável por parte dos fornecedores que têm que descrever os seus produtos no sistema de catálogo disponibilizado, o que representa um esforço suplementar. Na realidade, esta solução é uma extensão das soluções centralizadas em que os fornecedores comunicam com o cliente através de um protocolo previamente definido, e. g., EDI.

A aplicação deste modelo está limitada a grandes clientes, devido aos custos de implementação, à forte necessidade de competências na área das TIC, e à necessidade de atracção de fornecedores. Todavia, os grandes clientes, tipicamente multinacionais, não só têm uma larga experiência na implementação de sistemas semelhantes, como têm a capacidade de atracção suficiente para que os fornecedores encarem a necessidade de descrição dos seus produtos nos catálogos como uma tarefa admissível. Na verdade, por vezes a atracção é tão grande e a capacidade de comparação dos produtos tão poderosa,

que os fornecedores não só aderem, como oferecem condições extremamente vantajosas, o que conduz a distorções no mercado.

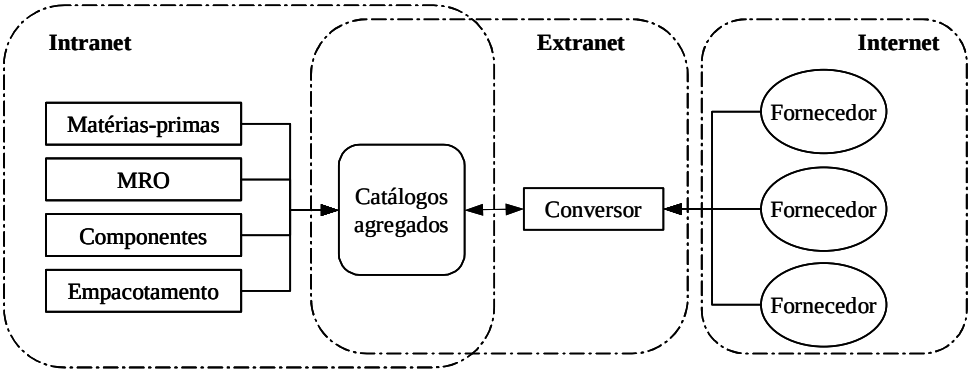


Figura 53 – Modelo de presença Web para compras em que a implementação está sob a responsabilidade do cliente

Em alternativa aos modelos previamente descritos existem **os mercados electrónicos**. Neste caso, um intermediário presta o serviço de pôr em contacto compradores e fornecedores através da Web, facilitando a fase de procura do processo de negócio. Estes mercados podem ser descritos pelas relações que estabelecem entre fornecedores e compradores [227] e tipo de acesso: privado ou público. Neste modelo enquadram-se os «e-hub», «vortals», comunidades de negócio, portais de informação, etc., tendo sido criados com o objectivo de eliminarem ineficiências das cadeias de fornecimento. Michael Evason [228], no final de 2001 apresenta como razões para a atenção prestada pelos media aos mercados electrónicos, essencialmente os elevados investimentos realizados, e por serem apresentados como a base da próxima geração de negócio. Desde então, diversos mercados electrónicos amplamente anunciados falharam, tendo sido, inclusive, removidos o que contribuiu para o actual descrédito neste processo de negócio. A verdadeira falha foi a falta de discernimento que não permitiu ver o óbvio, já que os mercados electrónicos não são mais do que um novo canal entre fornecedores e compradores e sem a oferta de vantagens reais as empresas não aderem à iniciativa. Apesar da facilidade e dos baixos custos para adesão aos serviços, a longo prazo, com o objectivo de maximizar os proveitos, existe a necessidade de fazer um investimento em TIC, i. e., integração com os sistemas de produção, com os catálogos proprietários. Os riscos associados incluem a dependência de um canal gerido por terceiros, perda de retorno no pagamento de taxas e de tecnologia não produtiva, (no caso do mercado não sobreviver), e a necessidade de investir em tecnologias diversas para satisfazer as necessidades de adesão ao mercado.

A atracção para os compradores não pode estar limitada à comparação dos preços dos produtos, uma vez que o factor mais importante não é o preço mas sim a relação qualidade-de-serviço/preço. Esta necessidade obriga os mercados a disponibilizarem ao comprador diversos indicadores que permitem diferenciar os fornecedores, o que não é uma

tarefa fácil, uma vez que este requisito dos compradores enfrenta naturais obstáculos por parte dos fornecedores.

Um estudo sobre os mercados electrónicos europeus, realizado no final de 2001, revelou que existiam 150 mercados de grande dimensão e 1250 de pequena dimensão [228]. Contudo, 40 dos mercados de grande dimensão e 300 de pequena, eram de capitais públicos e dos 134 operacionais em 2000, somente, 64 por cento continuavam activos (32 por cento tinham encerrado e os restantes estavam suspensos temporariamente).

Genericamente, são indicadas como razões mais comuns para o insucesso dos mercados electrónicos:

- a falta de confiança nos sistemas disponibilizados que, por pressão do mercado, foram colocados em produção com graves falhas estruturais, o que conduziu ao seu descrédito;
- a falta de preparação das empresas para participarem numa plataforma aberta para troca de informação, tendo em conta que o domínio das TIC está longe de ser uma característica comum na maioria das empresas, em especial nas PMEs;
- a resistência ao pagamento de taxas;
- a falta de serviços de valor acrescentado devido ao elevado custo associado ao seu desenvolvimento, que consumiram os recursos financeiros dos investidores sem que se materializassem de forma determinante. Os investimentos são esmagadores pois envolvem as diversas fases de suporte ao processo de compra e têm que estar implementados de forma escalável, fiável e robusta, o que não se adequava aos tempos de desenvolvimento impostos;
- a falta de integração com os sistemas dos utilizadores e. g., sistemas financeiros, de gestão de mercadorias, de logística e de compras, que permitiriam ao utilizador, comprador e fornecedor, tirar completo partido dos sistemas de mercado electrónico.

As razões indicadas afastaram, ao mesmo tempo, compradores e fornecedores, não permitindo aos investidores o esperado retorno de investimento. Sem uma massa crítica de compradores que iniciem o processo de compras electrónicas não existe atractivo comercial para um fornecedor aderir e criar os seus catálogos electrónicos. Naturalmente que a falta de fornecedores não encoraja a adesão de novos compradores, conduzindo ao abandono dos que aderiram na esperança de assistirem à evolução do mercado, criando um ciclo de participação reduzida.

A maior parte dos fornecedores não está ainda preparada para a construção de catálogos electrónicos e a pressão sentida, por parte dos compradores, ainda não é suficiente para desencadear o início do processo, ao que acresce a sensação dos fornecedores de que os

benefícios essenciais são colhidos pelos compradores, o que permite justificar o adiamento da criação dos catálogos electrónicos.

5.2 O projecto DEEPSIA

O projecto Deepsia IST-1999-20483, foi financiado pela comissão Europeia enquadrado no Quinto Programa Quadro de Investigação sobre Tecnologias para Sociedade de Informação, na II acção-chave, relacionada com novos métodos de trabalho e comércio electrónico. O principal objectivo da II acção-chave era o aumento da eficiência no trabalho, consequentemente a competitividade e, ao mesmo tempo, melhorar a qualidade de vida no ambiente de trabalho.

O consórcio encarregue de realizar o projecto Deepsia era composto pelas entidades apresentadas na Tabela 6.

Empresa	Actividade principal	País
Comarch	Empresa de TIC Polaca	Polónia
Indra	Empresa de TIC, e fornecedora de soluções Web	Espanha
USP	Universidade de São Paulo	Brasil
Centre for Electronic Commerce, Sunderland University	Centro de Investigação orientado para Negócios	Reino Unido
Uninova	Instituto de Investigação de Novas Tecnologias	Portugal
ULB	Universidade Livre de Bruxelas	Bélgica
Zeus Consulting	Empresa de consultadoria em TICs	Grécia

Tabela 6 – Lista dos parceiros de consórcio do projecto DEEPSIA

O nome Deepsia é o acrónimo de «Dynamic on-linE Purchasing System based on Intelligent Agents». O projecto Deepsia visava a criação de uma ferramenta *on-line*, desenvolvida para PME, escalável e de fácil utilização.

A coordenação técnica do projecto foi entregue ao UNINOVA, **tendo sido utilizado enquadramento global proposto nesta dissertação como ferramenta-base para a obtenção dos resultados propostos.**

Os modelos de negócio baseados em cadeias de fornecimento sequenciais têm sido postos em causa e, progressivamente, substituídos por modelos de cadeias de valor acrescentado. Todavia, a consolidação da Web permite prever novos modelos baseados em redes de valor acrescentado descentralizados alargando as possibilidades de negócio. Neste novo modelo, as entidades envolvidas na criação de um produto, ou serviço final, contribuem com matérias-primas ou serviços estando ligadas através da Internet. O princípio-base deste modelo assenta na gestão eficaz das relações de cada entidade, que pode, a cada momento, seleccionar um novo fornecedor que oferece melhores garantias criando uma

cadeia de valor alternativa. Relações que tradicionalmente são fortes e estáveis podem ser agora postas em causa na procura de uma solução mais ágil e colaborativa. Algumas empresas, em especial no sector das tecnologias de informação, aproximam-se deste modelo com resultados surpreendentes. Os fornecedores de equipamento informático criam, com distribuidores e outros fornecedores de equipamentos, redes de valor acrescentado que permitem a cada momento seleccionar, de forma expedita via Internet, qual a melhor solução para o cliente final.

O processo de «procurement» desempenha um papel relevante na cadeia de fornecimento das PMEs e a optimização deste processo pode aumentar a rentabilidade das empresas em questão. O desafio está em identificar o modelo correcto que ofereça às PMEs uma ferramenta suficientemente fácil para ser utilizada por pessoal que raramente tem um elevado grau de domínio das tecnologias de informação e, ao mesmo tempo, suficientemente eficaz para que possa oferecer um ganho substancial que reduza os custos de operação dos negócios realizados via Internet.

O projecto DEEPSIA propunha a criação de uma ferramenta que auxiliasse a criação de um catálogo personalizado para armazenamento de informação disponível num conjunto de sítios Internet que as PME pretendessem monitorar. Com este objectivo, o sistema manteria o catálogo personalizado devidamente actualizado com as informações contidas num conjunto de sítios Internet que são previamente seleccionados pelo utilizador. A recolha de informação, efectuada fora do tempo nobre de comunicação, seria realizada por um conjunto de Agentes Inteligentes capazes de aprenderem com o tempo as necessidades dos utilizadores.

O catálogo criado armazena e apresenta os produtos identificados no conjunto de sítios Internet fornecidos pelo utilizador, segundo uma taxionomia pré-definida, permitindo que o processo de pesquisa não envolva a consulta dos sítios Internet, sendo substituída pela consulta dos produtos no catálogo pessoal. A informação armazenada sobre os produtos está relacionada com a sua descrição, o preço, descrição do fornecedor e existência em armazém. Desta forma, espera-se auxiliar as PME a encontrar as melhores ofertas para as suas necessidades no processo de «e-procurement», disponibilizando uma ferramenta que permita não só adquirir produtos e serviços com uma melhor relação custo/benefício como a identificação de novos fornecedores. O modelo de negócio que sustenta esta ferramenta posiciona as PMEs como compradoras, ao contrário do modelo tradicional, em que são vistas como fornecedoras de produtos e serviços, através de portais e centros de vendas digitais.

As soluções equivalentes ao DEEPSIA, identificadas no início de 2000, estavam longe de ter atingido a sua maturidade, sendo apresentadas no anexo A.5, que não pretende ser exaustivo, mas sim ilustrativo, do conjunto de opções que estavam disponíveis permitindo,

assim, evidenciar as diferenças da solução proposta. Foram utilizadas diversas fontes de informação, em particular o knowledgeStorm [229] (um directório de soluções aplicacionais).

A esmagadora maioria das soluções estava limitada a pesquisas por palavra-chave, através de interfaces pouco intuitivas, baseadas na navegação directa nos elos [230]. Estas aproximações de muito baixo nível diminuíram as expectativas criadas para o comércio electrónico, tendo em conta que não contribuíram para a obtenção de sistemas que permitissem:

- **a selecção automática da melhor oferta:** A oferta de produtos na Web continua a aumentar em quantidade e diversidade. Paradoxalmente, o aumento de oferta contribui, naturalmente, para dificultar o processo de tomada de decisão, sendo cada vez mais difícil ao utilizador identificar a melhor solução. Neste sentido, é imperativo encontrar processos automáticos de pesquisa de informação com vista a identificar qual o melhor sítio Internet que disponibiliza a solução mais adequada para o utilizador (em termos de custo, tempo de entrega, confiança no fornecedor, etc.);
- **a personalização das pesquisas:** São poucos os sítios que personalizam as suas interfaces tendo em conta as necessidades do utilizador, o que obriga à construção de interfaces genéricas que, raramente, são intuitivas para um público de espectro alargado característico dos mercados electrónicos. É necessário incorporar mecanismos de identificação de perfis e padrões de comportamento que permitam realizar a personalização em tempo real das interfaces ao utilizador em causa;
- **a transparência:** A disponibilização de informação na Web sobre os produtos e seus custos abre novas possibilidades à realização de estudos comparativos. A realização de uma análise de custo detalhada no plano físico, apesar de possível, não é viável, tendo em conta que potenciais economias não compensariam os custos envolvidos na recolha de informação. Todavia, a Web oferece uma plataforma que, bem explorada permite, em tese, a aquisição dos melhores produtos ao mais baixo custo;

A maior parte das soluções eram desadequadas às PMEs uma vez que ofereciam soluções completas de compras electrónicas para empresas de grande dimensão, que possuam parcerias colaborativas com alguns dos seus fornecedores. Estas soluções incluíam a gestão de catálogos multifornecedor, gestão de lojas digitais para B2B e, ou, B2C o que é manifestamente desadequado às necessidades das PMEs.

Não foram identificadas soluções semelhantes à proposta do DEEPSIA, em especial adequadas às PMEs [230]. As soluções que oferecem gestão dinâmica de catálogos para compras são adequadas a empresas de grande dimensão.

Genericamente, as soluções existentes eram demasiado dispendiosas, difíceis de implementar e de utilizar.

O protótipo do projecto foi construído tendo por **base o enquadramento global e a arquitectura de referência proposta nesta dissertação**. A sua implementação foi realizada numa filosofia ASP que permitiu obter as funcionalidades identificadas na Figura 54. A análise da figura permite identificar, na zona inferior, os principais módulos do sistema, (o catálogo e o sistema de multiagentes) e na zona superior, (externo ao sistema), os sítios Internet. As interacções apresentadas descrevem a existência de uma interface privilegiada entre o catálogo e alguns sítios, permitindo pesquisas directas. A figura ilustra, igualmente, a relação com o sistema de multiagentes responsável por actualizar os dados do catálogo.

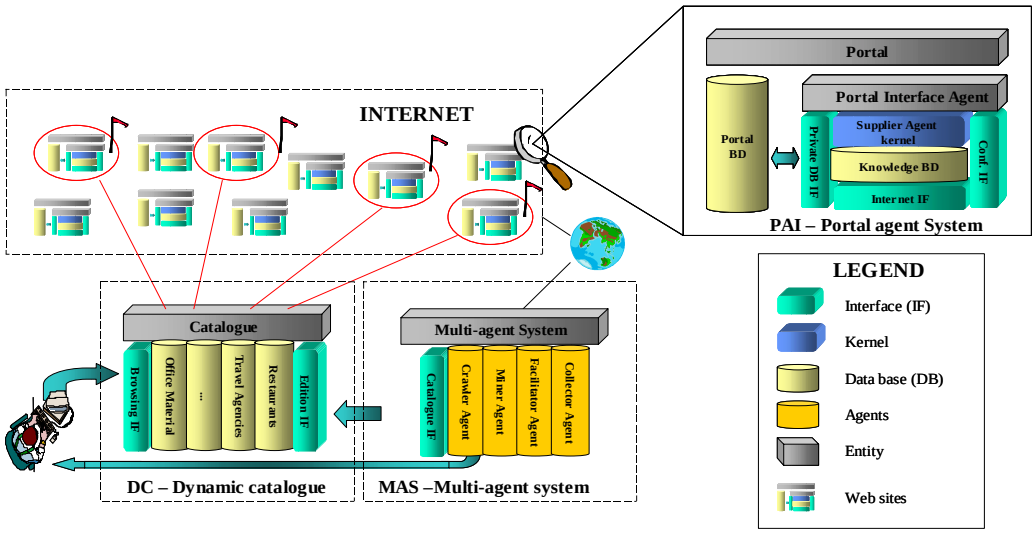


Figura 54 – Apresentação da arquitectura geral do projecto DEEPSIA

As características genéricas da ferramenta proposta são:

- a perspectiva centrada na PME;
- o desenhado para corresponder aos requisitos individuais;
- a existência de uma interface amigável, baseada numa interface com recurso a catálogos personalizados;
- a capacidade de actualização do catálogo de forma semiautomática, através da recolha de informação existente em sítios Internet;
- a actualização dos catálogos baseada, tanto quanto possível, na utilização de processos automáticos recorrendo a um conjunto de agentes que permite automatizar o processamento dos sítios Internet.

As características propostas respondem às necessidades identificadas em Dezembro de 2003, por Danny Sullivan [231] onde apresenta que 41 por centos dos utilizadores que procuraram adquirir um produto através da Web utilizaram uma ferramenta de pesquisa

indicando, como factor principal para a sua utilização, a capacidade de comparação rápida de preços (73 por cento), necessidade de comparação dos produtos (54 por cento) e identificação de uma loja de venda de produtos (45 por cento).

Após a implementação do protótipo, iniciou-se **a aplicação da metodologia específica de suporte à derivação de sistemas particulares** que permitiu operacionalizar o protótipo com vista à recolha e catalogação da informação sobre produtos à venda em sítios na Internet. A execução das tarefas e os resultados obtidos são apresentados em detalhe nas próximas secções.

5.3 Definição da ontologia de representação de domínio

A primeira tarefa da metodologia específica de suporte à derivação de sistemas particulares é a definição da ontologia de representação de domínio, sendo necessário descrever dois domínios de conhecimento: *i)* conhecimento sobre os assuntos considerados relevantes; *ii)* conhecimento sobre os conceitos catalogáveis.

Para a representação dos conceitos catalogáveis, optou-se por identificar uma ontologia já definida, que permitisse descrever produtos e serviços. Todavia, a área de representação de conhecimento está longe de ter atingido a maturidade e não foi possível identificar nenhuma que abordasse e cumprisse integralmente os requisitos. Assim, optou-se por adaptar a taxionomia UNSPSC – Universal Standard Products and Services Classification desenvolvida pela ECCMA[42], com o intuito de normalizar as relações B2B e com o objectivo de poder ser utilizada por empresas de todas as dimensões. A UNSPSC continha, à data, cerca de 12 000 entidades[43] distribuídas por cinquenta e seis sectores de actividade industrial, classificando, em simultâneo, produtos e serviços visando a utilização num ambiente de mercado global.

As razões principais que suportaram a selecção do código UNSPSC foram:

- a cobertura de uma vastíssima gama de produtos e serviços que podem ser transaccionados. Os doze mil conceitos existentes abrangem, desde o simples lápis, até computadores, passando por serviços de limpeza e segurança;
- a cobertura de 56 sectores industriais desde a electrónica, à química, a serviços que vão desde a medicina à educação, a manufactura, ao sector automóvel, etc;

[42] ECCMA – Electronic Commerce Code Management Association, associação internacional que se apresenta como associação Internacional sem fins lucrativos presentes, autónoma, que coordena a gestão e desenvolvimento da taxonomia UNSPSC.

[43] Entidade – Tradução do autor para «Commodity». Os outros níveis são de tradução directa, não merecendo nenhum reparo.

- o facto do código ser aberto, uma norma global, disponível para o grande público, sem restrições ou licenças de utilização. (Uma cópia do código está disponível para análise e impressão a partir do elo http://eccma.org/unspsc/browse/);
- a compatibilidade entre o código UNSPSC e outras normas de classificação (e. g., CPV, NAICS, SIC, HTS).

A taxionomia UNSPSC é um código de dez dígitos de classificação hierárquica, com cinco níveis de profundidade, em que cada nível (ver Tabela 7), é representado por dois dígitos e uma descrição textual [232].

Nível	Descrição
Segmento	Agregação lógica das famílias por funcionalidade.
Família	Um grupo reconhecido de classe de entidades inter-relacionadas.
Classe	Um grupo de entidades que partilham um uso ou função comum.
Entidade	Um grupo de produtos ou serviços substituíveis.
Livre	Campo para utilização livre.

Tabela 7 – Descrição dos níveis hierárquicos do código de classificação UNSPSC

A codificação de cada entidade é, assim, composta por dez dígitos agrupados sequencialmente em pares (níveis) de forma a construir um identificador único, (e. g., o código de cadeira é *56.10.15.04.00*). A Tabela 8 apresenta exemplos de codificação de conceitos na área do mobiliário.

Nível	Exemplo
Segmento	[56.00.00.00.00] Mobiliário e Acessórios.
Família	[56.10.00.00.00] Mobiliário de interiores.
Classe	[56.10.15.00.00] Móveis.
Entidade	[56.10.15.02.00] Sofás.
Entidade	[56.10.15.04.00] Cadeira.

Tabela 8 – Exemplos de codificações UNSPSC de conceitos na área do mobiliário

Os dois últimos dígitos que referenciam a entidade são sempre 00, uma vez que são campos livres dependentes dos fornecedores.

Para a representação do conhecimento sobre os assuntos considerados relevantes, optou-se por utilizar somente duas categorias de classificação de documentos, (documento normal ou documento de venda).

A importação dos dados foi realizada pela descrição de um subconjunto de produtos da taxionomia e dos conceitos (normal e venda), através do Protegé, em formato OWL, tendo sido posteriormente incorporada pelo agente Tutor permitindo contextualizar o protótipo ao tema de produtos.

5.4 Indução de um SAD para os Navegadores

A tarefa seguinte, na metodologia de derivação de um sistema particular, é a indução de um SAD para os Navegadores, permitindo que passem a identificar documentos relevantes para o utilizador (no caso concreto os documentos que apresentam produtos para venda).

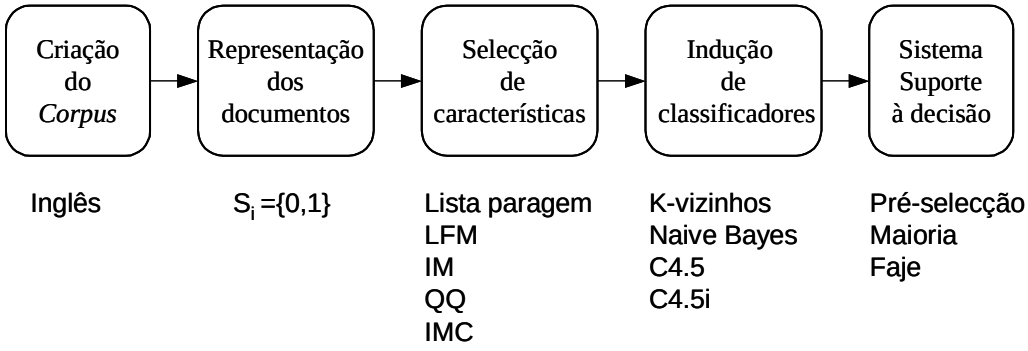


Figura 55 – As tarefas executadas para a indução do SAD para os agentes Navegadores

As etapas e respectivas tarefas executadas são descritas na Figura 55 e apresentadas em detalhe nas próximas secções.

5.4.1 Os resultados apresentados

A utilização de duas categorias para classificação de documentos induz uma forte correlação entre as métricas precisão e rechamada. O aumento da precisão na categoria de vendas reflecte-se no aumento da precisão dos documentos normais e vice-versa. Todavia, o reflexo não é proporcional, dependendo do número de exemplos de cada classe. Na verdade, quanto maior for o desequilíbrio entre classes, menor a proporcionalidade. Esta análise é igualmente válida para a métrica rechamada.

No caso particular, a perda de páginas de venda é muito mais gravosa que a perda de páginas normais, tendo em conta que o objectivo é conseguir obter o maior número de páginas de venda. Porém, é necessário manter presente que o número de páginas, erroneamente classificadas como normais, contribui para a diminuição do desempenho global do sistema, pois obriga, a montante, à sua filtragem. Contudo, pelo exposto e com o objectivo de diminuir a quantidade de dados apresentados, optou-se, na maioria dos casos, pela análise dos algoritmos tendo em consideração as métricas para o caso das vendas.

Os resultados experimentais apresentados nesta secção seguiram o método sugerido na metodologia, de validação cruzada por 10 secções.

5.4.2 A criação do *corpus*

O *corpus* foi criado tendo em conta a metodologia sugerida para assegurar a representatividade do domínio. Neste sentido, definiram-se regras sintácticas, semânticas,

de reconhecimento de estruturas e atributos, de reconhecimento inequívoco de conteúdos e de recolha aleatória, que foram utilizadas no processo de aquisição dos dados.

Para a **definição sintáctica dos dados** definiram-se as seguintes regras:

- **Palavra**: Conjunto de caracteres entre separadores;
- **Separadores**: [‘ ‘,’\t’, ‘\n’, ‘\r’, ‘\’’, ‘,’, ‘.’, ‘@’, ‘?’, ‘\”’, ‘!’, ‘Â«’, ‘Â»’, ‘\\’, ‘:’, ‘;’, ‘#’, ‘=’, ‘{‘, ‘}’, ‘[‘, ‘]’, ‘(‘, ‘)’, ‘/’, ‘%’, ‘|’, ‘+’, ‘-‘, ‘*’].

Para o **reconhecimento semântico dos dados:**

- **Números** [0-9] #number;
- **Moeda** [$,Â£,€] #currency.

No que se refere **ao reconhecimento de estruturas e atributos**, optou-se por armazenar os atributos sobre o texto, («title», «bold», «color», etc.) sendo, em seguida, as palavras reconhecidas marcadas como pertencentes a texto de linguagem natural ou texto de linguagem HTML.

Para **a definição inequívoca dos conteúdos** que fazem parte de uma classe optou-se por identificar os **documentos de venda**, sendo os restantes marcados como **páginas normais**. As regras utilizadas para o reconhecimento de um documento de venda foram:

i) a apresentação de um produto;
ii) a existência de um preço explícito, ou da referência a fornecimento de preço por consulta.

Finalmente, para **assegurar o processo de recolha aleatória de exemplos** realizou-se uma consulta ao sítio *Yahoo!*, à secção *Business_and_Economy, Shopping_and_Services*, tendo sido seleccionadas lojas por **selecção aleatória sem reposição**. Os textos presentes em todas as páginas são em inglês. Foram carregados localmente cem sítios, (ver anexo A.6) permitindo uma cópia fiel, sempre disponível e constante.

De cada sítio foram extraídas cinquenta páginas, uma vez mais **aleatoriamente sem reposição,** que foram posteriormente processados e classificados. O facto do *corpus* não apresentar 5000 páginas resulta da existência de diversos sítios com menos de cinquenta páginas.

No caso de uma aplicação real do DEEPSIA à monitorização de um conjunto limitado de sítios Internet, o universo deixa de ser infinito e aconselha-se, nesse caso, o abandono da pesquisa de exemplos aleatórios e a sua substituição pelos sítios que se pretende monitorar.

5.4.2.1 Caracterização do corpus

O *corpus* ficou composto por um total de 3986 das quais 2846 foram classificadas como de venda e 1140 normais. A informação armazenada inclui, somente, informação textual, (textos em linguagem natural, todos os dados referentes às marcas, linguagens de programação, e meta-informação, etc.) não tendo sido considerados os elementos gráficos.

A Figura 56 apresenta o histograma da dimensão dos documentos.

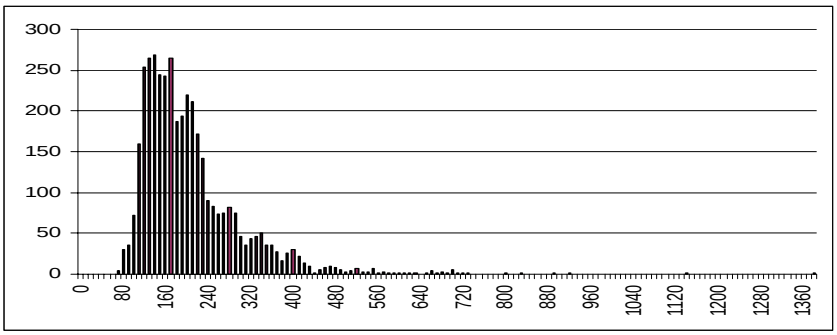


Figura 56 – Histograma da dimensão dos documentos que compõem o *corpus*. O eixo das abcissas representa a dimensão do documento em palavras e as ordenadas o número de documentos

Os exemplos apresentam uma dimensão média de 202,09 palavras, com um desvio-padrão de 97,70, tendo a maior página a dimensão de 1371 palavras e a menor somente 67.

5.4.3 A representação dos documentos

Para a representação dos documentos optou-se por uma das formas mais simplificada da equação (59), forçando:

$$C_2 = C_3 = 0\,, \tag{81}$$

$$\text{e } se \quad n_f(i) > 0 \quad então \quad n_f(i) = 1\,, \tag{82}$$

$$\text{com } N_{Tot} = 1\,, \tag{83}$$

resultando no vector de existência de característica.

A condição (81) foi utilizada, tendo em conta que o desconhecimento *a priori* da relevância, no estudo de caso dos atributos de texto e da posição no texto na página obrigava à definição de valores para C_1 e C_2 sem uma justificação fundada em pressupostos válidos.

No caso da condição (82) e (83), o objectivo visou a redução do termo qualitativo de verdadeiro ou falso quanto à presença da palavra no documento.

5.4.4 A selecção de características

A selecção das características candidatas a pertencer ao vector de representação é uma tarefa essencial tendo em conta o elevado número de possíveis opções. A utilização de

todas tornaria o processo demorado e dificultaria a obtenção de bons resultados, tendo em conta a dimensão do universo. Desta forma, os processos implementados, apresentados em detalhe nas próximas secções, revelaram-se essenciais permitindo a indução eficiente de classificadores com elevado desempenho.

5.4.4.1 Lista de paragem

Para a remoção das características, através da utilização de listas de paragem, adoptou-se a lista de paragem DVL/Verity. Esta lista, criada pelo DTIC[44], contém a maioria das palavras que constam na lista de paragem inicial Verity (www.verity.com), à qual decidiram adicionar palavras da lista DTIC-DROLS[45]. A escolha deveu-se essencialmente à comprovada experiência do DTIC na utilização de textos em língua inglesa, especialmente, no conhecimento adquirido na base de dados DROLS. (A lista completa DVL/Verity com as 456 palavras pode ser encontrada no anexo A.7).

Após a filtragem dos documentos, o *corpus* ficou com um total de 32,477 características. A Tabela 9 apresenta a distribuição das características presentes pelas classes em análise.

Descrição	Número
Características presentes em documentos de venda	26 349
Características presentes em documentos de não venda	16 336
Características presentes exclusivamente em documentos de venda	15 941
Características presentes exclusivamente em documentos de não venda	5 928
Características presentes em ambos os tipos de documentos	10 608

Tabela 9 – Distribuição das características do *corpus* do DEEPSIA pelas classes em análise

A frequência média de ocorrência de uma característica **nos documentos de venda é de 20,84** com uma variância de 164,12, e a frequência média de ocorrência de uma característica **nos documentos normais é de 7,5**, com uma variância de 59,4.

Outra análise possível relaciona-se com a ocorrência das características nos documentos por categoria. A Figura 57 apresenta graficamente as características seleccionadas (por categoria de documento Venda, Normal e Ambas) após remoção por limiar de frequência crescente, i. e., o total de características consideradas, após a eliminação por ocorrência iguais ou inferiores ao limiar de frequência.

[44] O DTIC é um elemento fundamental na execução do Programa de Informação Científica e Tecnológica do Departamento de Defesa dos EUA. O DTIC é parte da Agência dos EUA, DISA *«Defense Information Systems Agency»*.

[45] DROLS «*Defense RDT&E Online System*» – Sistema interactivo em linha classificado que permite o acesso directo às bases de dados do DTIC.

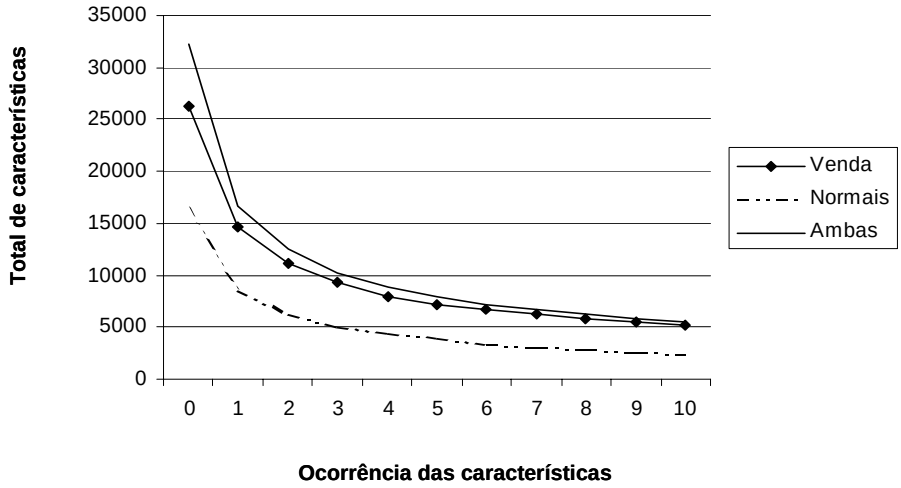


Figura 57 – O número de características consideradas tendo em conta a sua ocorrência, por cada categoria (Venda, Normal, Ambas). O eixo das abcissas representa o valor a partir do qual a característica é considerada, e o eixo das ordenadas apresenta o número total de características por cada categoria

Como é possível constatar, a maioria das características apresenta um reduzido número de ocorrências, e a análise dos dados revela, por exemplo, que 72,46 por cento das características presentes nos documentos de venda têm ocorrência inferior a 6 e 80,25 por cento inferior a 11. No caso dos documentos normais, os valores aumentam ligeiramente, atingindo os 75,99 por cento e 84,83 por cento, respectivamente, para o limite superior de 6 e 11.

Estes dados permitem antecipar tal como se esperava, que a maioria das características não são relevantes para a identificação do tipo de documentos.

5.4.4.2 Limiar de frequência mínimo

O método de selecção por limiar de frequência mínimo (LFM) utiliza a baixa ocorrência de uma característica para realizar a sua eliminação. Tendo em conta que não se identificou nenhum estudo que apresentasse um método para a sua definição prévia, optou-se pela determinação experimental, que foi realizada pela análise das ocorrências das características, previamente ordenadas pelo método da Informação Mútua e do Qui-quadrado. A Figura 58 apresenta a frequência de ocorrências das características em intervalos crescentes para os dois métodos. Os dados que deram origem aos gráficos são apresentados em formato tabela no anexo A.8.

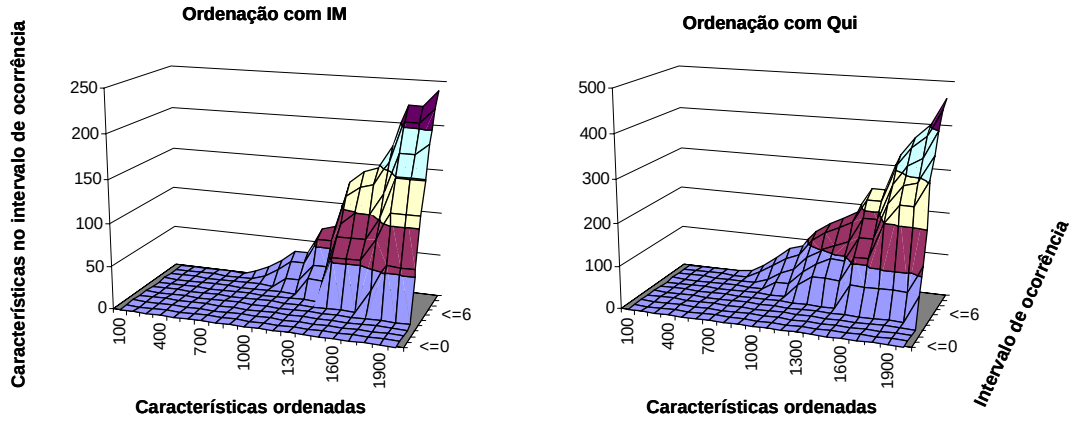


Figura 58 – Frequência das características, por intervalos de selecção, ordenadas pelos métodos da Informação Mútua e Qui-quadrado. As abcissas apresentam as características, o eixo em profundidade o intervalo de ocorrência e as ordenadas, o número de características no intervalo de ordenação, que apresentam a ocorrência analisada

A análise dos gráficos permite identificar que, em ambos os métodos, as primeiras 500 características apresentavam uma ocorrência superior a dez nas duas classes em estudo. Tendo em conta que se pesquisa um vector de dimensão bastante inferior, a utilização de um valor abaixo deste limite permite eliminar um conjunto superior de características. Uma vez mais nos dois métodos foi possível verificar, que com a utilização de um LFM=3, não se eliminaria nenhuma característica relevante entre as 2000 melhores.

Na construção do protótipo optou-se pela utilização de um LFM=5, o que permite reduzir o número de características a ordenar aproximadamente em 75 por cento, tendo reduzido o número de características candidatas para 7,969.

A introdução de um novo método de ordenação ou a alteração substancial do *corpus*, implicará, naturalmente, a determinação do LFM mais adequado.

Não foi utilizado o limiar de frequência superior (LFS) previsto na metodologia.

5.4.4.3 Critérios de ordenação

No processo de seriação das características utilizaram-se dois critérios de ordenação: a Informação Mútua (IM) e o Qui-quadrado[46](QQ). As listas de ordenação obtidas foram, tal como se esperava, muito semelhantes, devido à natureza dos métodos. Todavia permitiram uma validação mútua e a identificação de problemas que conduziram à maturação da necessidade de eliminar as variáveis correlacionadas. Realizaram-se diversas experiências, que permitiram identificar e seleccionar qual das ordenações obtidas apresentava, globalmente, o melhor desempenho para o caso concreto.

[46] Qui-Quadrado representado por vezes como X^2

A Tabela 10 apresenta as vinte melhores características seleccionadas pelo método IM e a respectiva posição ocupada no método X^2, permitindo uma comparação de resultados do topo das listas.

Característica	Ocorrência doc. Venda	Ocorrência doc. Normal	Posição IM	Posição QQ
#currency	2754	258	0	0
Price	2044	169	1	1
Input	2553	692	2	2
Form	2596	729	3	3
B	2625	840	4	4
X	856	119	5	5
#number	2844	1073	6	6
Quantity	592	70	7	8
Black	498	54	8	12
Radios	12	72	9	7
Actinicretailpricetext	175	0	10	52
Actinicprices	175	0	11	53
Height	477	58	12	22
Tr	2800	1037	13	9
Music	659	108	14	20
Table	2801	1038	15	10
Td	2800	1038	16	11
Actinicactions	179	2	17	59
Script	2230	710	18	14
Committed	0	40	19	18
Web	166	186	20	13

Tabela 10 – As vinte primeiras características da lista pelo método IM e a respectiva posição em QQ

A semelhança dos resultados obtidos pelos dois métodos é elevada, sendo coincidente até à sétima posição, altura em que é possível identificar alterações nas posições atribuídas. Com o aumento da dimensão da lista, a comparação das semelhanças passa a ser uma tarefa pouco clara, pelo que se utilizou um processo de análise gráfica do número de características comuns em intervalos crescentes com incrementos de uma característica. Por outras palavras, sendo

$$A = \{A_{1,}, \ldots, A_{k,}\} \subseteq \Delta, \tag{84}$$

$$B = \{B_{1,}, \ldots, B_{k,}\} \subseteq \Delta, \tag{85}$$

duas listas possíveis de k características do conjunto total de características Δ, e

$$\forall_{i \in [1,k]} F[i] = \sum_{j=1}^{i} \delta\left(A_j, \forall_{h \in [1,j]} B_h\right) \tag{86}$$

F corresponde a um vector de valores que permite visualizar o conjunto de características comuns para cada intervalo possível. Chama-se a atenção que, na equação (86), tendo em conta que A e B representam listas ordenadas sem repetição, a função $\delta\left(A_j, \forall_{h\in[1,j]}B_h\right)$ só apresenta dois valores possíveis, zero ou um, respectivamente, para o caso em que a característica A_j está presente na sublista B_h em análise.

A Figura 59 apresenta os resultados obtidos, tomando por base a lista de ordenação realizada com o método IMC. O gráfico da esquerda apresenta a comparação das listas até ao máximo de duzentas características, enquanto que o gráfico da direita realiza a mesma operação até às duas mil.

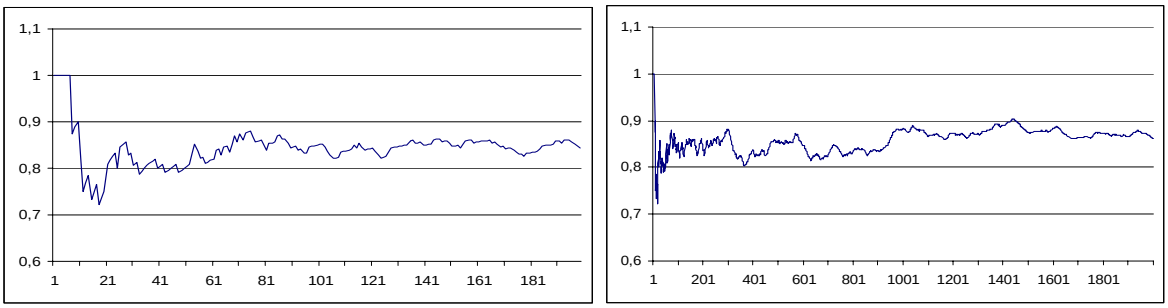


Figura 59 – Representação gráfica da semelhança entre a lista de ordenação de características obtida pelo método QQ, tendo como referência o método IM em intervalos crescentes de características. O eixo das abcissas representa os intervalos de comparação e o eixo das ordenadas o número de características comuns entre as duas listas no intervalo em estudo, F[i]

O grau de semelhança é total até à sexta característica, a partir da qual as listas deixam de ser coincidentes e passam a apresentar diferenças que têm o seu valor máximo de 0,722, quando as listas possuem 17 características. O valor médio da semelhança é de 0,858, com um desvio-padrão de 0,02447, o que atesta da equivalência global dos resultados obtidos pelos dois métodos.

Uma análise dos resultados apresentados na Tabela 10, contextualizada ao âmbito do sistema, consente uma explicação intuitiva baseada na relação directa da presença das características seleccionadas, principalmente nas páginas de venda.

No topo da lista (*#currency*[47]*, price, #number*[48]), respectivamente, o primeiro, segundo e sexto lugar nos dois métodos, estão características relacionadas com a existência de preços nas páginas de venda. O conjunto (*Input, Form)* está ligado ao processo de submissão de dados disponíveis no HTML, essencial para implementar a venda de produtos. A selecção de (*Tr, Table, Td)* está relacionada com o facto dos conceitos serem apresentados,

[47] «#currency» – representa a presença de um símbolo de moeda $ £ €.

[48] «#number» – representa a presença de um número.

usualmente, em tabelas e o conjunto (*B, quantity, Height)* com o modelo tradicional de apresentação e venda de produtos.

Existem algumas características que estão fortemente relacionadas com o processo de criação dos sítios, como são o caso das palavras de programação *(actinicretailpricetext, actinicprices, actinicactions)* e outras relacionadas com tipos de produtos específicos, como é o caso de *Music* e *Black.* Estas características são, naturalmente, pouco interessantes (devido à sua dependência), se o objectivo da identificação dos documentos não estiver limitado a um conjunto pré-selecionado de sítios. Todavia, no caso em estudo, a sua selecção é relevante, tendo em conta que tira partido do próprio método de construção dos sítios, em especial da meta-informação presente nas páginas e que permite a interacção com o utilizador.

A título de curiosidade a característica que despertou maior interesse foi o «*x*», por não permitir uma explicação intuitiva imediata. Todavia, uma análise dos documentos permitiu identificar que a sua selecção se deve ao facto da apresentação dos produtos incluir, normalmente, a dimensão num formato que separa as medidas através da letra "*x*" (e. g., 14 cm x 22 cm).

Resumindo, a maioria das vinte primeiras características foi seleccionada por estar relacionada com os documentos de venda, sendo excepções o «*rádios», «script», «commited», e «web».*

Como se depreende da análise efectuada, existe uma **elevada correlação entre as características seleccionadas**. Esta constatação conduziu à utilização do método de optimização IMC para eliminar as variáveis correlacionadas, enriquecendo, assim, a contribuição que cada característica oferece pela sua inclusão no vector de representação dos documentos.

5.4.4.4 Optimização IMC- Algoritmos genéticos

A optimização das listas obtidas foi realizada pelo método da Informação Mutua Conjunta (IMC). O cálculo de IMC foi realizado pelo recurso a algoritmos genéticos, o que conduziu à realização de diversas experiências com vista a identificar qual a melhor configuração dos parâmetros de configuração disponíveis (ver Tabela 5). As mais relevantes são apresentadas de seguida.

A utilização do conjunto de todas as características aumenta o tempo de execução dos algoritmos, pelo que foram realizadas diversas experiências com vista a **determinar qual a dimensão adequada para o cromossoma**.

O processo experimental consistiu na realização da evolução de populações, incrementando de forma progressiva a dimensão dos cromossomas, sendo registado o valor do melhor

cromossoma presente em cada geração. Foram realizadas experiências de optimização com as duas ordenações obtidas pelos métodos IM e do QQ. As experiências iniciaram-se com cromossomas de dimensão mínima de 256 *bits,* duplicando a sua dimensão até ao valor máximo 2048. Cada experiência foi realizada dez vezes e os valores apresentados são determinados pelo cálculo do valor médio do melhor cromossoma em cada geração.

As Figura 60 e Figura 61 representam a evolução do valor da função de avaliação do melhor cromossoma presente nas 250 primeiras populações utilizando, respectivamente, as variáveis presentes nas listas IM e do QQ. As experiências foram realizadas até à 3 000 geração, todavia os valores obtidos estabilizavam a partir da geração 150, pelo que os gráficos só apresentam os resultados até à geração 250.

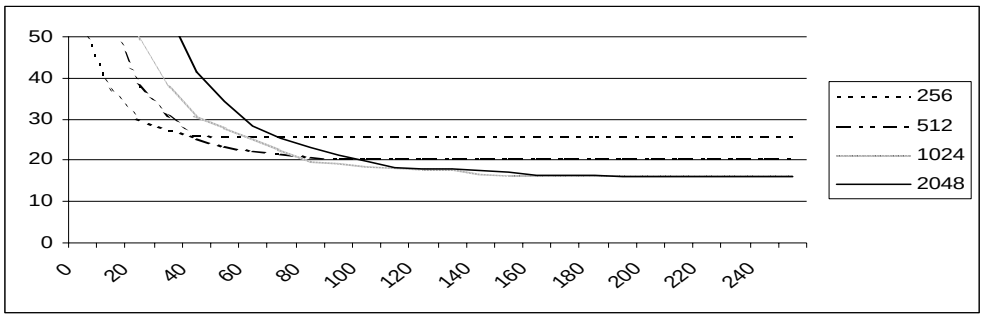


Figura 60 – Representação gráfica da evolução do melhor cromossoma presente nas duzentas e cinquenta primeiras gerações, utilizando a lista de ordenação resultante do método IM. O eixo das abcissas representa as gerações, e o eixo das ordenadas, o valor da função de avaliação

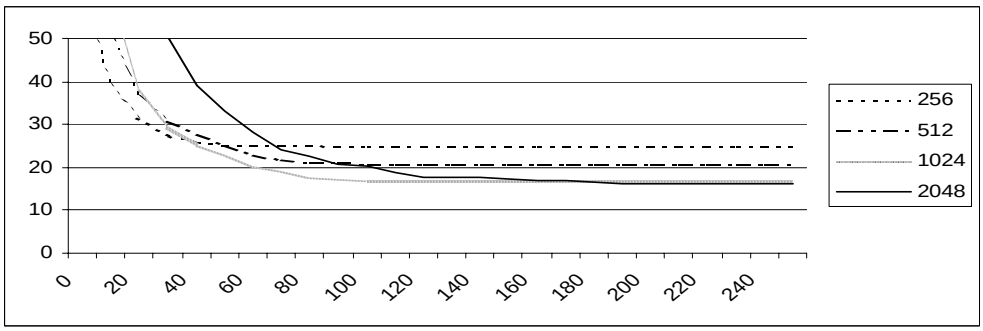


Figura 61 – Representação gráfica da evolução do melhor cromossoma presente nas duzentas e cinquenta primeiras gerações, utilizando a lista de ordenação resultante do método QQ. O eixo das abcissas representa as gerações, e o eixo das ordenadas, o valor da função de avaliação

Em todos os casos é visível que o valor da função de mérito do cromossoma evolui de forma constante e convergente (quanto menor o valor obtido melhor a função). O aumento da dimensão do cromossoma contribui, não só para uma convergência mais retardada, como para a obtenção de resultados muito inferiores nas gerações iniciais, consequência directa do maior espaço de pesquisa. Porém, o aumento da dimensão dos cromossomas permite a

obtenção de resultados finais superiores. Esta melhoria é muito significativa até atingirmos a dimensão de 1024 *bits*, momento em que não se obtêm incrementos substanciais.

Para simplificar a comparação dos resultados obtidos entre as duas listas, apresentam-se na Tabela 11, os melhores resultados obtidos e a geração em que foram atingidos, para todas as experiências efectuadas.

Experiência	256 IM	256 QQ	512 IM	512 QQ	1024 IM	1024 QQ	2048 IM	2048 QQ
Valor	25,566	24,904	20,395	20,544	16,369	17,048	15,895	16,0722
Geração	46	82	93	100	147	151	241	222

Tabela 11 – Melhores resultados obtidos para a função de avaliação

O melhor resultado absoluto foi atingido com a optimização da **lista IM, para cromossomas de dimensão 2048, na geração 241**, todavia, o resultado obtido é muito semelhante aos restantes não permitindo distinguir os dois métodos.

O aumento da dimensão para 2048 não permitiu atingir resultados substancialmente significativos o que, aliado aos elevados custos associados à sua obtenção, conduziu à finalização das experiências com esta dimensão máxima.

A análise das experiências anteriores permitiu identificar uma convergência muito rápida das populações. Este facto, conduziu **à suspeita de uma potencial perda de diversidade**, o que conduziria a uma convergência muito rápida para um máximo local.

Na experiência anterior, utilizou-se mutação não-efectiva com uma probabilidade de ocorrência de 1 por cento. O facto da mutação ser não efectiva, diminui o número de *bits* afectados, (tendo em conta que a mutação é realizada por troca de dois *bits*), o que, no caso concreto, corresponde, na maioria dos casos, à troca de um zero por outro zero. Por outras palavras, a mutação só acontece efectivamente, quando se trocam *bits* de valor diferente. A probabilidade de estarmos a trocar dois *bits* diferentes é de

$$P_1 = K / L \tag{87}$$

$$P_0 = 1 - P_1 = 1 - K / L = \frac{L - K}{L} \tag{88}$$

$$P_{diferentes} = P_{1,0} + P_{0,1} = 2\frac{K}{L} \times \frac{L - K}{L} = 2\frac{K(L - K)}{L^2} \tag{89}$$

$$P_{Efectiva} = P_m \, x \, P_{diferentes} \tag{90}$$

sendo P_1 a probabilidade de um *bit* estar a um, P_0 a probabilidade de estar a zero, e $P_{1,0}$ e $P_{0,1}$, respectivamente a probabilidade de, sendo seleccionado um *bit* a zero, se trocar por um *bit* a um e vice-versa.

No caso das experiências com cromossomas de maior dimensão, temos $L = 2048$ e $K = 100$ o que, por substituição em (89) permite obter uma probabilidade de troca de *bits* diferentes de 0.0928, o que reduz a probabilidade efectiva ($P_{Efectiva}$) para 9.28×10^{-4}.

Realizaram-se, então, experiências com vista a aumentar a diversidade das populações com o recurso a alteração do operador para mutação efectiva. Neste caso, a ocorrência de mutação obriga à troca do valor do *bit* seleccionado por outro de valor inverso. A Figura 62 apresenta os melhores resultados obtidos com mutação efectiva, (uma vez mais para diferentes dimensões de cromossomas) e, como referência, o melhor valor obtido para cromossomas de dimensão 2048 sem mutação efectiva (2048 SW).

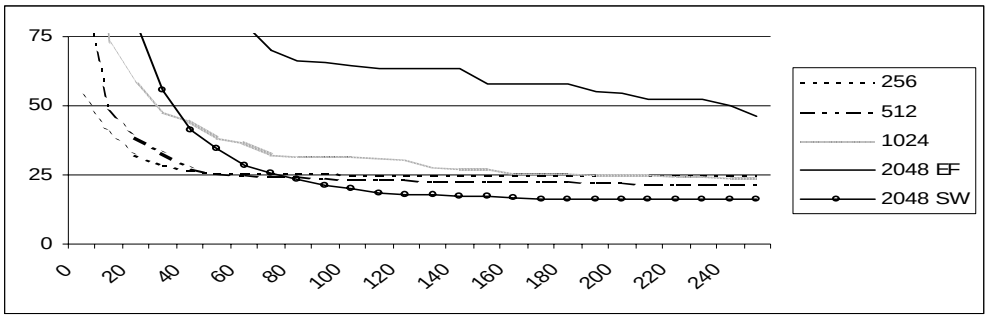


Figura 62 – Representação gráfica da evolução do melhor cromossoma presente nas duzentas e cinquenta primeiras gerações com mutação efectiva, à excepção de 2048 SW. Lista de ordenação original resultante do método QQ. O eixo das abcissas representa as gerações e o eixo das ordenadas, o valor da função de avaliação

Nesta nova configuração, ao contrário do que se esperava, não foi possível obter melhores resultados, o que **não confirmou a hipótese de localização em máximos locais por falta de diversidade**. Todavia, e tal como se esperava, a existência de mutação efectiva atrasou substancialmente o processo de convergência, **o que conduziu à sua não utilização**.

A Tabela 12 apresenta, uma vez mais, um exemplo das vinte melhores características seleccionadas pelo método IM e QQ, incluindo agora os resultados da optimização pela IMC.

O caso apresentado foi seleccionado, por ser representativo do conjunto de soluções encontradas no decurso das experiências efectuadas. De uma forma genérica, **a maioria das características previamente consideradas como relevantes, são eliminadas no processo de optimização**, tendo em conta a elevada correlação entre as características presentes na lista inicial. Todavia, como se pode verificar, são seleccionadas **características de cada conjunto de características correlacionadas, eliminando redundância sem perda de informação.**

Lista IM	Optimização
#currency	0
Price	Eliminada
Input	Eliminada
Form	Eliminada
B	Eliminada
X	1
#number	Eliminada
Quantity	Eliminada
Black	2
Radios	Eliminada
Actinicretailpricetext	Eliminada
Actinicprices	Eliminada
Height	Eliminada
Tr	Eliminada
Music	Eliminada
Table	Eliminada
Td	3
Actinicactions	4
Script	5
Committed	Eliminada

Lista QQ	Optimização
#currency	0
Price	Eliminada
Input	Eliminada
Form	Eliminada
B	Eliminada
X	1
#number	Eliminada
Rádios	Eliminada
Quantity	Eliminada
Tr	Eliminada
Table	Eliminada
Td	Eliminada
Black	2
Web	3
Script	4
Address	Eliminada
Credit	Eliminada
Exe	Eliminada
Committed	Eliminada
Receivers	Eliminada

Tabela 12 – Apresentação das vinte melhores características seleccionadas pelo método IM e QQ, e o resultado da optimização efectuado pelo processo da Informação Mútua conjunta

5.4.4.5 Análise comparativa de desempenho das ordenações

A aplicação dos dois métodos de selecção de características e as suas respectivas optimizações permitiram a obtenção de quatro ordenações, Informação Mútua (IM), Qui-Quadrado (QQ), IM Optimizada (IMO) e QQ optimizada QQO. As experiências seguintes procuram identificar a melhor lista para o caso concreto.

O primeiro estudo realizado consistiu na **utilização das quatro listas** para a classificação das páginas com o recurso ao **algoritmo de classificação de vizinhos mais próximos com K=1**. A escolha do algoritmo dos vizinhos mais próximos prende-se com a sensibilidade do mesmo à qualidade das características que representam a página, sendo pouco imune ao ruído.

A experimentação realizada consistiu em utilizar o classificador com vectores de representação de dimensão crescente, com incrementos de dez, o que permitiu avaliar a influência da introdução de novas características e a qualidade das características seleccionadas para o topo das listas. Os próximos gráficos apresentam a precisão, a rechamada e o valor de F1 para as diversas dimensões de vector. A análise comparativa dos dois primeiros gráficos deve ter em conta a diferença de escala que pode ser enganadora. Na rechamada os valores oscilam entre duas décimas, enquanto que na

precisão os valores variam em dez décimas, pelo que a influência nos resultados obtidos pela métrica F1 é muito superior no caso dos valores da precisão, por oposição dos valores da rechamada[49].

Tal como era esperado, os melhores resultados para os vectores de menor dimensão foram atingidos pelas listas optimizadas. Este resultado não surpreendeu, tendo em conta que a influência de variáveis correlacionadas é tanto mais relevante quanto menor for o vector. Quanto menor for o vector mais crítica é a contribuição de cada característica e, logo, a introdução de características que não contribuem com poder discriminativo induz resultados mais visíveis.

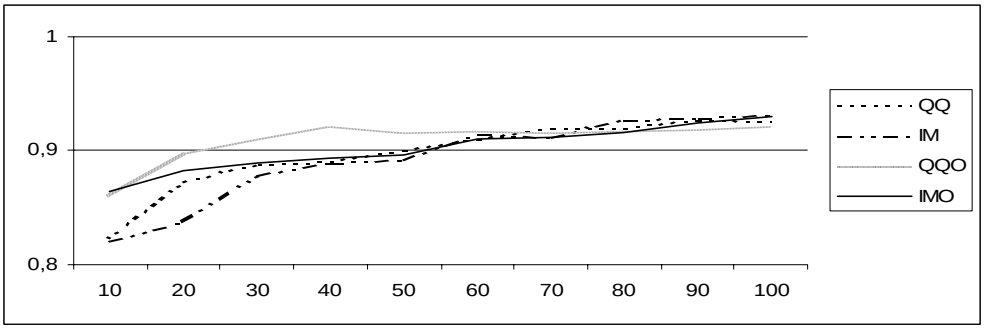


Figura 63 – Evolução da precisão dos classificadores de 1-vizinho utilizando vectores de dimensão crescente para as quatro listas de ordenação

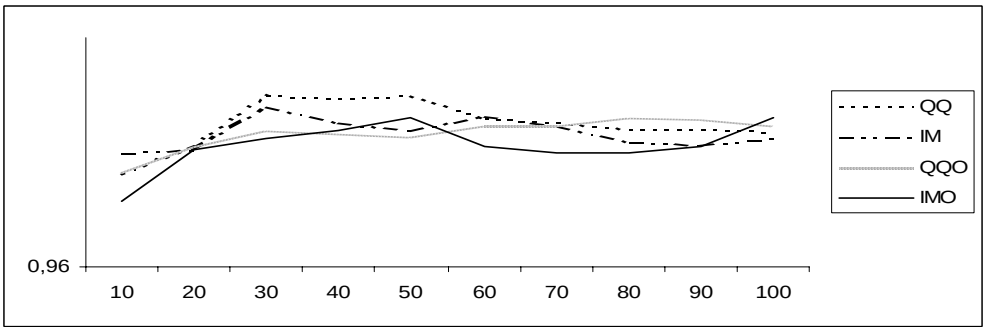


Figura 64 – Evolução da rechamada dos classificadores de 1-vizinho utilizando vectores de dimensão crescente para as quatro listas de ordenação

Para além de ser possível identificar o melhor desempenho das listas optimizadas, é igualmente possível verificar **o melhor desempenho do método de Qui-quadrado**. Analisando os resultados, agrupados em listas optimizadas e listas não optimizadas chega-se à conclusão que, nos dois casos, o desempenho do método qui-quadrado é superior, especialmente para os vectores de menor dimensão.

[49] Esta chamada de atenção não se restringe aos resultados obtidos exclusivamente nesta experimentação, sendo comum a maioria das experiências realizadas pelo que deve ser tomada em consideração na análise das restantes experiências.

Os resultados superiores apresentados pelo qui-quadrado optimizado podem ser justificados **pelo facto do método tirar partido das duas técnicas**, uma vez que a primeira ordenação é realizada com o qui-quadrado, mas a optimização é realizada com o método da informação mútua conjunta. Desta forma, a ordenação é conseguida com a reunião dos dois métodos.

Todavia, as conclusões anteriores não se mantêm para vectores de dimensão igual ou superior a setenta características, altura em que o desempenho de todas as listas é muito semelhante, não permitindo afirmar, a partir desse momento, que um método é preferível.

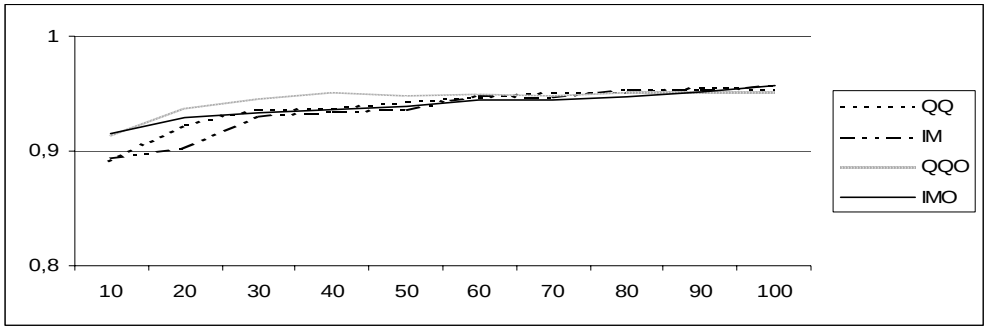


Figura 65 – Evolução da métrica F1 dos classificadores de 1-vizinho utilizando vectores de dimensão crescente para as quatro listas de ordenação

A última constatação é corroborada pela métrica F1, apresentada na Figura 65, que apresenta resultados muito semelhantes para os valores obtidos para vectores com mais de setenta características. O gráfico não permite observar, mas os valores obtidos para vectores superiores a cem características estabilizam nos valores obtidos verificando-se inclusive, por vezes, uma ligeira diminuição do desempenho global.

Com vista a validar as observações efectuadas, realizou-se **uma segunda experiência,** apresentada nas figuras seguintes, **agora para três vizinhos**. Genericamente **os resultados são equivalentes**, os valores obtidos com **as listas optimizadas são superiores, o aumento da dimensão dos vectores atenua as diferenças e os valores obtidos com os métodos qui-quadrado no início são superiores aos obtidos com a Informação Mútua**.

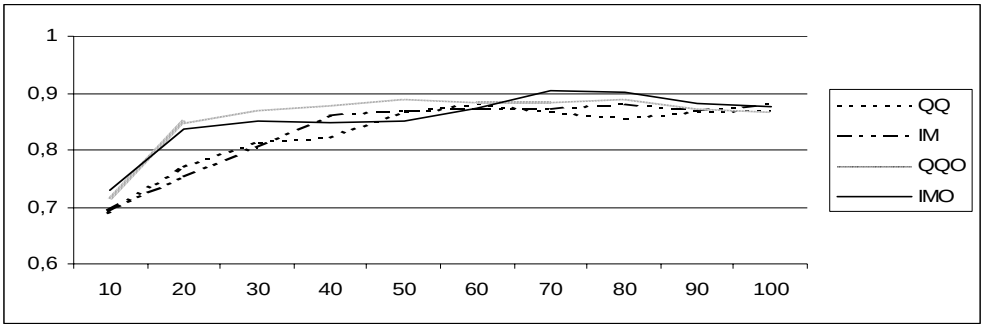


Figura 66 – Evolução da precisão dos classificadores com a utilização de vectores de dimensão crescente usando três vizinhos

Todavia, os **resultados globais são inferiores**, apesar das listas da **informação mútua apresentarem um resultado relativo superior**, visíveis nos resultados da rechamada e de F1, onde é possível verificar que os resultados da Informação Mútua optimizados são superiores, em especial no intervalo entre as sessenta e noventa características.

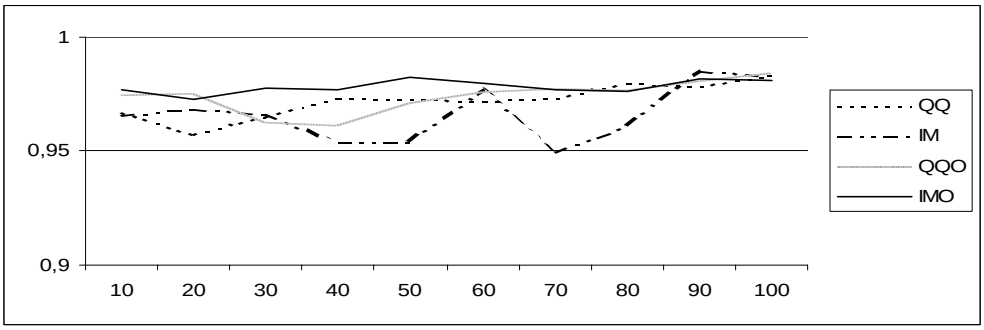


Figura 67 – Evolução da rechamada dos classificadores com a utilização de vectores de dimensão crescente para três vizinhos

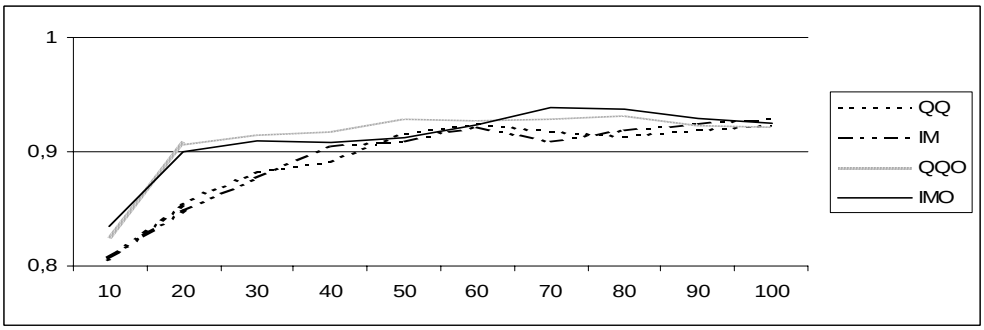


Figura 68 – Evolução da métrica F1 dos classificadores com a utilização de vectores de dimensão crescente para três vizinhos

A ordenação seleccionada para as etapas seguintes foi a conseguida pelo método do qui-quadrado optimizado obtido com cromossomas de dimensão 2048 sem mutação efectiva.

5.4.5 Os classificadores

Com o objectivo de identificar a técnica mais adequada para o problema em estudo, foram induzidos diversos classificadores tendo por base a lista de características previamente seleccionadas. As experiências realizadas visaram não só identificar o melhor método mas, igualmente, qual a melhor configuração para cada método.

5.4.5.1 Método dos k-vizi*nhos*

O método dos k-vizinhos foi avaliado em diversas experiências para determinação da melhor configuração possível.

A primeira experiência procura avaliar a influência da variação do número de características no desempenho do classificador, utilizando um único vizinho. Os resultados do desempenho para os documentos de venda são apresentados na Figura 69 e, para os documentos normais na Figura 70. As experiências foram efectuadas até às 1000 características, contudo os gráficos só apresentam os resultados até ao momento em que deixam de existir alterações no desempenho, mais concretamente até às 120 características.

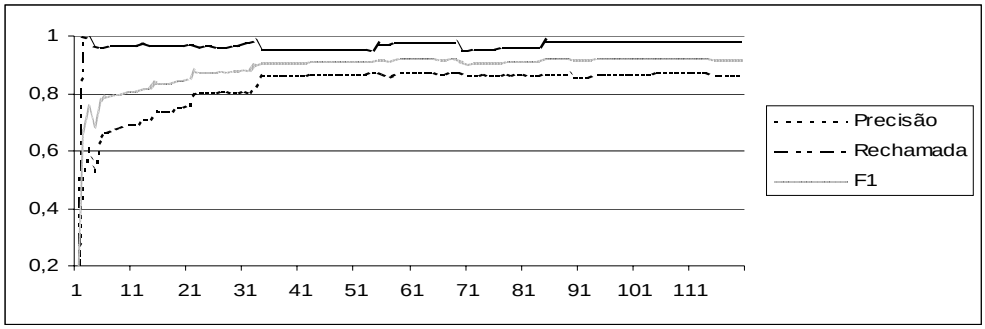


Figura 69 – Resultados do desempenho do classificador 1-vizinho com o aumento até 120 do número de características. O eixo das abcissas representa o número de características utilizadas e o eixo das ordenadas, os resultados das métricas em estudo para as páginas de venda

Tal como se esperava, em ambos os casos, o aumento do número de características permitiu, numa primeira fase, melhorar o desempenho do estimador, após o qual o seu comportamento estabiliza.

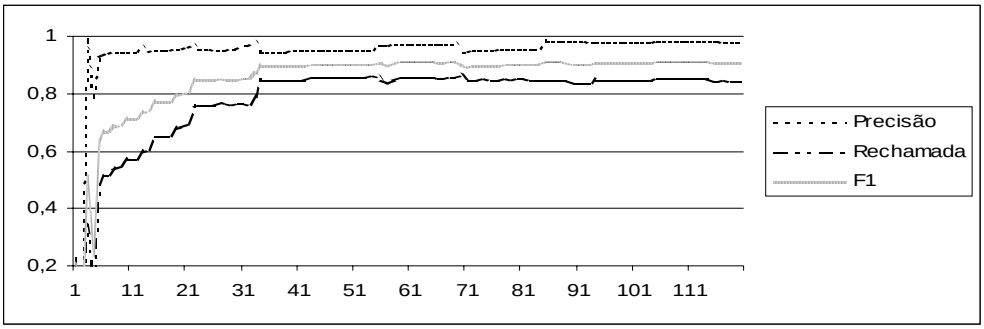


Figura 70 – Resultados do desempenho do classificador 1-vizinho com o aumento até 120 do número de características. O eixo das abcissas representa o número de características utilizadas e o eixo das ordenadas, os resultados das métricas em estudo para as páginas normais

Nesta análise, optou-se por apresentar os resultados para as duas classes, demonstrando experimentalmente a inter-relação, previamente descrita, das duas métricas. Os valores para a precisão e rechamada, na classificação dos documentos de venda, têm um comportamento inverso aos correspondentes nos documentos normais e vice-versa. O comportamento do método para a classe normal é, assim, o inverso do descrito para a classe das vendas.

Os resultados com vectores de representação até quatro características são extremamente negativos. Na prática, o classificador limita-se a considerar todas as páginas como pertencentes à categoria de vendas, o que é verificável pelo valor extremamente elevado da rechamada, para o caso categoria vendas, acompanhado, naturalmente, de uma precisão muito baixa. Conclui-se assim, que com muito poucas características, o método não é capaz de reconhecer os documentos, o que permite afirmar que **as características do topo da lista, apesar de importantes, são insuficientes para descrever cabalmente os documentos**.

A partir da inclusão da quinta característica o classificador começa finalmente a apresentar resultados interessantes e progressivamente melhores, até se chegar a vectores de representação de 120 características, altura a partir da qual o desempenho estabiliza. O valor da métrica F1 atinge o seu máximo às 114 características.

A Tabela 13 apresenta os valores mínimos e máximos para cada uma das métricas no caso da classificação de documentos de venda.

	Precisão		**Rechamada**		**F1**	
	Posição	Valor	Posição	Valor	Posição	Valor
Mínimo	4	0,5285	71	0,9490	4	0,6829
Máximo	70	0,8764	90	0.9855	114	0,9252

Tabela 13 – Valores mínimos e máximos para as três métricas na classificação dos documentos de venda

Da sua análise é possível observar a influência determinante na métrica F1 dos resultados obtidos para a precisão, o que acontece devido à baixa variância da rechamada, somente 0,012 que é aproximadamente seis vezes inferior à variância da precisão 0,074. O pior e o melhor resultado para a métrica F1 são obtidos para valores semelhantes da precisão, respectivamente com quatro e com cento e catorze características.

A experiência seguinte procura avaliar a influência do número de vizinhos no desempenho do classificador. Para tal utilizou-se o melhor caso anterior (com as melhores cento e catorze características) e analisou-se o desempenho dos classificadores alterando o número de vizinhos para o cálculo da estimativa. Os resultados são apresentados na Figura 71 e permitem concluir que, no caso concreto, o aumento dos vizinhos não é determinante, reduzindo inclusive a precisão e, consequentemente, a métrica F1. Os resultados apresentados são exclusivamente referentes à classificação de documentos normais, uma vez que os resultados obtidos para a classe de vendas são complementares.

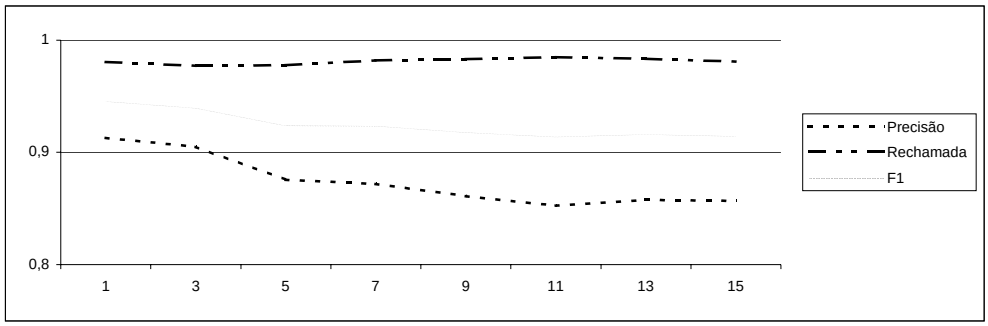


Figura 71 – Resultados com aumento do número de vizinhos para o conjunto das melhores 160 características da ordenação da qui-quadrado optimizada. O eixo das abcissas representa o número de vizinhos considerados e o eixo das ordenadas, os resultados para as métricas das classificação de documentos de venda

O primeiro aumento, até três vizinhos, não altera significantemente os resultados. Todavia, os seguintes comprometem a precisão, não compensando a ligeira melhoria dos resultados da rechamada. Desta forma, é possível concluir que, para o caso concreto, o aumento dos vizinhos não apresenta resultados favoráveis, **pelo que o número de vizinhos a considerar deve ser de um.**

5.4.5.2 Método Naive Bayes

O método Naive Bayes foi analisado em seguida, tendo-se realizado um estudo para avaliar qual o número mais adequado de características. Os resultados são apresentados nas Figura 72 e Figura 73 e apresentam o desempenho, respectivamente, para a classificação de documentos de venda e normais.

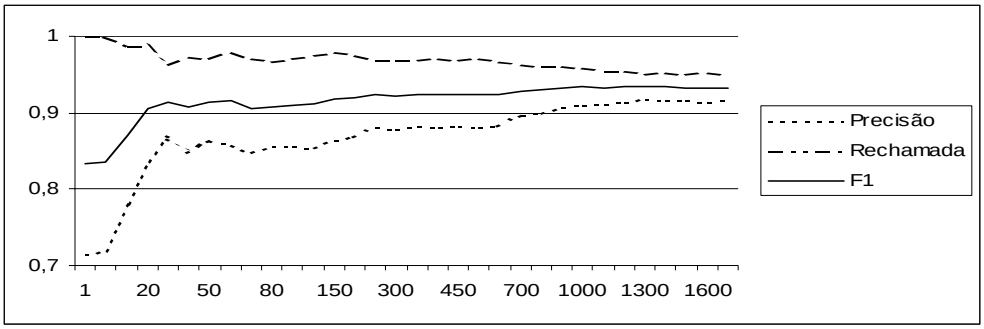


Figura 72 – Resultados do desempenho do classificador Naive Bayes com aumento do número de características na classificação de documentos de venda. O eixo das abcissas representa o número de características utilizadas no processo de indução do classificador e o eixo das ordenadas, os resultados para as métricas em estudo

A análise dos dados permite concluir que o classificador só começa a ser eficaz após as vinte características, altura a partir da qual apresenta um comportamento incremental até estabilizar, perto do seu máximo, após as 1300 características.

Na primeira fase, o classificador é incapaz de reconhecer os documentos limitando-se a marcar a esmagadora maioria como pertencente à categoria de vendas. Neste estádio, a métrica de rechamada obtém valores excelentes, chegando a atingir, com a utilização de uma única característica os 100 por cento, o que é natural uma vez que todos os documentos são classificados na categoria de venda não existe perda de informação relevante. Contudo, a precisão é, muito baixa, compensando desta forma a influência positiva da rechamada na métrica F1.

Na segunda fase, com a introdução de novas características, o vector de representação passa a ser significativo, contribuindo, consequentemente, para assegurar a diminuição do grau de incerteza associado a cada observação, e para melhorar a eficácia, o que acontece até às 1300 características, altura em que estabiliza.

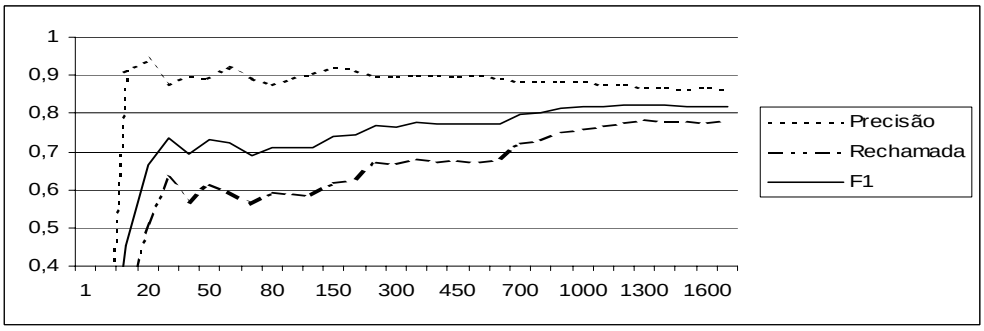


Figura 73 – Resultados do desempenho do classificador Naive Bayes com aumento do número de características na classificação de documentos de normais. O eixo das abcissas representa o número de características utilizadas no processo de indução do classificador e o eixo das ordenadas, os resultados para as métricas em estudo

É importante chamar à atenção que a estabilização das métricas nas duas categorias verifica-se em valores distintos, sendo superior no caso da precisão. Esta diferença está relacionada com o peso relativo que cada classificação possui na avaliação final. No caso concreto, o peso de cada nova classificação é muito superior na categoria normal, devido à sua menor representatividade no *corpus*.

	F1 (venda)		**F1 (normais)**	
	Posição	Valor	Posição	Valor
Mínimo	1	0,8331	1	0
Máximo	1200	0.9344	1300	0,8233

Tabela 14 – Valores mínimos e máximos para a métrica F1 nas duas classes de classificação.

A Tabela 14 apresenta os valores extremos obtidos para a métrica F1 permitindo indicar que um bom vector de representação possuirá uma dimensão entre as 1200 e as 1300 **características.**

5.4.5.3 Métodos de árvore de decisão

O método C4.5, assim como o método C4.5 iterado, foram, igualmente, testados na procura do melhor desempenho. Os primeiros testes foram efectuados para o C4.5 e permitiram verificar que os resultados obtidos pelas árvores com poda foram superiores aos resultados das árvores sem poda. Procurou-se, posteriormente, identificar experimentalmente qual seria a melhor configuração para os parâmetros de poda. A experiência inicial consistiu em alterar o número de características empregues e dois parâmetros do método de indução: a *confiança para estimativa de erro na poda (C)* e *o número mínimo de elementos presentes num determinado nó (M*). Os melhores resultados são apresentados na Tabela 15, incluindo os valores de caracterização das árvores obtidas (profundidade e número de nós).

A análise dos dados permite concluir que *M* é o parâmetro determinante, pois os melhores resultados estão associados, invariavelmente, às experiências com *M*=2 (valores superiores para este parâmetro conduziram invariavelmente à diminuição de desempenho na métrica F1). A melhoria do desempenho (devido ao *M=2*) é, todavia, acompanhada pela diminuição da eficiência, uma vez que as árvores induzidas têm uma profundidade muito superior e utilizam um maior número de nós. Os melhores resultados obtidos para o *M=2* acontecem para valores de factor de confiança entre os 50 por cento e os 100 por cento. Os três melhores resultados assinalados na tabela, apresentam, por coincidência, o valor de 0,9753 para a métrica F1.

M	C	Caract.	F1	Rechamada	Precisão	Profundidade	N. Nós
50	50	900	0,9524	0,9810	0,9253	3,6	8,2
50	5	60	0,9525	0,9842	0,9227	3	7
50	25	900	0,9530	0,9838	0,9240	3,4	7,8
50	20	900	0,9530	0,9838	0,9240	3,4	7,8
20	5	60	0,9598	0,9835	0,9372	5,1	13,2
20	25	60	0,9610	0,9842	0,9388	5,2	13,8
20	20	60	0,9611	0,9838	0,9395	5,2	13,8
20	50	60	0,9613	0,9842	0,9395	5,4	14,2
10	5	1600	0,9658	0,9863	0,9462	9,3	25
10	25	300	0,9681	0,9842	0,9526	11,2	33,6
10	20	300	0,9681	0,9842	0,9526	11,1	33,4
10	50	1900	0,9682	0,9831	0,9537	13,1	35
5	5	1200	0,9682	0,9873	0,9497	15,2	39,4
2	0,001	1700	0,9702	0,9873	0,9536	23,6	59
2	1	1700	0,9710	0,9870	0,9555	23,9	60,6
5	50	1400	0,9711	0,9835	0,9590	23,5	69,4
5	20	150	0,9716	0,9838	0,9596	16,1	55,2
2	5	1700	0,9718	0,9870	0,9571	24,9	63,6
5	25	150	0,9725	0,9817	0,9635	16,1	58,2
2	10	1700	0,9727	0,9873	0,9585	26,1	67,6
2	20	1900	0,9728	0,9863	0,9598	27,2	72,2
2	25	2200	0,9737	0,9866	0,9610	28,9	74,8
2	30	1900	0,9742	0,9845	0,9640	34	94,6
2	50	1800	0,9751	0,9831	0,9673	36,4	108,2
2	70	600	**0,9753**	**0,9821**	**0,9685**	32,8	126,8
2	100	1700	**0,9753**	**0,9838**	**0,9670**	36,3	111
2	90	1700	**0,9753**	**0,9838**	**0,9670**	36,3	111

Tabela 15 – Resultados experimentais do impacto da variação dos parâmetros de indução do método C4.5. Os resultados estão apresentados por ordem crescente de desempenho, respeitando a ordenação imposta por F1, seguida de rechamada e precisão

Procurou-se, em seguida, com M=2 e C=90, identificar qual o melhor número de características na indução das árvore de decisão. Os resultados experimentais são apresentados nas próximas figuras e permitem concluir, uma vez mais, que o aumento de características conduz a melhoria de desempenho.

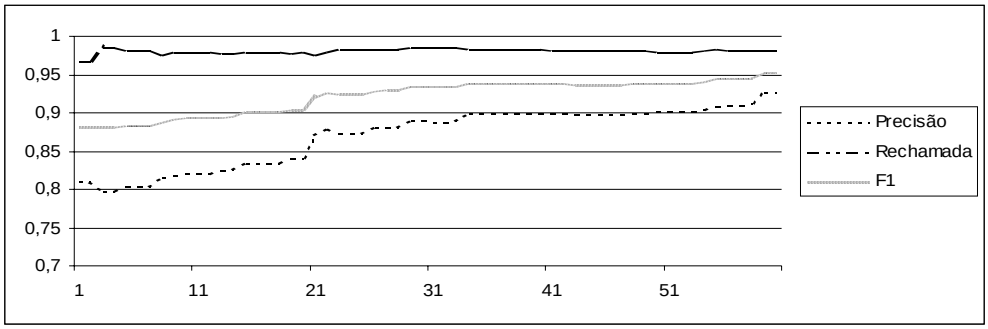


Figura 74 – Resultados do desempenho do classificador C4.5 com aumento, até 60, do número de características. O eixo das abcissas representa o número de características utilizadas no processo de indução do classificador e o eixo das ordenadas, os resultados para as métricas em estudo para as páginas de venda

A Figura 74 permite, uma vez mais, constatar que utilizando poucas características não se obtêm resultados adequados. Somente com o aumento do número de características é possível obter um classificador com um desempenho interessante. Os resultados estabilizam a partir das sessenta características como se pode verificar na Figura 75.

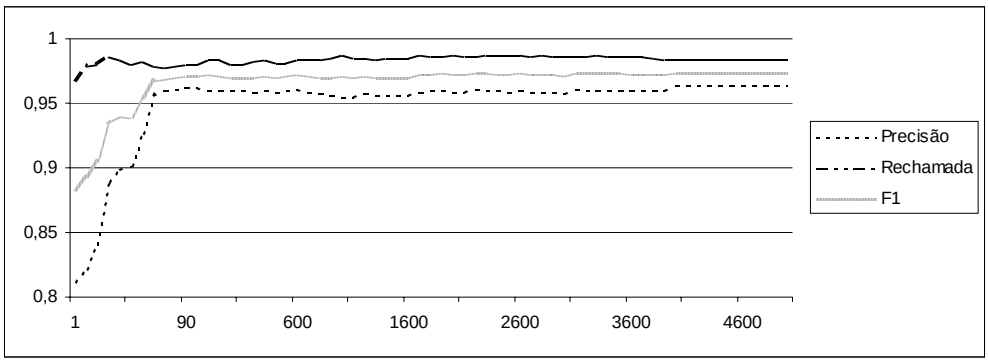


Figura 75 – Resultados do desempenho do classificador C4.5 com aumento, até 5000, do número de características. O eixo das abcissas representa o número de características utilizadas no processo de indução do classificador e o eixo das ordenadas, os resultados para as métricas em estudo para as páginas de venda

Os melhores resultados absolutos foram obtidos com a utilização de um vector de representação com 1700 características.

A experiência seguinte procurou avaliar o **método C4.5 iterativo,** que foi testado utilizando os mesmos valores para os parâmetros de indução, vector de representação com 1700 características, *M=2* e *C=90* e fazendo-se variar, agora, o número de características inseridas em cada iteração. Obtiveram-se os resultados apresentados na Tabela 16.

Nº de caract. inseridas em cada iteração (NC)	Rechamada	Precisão	F1	Profundidade	Nº Nós
1	0,9814	0,9579	0,9695	5,7	14,7
2	0,9828	0,9607	0,9716	6	17,1
5	0,9796	0,9665	0,9730	6,8	20,6
10	0,9757	0,9707	0,9732	7,7	22,6
20	0,9793	0,9711	**0,9752**	8,6	27,5
25	0,9786	0,9689	0,9737	9,3	28
50	0,9747	0,9687	0,9717	10,3	31,7
100	0,9729	0,9689	0,9709	12,8	31,3

Tabela 16 – Resultados do algoritmo C4.5 iterativo para os parâmetros *C=90* e *M=2*, sem remover características não utilizadas

Como se pode observar, o número de características inseridas em cada iteração (NC) permite aumentar ligeiramente o desempenho das árvores, até um determinado momento, altura em que os resultados começam a piorar. O máximo absoluto verifica-se com a inserção de grupos de vinte características em cada nova iteração. A compactação das árvores é igualmente afectada, piorando com o aumento do número de características inseridas.

Comparativamente com o C4.5 original, os resultados demonstram que os dois algoritmos apresentam um desempenho semelhante. Por coincidência os dois algoritmos apresentam o mesmo valor máximo de 0,975 para a métrica F1.

Contudo, comparando as características das árvores geradas (os valores da profundidade e do número de nós das árvores induzidas), **verifica-se que a compactação é muito superior no algoritmo C4.5 iterativo, conseguindo-se, assim, árvores mais eficientes**.

A Tabela 17 apresenta a caracterização das árvores obtidas utilizando os dois algoritmos. Para o C4.5 foram utilizados os parâmetros *M=2* e *C=90* e para o C4.5i, os parâmetros *M=2, C=90 e NC=20*.

Nº Caracteristicas	Profundidade		Número de Nós	
	C4.5i	C4.5	C4.5i	C4.5
60	17,2	15,4	18	80,2
70	15,2	16,4	18,6	79,2
80	16,2	18,3	18,8	87,4
90	15,7	18,3	20,6	83,4
100	15,3	18,6	21,3	82,8
150	13,5	20,6	22,2	84,8
200	13,1	21,6	22,4	81,2
250	13,1	22,8	23,8	88,8
300	12,7	23,9	22,7	84,6
350	12,6	23,4	23,1	82
400	12,5	23,9	22,2	83
450	12,5	24,3	23,1	86,4
500	12,2	25,2	22,7	86,8
600	12,4	25,4	22,6	86,6
700	12,4	25,3	23,1	82,8
800	12,3	27,9	23,2	83,4
900	11,6	26	22,8	77,2
1000	11,4	27,1	22,7	79
1100	11,8	26,8	23,3	78
1200	12,1	27	23,2	78,6
1300	12	27,6	23	79
1400	11,9	28	23,4	78
1500	11,4	28,2	23,3	77,8
1600	11,4	28,5	23,3	79
1700	10,6	28	23,8	75,6

Tabela 17 – Resultados da compactação (profundidade e número de nós) das árvores induzidas através dos algoritmos C4.5 e C4.5i para diferentes dimensões de vectores de representação

A análise da tabela permite concluir que no caso do único factor relevante ser o desempenho, o C4.5 iterativo não deve ser utilizado, uma vez que tem uma fase de aprendizagem muito mais lenta, todavia se para além da eficácia, o factor determinante for a eficiência, o algoritmo deve ser considerado.

5.4.5.4 Análise comparativa dos classificadores

Nesta secção comparam-se os melhores classificadores induzidos através dos diversos métodos utilizados. A Tabela 18 apresenta os valores obtidos e o número de características utilizadas na indução do classificador.

	Precisão		Rechamada		F1	
	N. Caract.	Valor	N. Caract.	Valor	N. Caract.	Valor
1-vizinho	70	0,8764	90	0.9855	114	0,9252
Naive Bayes	1300	0,9169	1	1,0	1200	0,9344
C4.5	600	0,9685	1200	0,9873	1700	**0,9753**
C4.5i	1700 (20)	0,9711	1700 (2)	0,9828	1700 (20)	0,9752

Tabela 18 – Melhores resultados dos classificadores induzidos através das diversas técnicas para as vendas

É necessário ter presente que o número de características apresentado, somente corresponde à dimensão do vector de representação, para o caso do 1-Vizinho e do Naive Bayes. O significado para as árvores é do número de características utilizadas na indução. No caso do algoritmo C4.5i apresenta-se entre parêntesis o número de características inseridas em cada iteração.

Os valores obtidos com as árvores são muito superiores, tendo em consideração que nos melhores casos ultrapassam para F1 o valor de 0,975, por contraste com o Naive Bayes que nunca ultrapassou os 0,9344, e com o 1-Vizinho que obteve como melhor resultado 0,952.

A vantagem, aparente, registada com a utilização de poucas características para 1-vizinho resulta exclusivamente da apresentação dos resultados, é necessário manter presente que as árvores apresentam bons resultados mesmo com poucas características, a título de exemplo, com 60 características os resultados do C4.5 já são superiores a 0,96.

Tal como já tinha sido referido os resultados obtidos para as árvores são muito semelhantes quanto ao desempenho, contudo **o algoritmo C4.5i apresenta resultados muito superiores** se tivermos em consideração a eficiência.

5.4.6 O sistema de Suporte à Decisão

O último passo para a construção do SAD é decidir qual o processo de tomada de decisão a adoptar (PTD). No caso concreto, optou-se por experimentar três tipos de PTD, (pré-selecção, maioria e o método Fajé), utilizando diversos classificadores induzidos através de métodos distintos e das variações estudadas.

O PTD construído por pré-seleção, serve de referência, e foi obtido, naturalmente, com o melhor classificador induzido pelos métodos anteriores, o C4.5.

Nos restantes tipos de PTD, os melhores resultados foram **invariavelmente** obtidos com as árvores de decisão, pelo que a utilização de outros classificadores (k-vizinho ou Naive Bayes) de menor desempenho, nunca contribuiu para a obtenção de melhores resultados.

A Tabela 19 apresenta o **conjunto de classificadores seleccionados aleatoriamente** utilizados na construção do PTD apresentado na Tabela 20.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	F1
C1	0,9330	0,9567	**0,9655**	0,9492	0,9380	0,9578	**0,9182**	0,9619	0,9590	0,9394	0,94787
C2	0,9297	0,9482	0,9643	0,9565	0,9460	0,9500	**0,9220**	0,9637	0,9279	**0,9703**	0,94786
C3	**0,9125**	0,9545	0,9474	0,9496	0,9478	0,9518	0,9315	**0,9619**	0,9595	0,9496	0,94661
C4	**0,9218**	0,9369	0,9550	0,9460	0,9524	0,9536	0,9274	**0,9595**	0,9500	0,9460	0,94486
C5	**0,9276**	0,9532	0,9478	0,9460	0,9483	0,9578	0,9315	**0,9595**	0,9487	0,9559	0,94763
C6	0,9200	0,9429	0,9514	0,9438	0,9424	0,9578	**0,9143**	0,9583	0,9532	**0,9643**	**0,94484**
C7	**0,9241**	0,9545	0,9514	0,9578	0,9542	**0,9643**	0,9411	0,9613	0,9496	0,9500	**0,95083**
C8	**0,9261**	0,9532	0,9537	0,9542	0,9478	**0,9578**	0,9487	0,9491	0,9433	0,9482	0,94821
C9	0,9318	0,9562	0,9478	0,9376	0,9416	**0,9661**	**0,9218**	0,9619	0,9514	0,9523	0,94685
Max.	**0,9330**	0,9567	0,9655	0,9578	0,9542	0,9661	0,9487	0,9637	0,9595	**0,9703**	0,95083
Min.	**0,9125**	0,9369	0,9474	0,9376	0,9380	**0,9500**	0,9143	0,9491	0,9279	0,9394	0,94484
$\sigma(x)$	0,0065	0,0068	0,0068	0,0065	0,0052	0,0053	**0,0111**	**0,0043**	0,0094	0,0094	0,0018

Tabela 19 – O desempenho da métrica F1 dos classificadores utilizados nos DSS para cada um dos conjuntos de teste. Os valores máximos e mínimos foram destacados

Naturalmente que a selecção aleatória conduziu à utilização de classificadores que estão longe de serem os melhores (obtidos através da combinação óptima dos parâmetros de indução).

A análise dos dados da Tabela 19 permite verificar uma variação do desempenho considerável dos classificadores nos diversos conjuntos de teste, que apresenta um desvio-padrão máximo para o C2 de 0,0167. O melhor resultado absoluto foi obtido por C2 em T10 e o pior por C3 em T1. Os classificadores apresentaram os piores desempenhos para os conjuntos de teste T1 e T7, tendo os melhores ficado distribuídos por T3, T6, T8 e T10, o que confirma a existência de conjuntos «mais difíceis» e «mais fáceis» e a necessidade de técnicas de generalização de erro para obtenção de resultados fiáveis.

A Tabela 20 apresenta os resultados obtidos utilizando os Métodos: Fajé e Maioria, com um número crescente dos classificadores intermédios. A apresentação somente do número impar de classificadores deve-se à necessidade de um número impar de classificadores para o método da Maioria (apesar do Método Fajé permitir a utilização de um número par).

	Método Fajé						Regra da Maioria					
	Prec.	$\sigma(x)$	Rech.	$\sigma(x)$	F1	$\sigma(x)$	Prec	$\sigma(x)$	Rech.	$\sigma(x)$	F1	$\sigma(x)$
1	0,9308	0,0310	0,9768	0,0074	**0,9530**	0,0165	0,9308	0,0310	0,9768	0,0074	**0,9530**	0,0165
3	0,9375	0,0261	0,9761	0,0104	0,9562	0,0143	0,9376	0,0268	0,9775	0,0072	**0,9570**	0,0148
5	0,9354	0,0288	0,9772	0,0078	0,9556	0,0155	0,9329	0,0263	0,9782	0,0079	0,9549	0,0152
7	0,9411	0,0201	0,9747	0,0086	**0,9575**	0,0108	0,9321	0,0240	0,9782	0,0081	0,9544	0,0134
9	0,9357	0,0209	0,9754	0,0079	0,9550	0,0118	0,9313	0,0238	0,9789	0,0079	0,9544	0,0137

Tabela 20 – Resultados obtidos com PTD baseados na regra da maioria, e com o método Fajé utilizando um número de classificadores crescente para a classificação das vendas

Os resultados são interessantes tendo em consideração que os SAD construídos apresentam melhores resultados do que os classificadores intermédios utilizados. Todos os resultados obtidos para F1 são iguais ou superiores a 0,953, valor que não é atingido por nenhum classificador intermédio. A análise dos valor obtidos por cada classificador para cada conjunto de teste (sem a média), permite verificar que apenas por 35 por cento são superiores a 0,953, e que o valor médio é somente de 0,9473.

Desta forma, é possível concluir que existe uma vantagem na combinação de diversos classificadores, pois asseguram um melhor comportamento, ao que acresce o facto de serem mais estáveis, e logo menos sensíveis a variações no conjunto de dados.

Comparativamente, os desempenhos apresentados pelos dois métodos são muito semelhantes, podendo ser caracterizados como equivalentes.

A principal vantagem pertence contudo ao método Fajé e está relacionada com a eficiência, uma vez que potencialmente são necessários menos classificadores intermédios do que acontece com o método da maioria. No caso da maioria, é necessário encontrar $N / 2 + 1$ estimativas iguais para se poder atribuir a classificação final. No caso do método Fajé são utilizados somente os classificadores intermédios estritamente necessários para atingir uma folha do classificador final.

5.5 Definição das regras do SAD do Explorador

A definição do SAD para os exploradores passa pela operacionalização das regras que permitem **a identificação e extracção dos conceitos**, no caso concreto, a identificação de produtos e dos seus atributos, assim como **do conjunto de palavras-chave associadas** que permite a sua posterior classificação. No âmbito do estudo de caso optou-se por extrair, exclusivamente, a seguinte informação sobre produto: referência, descrição, preço, URL. As regras são operacionalizadas com o auxílio do agente Tutor e, finalmente, transferidas por mensagem aos Exploradores, condicionando o seu comportamento.

5.5.1 As regras de extracção de conceitos

Tal como foi descrito, o Tutor possui dois tipos de regras de extracção de conceitos: *i)* extracção de conceitos de tabelas; e *ii)* extracção de conceitos em folhas de texto.

Para a operacionalização da regra da **extracção de conceitos de tabelas** assume-se que a cada linha corresponde um produto, e a cada coluna um atributo de produto, tal como se pode observar na Figura 76, sendo a regra activada sempre que o Explorador encontra uma tabela no documento em análise. Utilizando esta regra, o Explorador assume a apresentação de conceitos relevantes sempre que identifica uma tabela com as características descritas, atribuindo o significado de um atributo de produto (em função do

cabeçalho de coluna), o que permite não só identificar produtos como, igualmente, separar a informação existente.

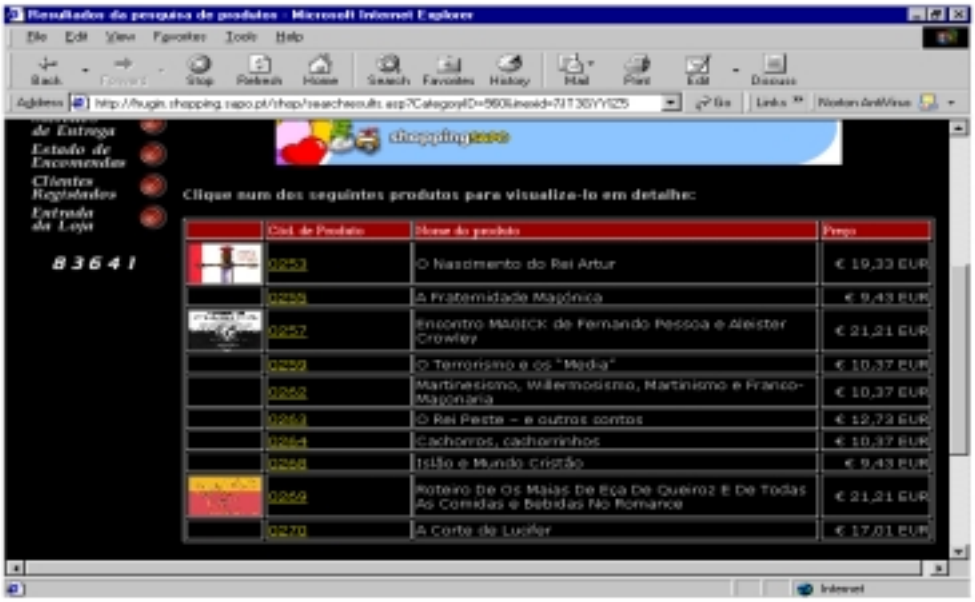


Figura 76 – Documento Web que apresenta informação de venda de produtos (livros)

A operacionalização da regra passa pela definição do **conjunto de palavras de cabeçalho** que permite associar significado semântico a cada coluna, de modo a identificar os diferentes atributos dos conceitos. O conjunto de palavras de cabeçalho, apresentado na Tabela 21, **foi criado utilizando as palavras mais comuns encontradas a descrever cada um dos atributos nos cabeçalhos**.

Atributo	Palavras
Código	Cod. Ref. Code Reference
Descrição	Description; Info; Product
Preço	Cost, Price, €, $, £
URL	URL

Tabela 21 – Conjunto de palavras que permite associar significado semântico às colunas de tabelas

No caso de ter sido possível atribuir significado semântico aos atributos, a extracção dos produtos é efectuada por iteração nas linhas, assumindo que em cada linha existe um produto que é descrito nas sucessivas colunas.

Para a operacionalização da regra de **extracção de conceitos em folhas de texto,** que tira partido da assunção de que diferentes atributos de conceito estão envolvidos por diferentes marcas HTML, é necessário definir as regras**: *i)* de filtragem de vectores; e *ii)* de definição semântica de conteúdo.**

A regra de filtragem de vectores é definida pela relação entre a descrição do produto e o seu preço, isto é, para cada descrição de produto existe um e só um preço referente a essa

mesma descrição. Logo, apenas serão interessantes os vectores que contenham um item referente ao preço do produto. Esse item terá de ter um valor numérico e um símbolo monetário para ser classificado como preço. A aplicação desta regra reduz drasticamente o número de vectores de folhas interessantes.

A regra de definição semântica de conteúdo visa permitir a extracção de informação do vector. A sua operacionalização obrigou às seguintes definições:

i) **Para o preço** do produto procura-se uma folha que contenha um valor numérico e um símbolo monetário, folha esta que será única, devido ao pré-processamento;

ii) Para a **identificação da descrição do produto** considera-se a folha com mais caracteres existente no vector, isto porque a maioria dos sítios comerciais tenta dar um pequeno resumo das características do produto na sua descrição. Por exemplo: 'Gangs de Nova Iorque – Ed. Especial DVD', 'Máquina de Secar Roupa Becken Compact Dryer' ou 'nanoChip WEEK PROMO AMD Athlon XP 2400+';

iii) Para os restantes atributos utilizam-se, uma vez mais, as palavras-chave definidas na Tabela 21.

Este método é muito eficaz conseguindo extrair com sucesso todos os produtos existentes em páginas comerciais, onde se usa tecnologia de páginas dinâmicas para construção de sítios. Outra vantagem deste método reside no facto de, também, conseguir extrair com sucesso os produtos que eram extraídos pelo método das tabelas, dado que cada linha de tabela faz parte de um nó html «tr», separando assim os produtos em diferentes vectores.

5.5.2 As regras para extracção de palavras-chave

Após o reconhecimento de um conceito é necessário compor o conjunto de palavras-chave, palavra-chave composta, que permite a sua classificação por parte do agente Catalogador. A palavra-chave composta é criada tendo em consideração: *i)* um subconjunto de atributos recolhidos no processo de identificação de conceitos; *ii)* informação resultante da interpretação do elo de localização da página URL.

No caso em estudo a primeira definição foi simples e foram **utilizados todos os atributos de conceito**: a referência de produto, descrição, o preço e o seu URL de apresentação**.**

Para a **interpretação dos elos** foi necessário analisar cada um dos sítios e construir as regras que permitem a sua interpretação, de modo a extrair a informação armazenada. Este tipo de regras está dependente dos sítios em análise, sendo dificilmente aplicável a casos desconhecidos todavia, tem a vantagem de conduzir a desempenhos muito elevados.

Apresenta-se em seguida um exemplo ilustrativo. Considere-se o endereço de página http://ess.shopping.sapo.pt/shop/searchresults.asp?CategoryID=2536. A análise permite identificar que o sítio Internet é o SAPO (.sapo.pt) e que está a apresentar o(s) produto(s) da categoria 2536 (CategoryID=2536). Esta constatação permite criar a seguinte regra: na

presença do sítio Sapo, verificável pela existência de «sapo.pt», procurar a palavra «CategoryID=» e incluir os dígitos seguintes e a palavra «sapo.pt» na palavra-chave composta.

5.5.3 A análise do desempenho

A análise do *corpus* permitiu concluir, que nos documentos da categoria de vendas, 36 por cento dos produtos são apresentados em tabelas, 30 por cento são detalhes dos documentos previamente apresentados nas tabelas, (logo não são relevantes) e que os restantes apresentam outras variações (e.g. imagens).

Neste contexto, a regra de extracção de conceitos de tabela permitiu capturar o equivalente a 66 por cento dos casos, e a regra da extracção de conceitos em folhas de texto 19 por cento, o que totaliza uma taxa de sucesso igual a 85 por cento. A restante informação não foi capturada, resultando numa perda para o utilizador.

O aumento da taxa de sucesso passa pela introdução de novas regras que permitam descrever as páginas que não são capturadas pelo actual sistema.

5.6 Personalização da ontologia para o Catalogador

A personalização da ontologia para o catalogador consiste na atribuição das palavras habitualmente associadas a cada produto seus atributos, aos respectivos conceitos que os representam na ontologia. Desta forma, e após uma análise dos dados em estudo, foi possível enriquecer a ontologia com o conjunto de palavras que melhor descreve os produtos em análise.

Para além das palavras que descrevem os conceitos, foi necessário incluir as palavras extraídas através da regras de interpretação dos elos. Só desta forma é possível tirar partido das palavras-chave criadas, (no exemplo «2536sapo.pt»), permitindo ao Catalogador a identificação inequívoca dos conceitos.

5.7 Análise crítica sobre o estudo de caso

O processo de recuperação de informação utilizou técnicas de aprendizagem em texto e atingiu taxas de desempenho extremamente elevadas, possibilitando a classificação de documentos em duas classes: venda e normais.

No que se refere ao processo de extracção de informação, as técnicas utilizadas estão relacionadas com regras «se-então» e palavras-chave, que permitiram descrever o processo de apresentação da informação.

A recuperação de informação é assim um processo automático, não cabendo ao utilizador fornecer informação, o que não acontece com o processo de extracção de informação.

A extracção de informação é um processo incremental, e interactivo, tendo em conta que os conceitos não reconhecidos são enviados para o classificador manual conduzindo ao seu futuro reconhecimento.

O resultado conjunto revelou-se eficaz na medida em que permite a recuperação e extracção da maioria da informação.

6 Conclusões e perspectivas

Uma solução definitiva para o problema abordado no decurso desta dissertação está longe de ter sido encontrada. A dimensão do problema, a multiplicidade de métodos e abordagens possíveis, e a interdependência de diversas áreas de conhecimento, asseguram que esta área de investigação estará fortemente activa nos anos vindouros. Espera-se, contudo, que os trabalhos efectuados tenham contribuído para encurtar o longo caminho a percorrer.

De acordo com o enunciado ao longo desta dissertação, o objecto central focalizou-se na construção de uma abordagem global para selecção e catalogação da informação armazenada na Web, contribuindo para a solução de um problema crescente: a inexistência de processos ágeis que permitam explorar o actual potencial de informação, que está oculto na dimensão esmagadora e na falta de estruturação intrínsecas à Web. A abordagem apresentada nesta dissertação propõe uma metodologia geral, uma arquitectura de referência e uma metodologia específica de suporte à derivação de sistemas particulares, que foram materializados num protótipo de investigação. Os resultados obtidos foram animadores, permitindo confirmar as expectativas iniciais e lançar os fundamentos de novas áreas de trabalho.

Os estudos inicialmente realizados, com vista ao levantamento do estado da arte, revelaram-se determinantes para a correcta definição das metodologias e da arquitectura proposta, assim como, para a selecção dos métodos e das ferramentas aplicadas. O recurso ao paradigma de multiagentes e a selecção de uma plataforma de desenvolvimento estável foram o ponto de partida que alicerçou os trabalhos efectuados.

Os resultados obtidos, no estudo de caso particular dedicado à criação de um catálogo de produtos, são prometedores, apesar de carecerem de uma validação posterior em contextos distintos, o que esteve fora do âmbito desta dissertação.

Este capítulo está dividido nas seguintes secções: *i)* análise das propostas efectuadas; *ii)* visão crítica e futuras áreas de trabalho; e *iii)* projectos futuros e ensino.

6.1 Análise das propostas efectuadas

No que respeita à **metodologia geral**, de uma forma sintética e genérica, pode-se afirmar que o trabalho apresentado inova ao propor uma solução centrada no utilizador, contrariando as iniciativas à data, fortemente centradas na personalização dos dados no fornecedor. A solução defendida assenta numa visão global e integrada de abordagem aos diversos problemas, desde a interface do utilizador até aos processos de recolha e catalogação de texto. Este processo geral e integrador permitiu assegurar a interoperabilidade dos diversos componentes, a existência de uma interface humano-máquina coerente e a divisão de tarefas de forma eficaz e eficiente. A aproximação holística seguida foi possível, devido, à experiência acumulada de desenvolvimentos aplicacionais envolvendo processos de integração e normalização de dados, e revelou-se essencial para a obtenção de uma solução harmoniosa.

No que respeita à **arquitectura de referência**, a utilização de uma aproximação suportada num sistema de multiagentes revelou-se determinante para ultrapassar as dificuldades impostas pela natureza do problema abordado. O paradigma de multiagentes foi aplicado como metodologia de desenho e como tecnologia de implementação. A utilização dos agentes permitiu obter uma solução de elevada modularidade, descentralizada e adaptável, que permitiu ultrapassar com naturalidade a pouca estruturação inicial do problema. A arquitectura proposta permite uma fácil adaptação a casos de estudo concretos, permitindo a alteração do número de agentes por tipo, conforme as necessidades específicas.

Os agentes colaboram para resolver problemas complexos, não possuindo uma consciência do objectivo global. A inteligência/racionalidade do sistema deixa assim, de ser propriedade de um único indivíduo, sendo transferida para a comunidade, ao contrário do que acontece nas tradicionais soluções distribuídas.

Como desvantagem, o sistema de agentes é difícil de controlar e nem sempre converge. É um paradigma recente, não existindo ainda uma teoria de suporte consistente às fases de análise e de implementação. Esta lacuna dificulta a obtenção de uma programação de elevada qualidade, livre de erros, segura e com tolerância a falhas. Acresce que as ferramentas disponíveis apresentam, naturalmente, os mesmos problemas, estando longe de serem fiáveis, eficientes, seguras e escaláveis.

Todavia, a arquitectura proposta revelou-se adequada às necessidades concretas do projecto de investigação (o estudo de caso), tendo sido utilizada por equipas de diversas nacionalidades.

A adopção de ontologias para a representação do conhecimento na arquitectura permitiu a criação de uma norma de dados utilizada de forma abrangente ao nível dos agentes, do catálogo e como interface com o utilizador. Esta opção revelou-se extremamente vantajosa pois evita a existência de mapas de conversões de conceitos.

A **metodologia de derivação de sistemas particulares** assenta na selecção, adopção e enriquecimento da ontologia de domínio e na definição dos sistemas de apoio à decisão (SAD) dos agentes.

O processo de definição da ontologia de domínio consiste na pesquisa e selecção da melhor proposta disponível. O problema mais comum será, sem dúvida, a inexistência de normas aceites que simplifiquem a tarefa do utilizador, o que pode obrigar, inclusive, à definição de raiz de uma ontologia específica. Todavia, a maior dificuldade está relacionada com a morosidade do processo manual de enriquecimento da ontologia, que permitir a sua utilização na classificação dos conceitos.

Os SAD utilizados são de duas naturezas e aplicados em objectivos distintos: *i)* aprendizagem supervisionada em texto para a recuperação de informação, (agente Navegador); *ii)* regras de «se-então» para a extracção de informação (agente Explorador); e *iii)* o enriquecimento da ontologia (agente Catalogador).

As aproximações adoptadas revelaram desempenhos extremamente satisfatórios, permitindo uma boa catalogação da informação. Todavia, apesar dos algoritmos de aprendizagem permitirem auxiliar a particularização da arquitectura, evitando alguns processos desmotivadores associados à selecção de opções, o processo de automatização está ainda longe de se encontrar finalizado obrigando, ainda, a uma forte intervenção de especialistas. O objectivo de libertar o utilizador das tarefas de particularização não foi, assim, totalmente atingido com a agravante das tarefas não automáticas serem morosas, repetitivas e exigirem competências técnicas específicas.

A derivação do SAD do agente Navegador é obtida pela realização das seguintes etapas: *i)* definição do *corpus*; *ii)* selecção de características; *iii)* indução de classificadores; e *iv)* definição de SAD. Na execução destas tarefas utilizaram-se diversos métodos conhecidos, todavia foram **propostas contribuições originais em diversos domínios, especificamente:**

i) uma metodologia para criação do *corpus* que permite aumentar o grau de confiança nos resultados obtidos;

ii) uma definição de um processo de optimização da selecção de características através da IMC, como método de eliminação de variáveis correlacionadas, com a utilização de algoritmos genéticos;

iii) uma alteração ao método C4.5, o C4.5 iterado, que permitiu a geração de árvores de decisão mais compactas;

iv) um método de construção de SAD, o método Fajé, criado com o objectivo de explorar estimativas de classificadores em minoria, que são ignorados pela regra da maioria.

Os trabalhos efectuados nesta área foram detalhadamente analisados nos respectivos capítulos, todavia, é possível efectuar um conjunto de considerações gerais resultantes da reflexão realizada. Em todas as etapas ficou patente: *i)* a extrema importância de uma validação empírica; *ii)* a dificuldade de aprendizagem sem uma assumpção de conhecimento *a priori*; *iii)* a surpreendente utilidade de métodos com representação limitada.

Foi possível verificar, igualmente, a inexistência de métodos de aprendizagem de âmbito genérico; a utilidade de cada método está relacionada com as premissas assumidas e cada aplicação obriga a uma atenção individual. Mesmo regras ditas «universais», tais como a conhecida «Occam's razor», que postula que «hipóteses mais simples, apresentam melhor desempenho», devem ser verificadas com atenção em cada novo caso.

Verificou-se que a existência de desvios indutivos diferentes, para cada método de indução, corrobora os esforços de combinação de métodos de distinta natureza que possam fazer a compensação automática necessária para a obtenção de sucesso, em casos práticos.

Ficou claro que o efeito da pesquisa (normalmente massiva) de hipóteses com significado, é uma questão central relacionada com a validação estatística e a computação eficiente. A elevada dimensionalidade do espaço de pesquisa, torna não negligenciável a possibilidade das hipóteses inferidas serem um simples resultado do acaso. É essencial a existência de uma metodologia de validação, que permita a generalização das estimativas com vista à legitimação dos resultados obtidos.

Finalmente, ficou patente que a análise de espaços de pesquisa de elevada dimensionalidade exige uma complexidade computacional, tanto quanto possível linear, o que não acontece na maioria dos métodos, especialmente na fase de indução. É essencial a adaptação dos métodos com vista à obtenção desta característica, em especial no caso de serem utilizados em bancos de dados, o que pode ser realizado por optimização algorítmica, sem alteração dos resultados ou, pela flexibilização de requisitos, o que torna inevitável a alteração dos resultados obtidos, assim como o aumento da instabilidade final.

Para a **derivação do SAD do agente Explorador** não foram identificadas técnicas de aprendizagem automática, devido às limitações de criação de um *corpus* de documentos representativo. A solução adoptada, de construção de regras «se-então», proporcionou a obtenção de um desempenho satisfatório com um conjunto reduzido de regras. Esta opção permitiu constatar que a maioria dos sítios Internet adopta uma estrutura-base de

apresentação muito semelhante, possibilitando a sua análise através de um pequeno conjunto de regras genéricas.

Para a **derivação do SAD do agente Catalogador** recorreu-se à utilização de palavras-chave na identificação dos conceitos. As palavras são adicionadas uma única vez, sendo, então, utilizadas nos reconhecimentos posteriores. As palavras-chave são determinadas por análise das páginas e por análise dos endereços das páginas. O facto da maioria dos sítios analisados serem gerados dinamicamente, leva a que os endereços contenham informações preciosas sobre os conteúdos apresentados. Esta informação foi capturada e permitiu a identificação correcta dos conceitos analisados.

Genericamente, os resultados obtidos comprovaram a adequação dos métodos utilizados e propostos, no contexto da classificação de documentos de venda, permitindo a obtenção de uma solução inovadora para a personalização do comportamento dos agentes de navegação.

Finalmente, o último componente, o desenvolvimento do protótipo, consiste, essencialmente, num trabalho de engenharia. O desenvolvimento foi realizado tendo em conta duas vertentes: a implementação do sistema com os seus diversos módulos e agentes e a implementação dos mecanismos de aprendizagem. Esta tarefa obrigou a um esforço de integração de diversos componentes aplicacionais (desenvolvidos utilizando tecnologias distintas), o que só foi conseguido devido a uma abordagem sistemática de isolamento de componentes e a uma política de normalização das interfaces e dos formatos de dados.

A escolha das ferramentas de desenvolvimento, em especial a linguagem de programação e a plataforma de agentes, contribuiu para o sucesso da construção do protótipo. A linguagem JAVA, aliada ao seu ambiente de desenvolvimento, permitiu a obtenção de um sistema robusto e fiável. O tradicional defeito de falta de eficiência, por ser interpretada, é mascarado pela elevada capacidade de processamento dos computadores e pelos tempos de latência impostos pela Internet.

A plataforma JADE, à semelhança de outras, apresenta algumas limitações, associadas, essencialmente, à segurança e escalabilidade, uma das razões muitas vezes apontada como responsável pelas reduzidas aplicações dos agentes. Todavia, apesar das referidas limitações, a plataforma contribuiu, igualmente, para os resultados obtidos pelo conjunto de funcionalidades oferecidas, pela flexibilidade apresentada e pela organização induzida pelas classes dos pacotes, especialmente vocacionada para o desenvolvimento de agentes, tendo permitido a identificação de soluções elegantes, revelando um grau de maturidade suficiente para garantir a estabilidade e fiabilidade necessárias ao desenvolvimento de um protótipo.

Quanto aos sistemas operativos e servidores utilizados, o desempenho obtido foi o esperado tendo em consideração a sua vastíssima aplicação e provas dadas.

O desenvolvimento realizado sobre a Internet, com o objectivo de testar um sistema de comunicação «instalável» entre os diversos componentes, (agentes ou não) provou que a aproximação seguida permite a obtenção de uma solução robusta.

É necessário contudo, manter presente, que um protótipo não é um sistema comercial, não apresentando semelhante robustez, fiabilidade e/ou escalabilidade, o que limita o seu uso imediato numa aplicação real. Apesar de todo o cuidado investido, o protótipo sofre de limitações inerentes a um desenvolvimento em laboratório, com o objectivo primordial de servir de base à validação de propostas: elevado número de simplificações, insuficiente tratamento de casos excepcionais, entre outras. A tudo isto, adicionando-se a imaturidade do paradigma de agentes e as insuficiências das plataformas disponíveis para implementação de sistemas de agentes.

Todavia, é possível afirmar que o protótipo constituiu uma ferramenta determinante no aperfeiçoamento da arquitectura, dos algoritmos e dos protocolos desenvolvidos e aplicados, permitindo demonstrar a sua aplicabilidade.

Relembra-se, para concluir, que o protótipo foi utilizado no projecto de investigação europeu, o DEEPSIA, e permitiu a realização de simulações com dados reais, revelando-se a ferramenta adequada. O demonstrador final foi instalado em São Paulo, Lisboa, Madrid, Bruxelas e Cracóvia o que é, por si só, revelador do grau de robustez atingido.

6.2 Visão crítica e futuras áreas de trabalho

O trabalho apresentado, mais do que relatar a execução de uma proposta, funciona como ponto de partida para novos desafios. Existe um espaço de progressão real em todas as dimensões abordadas, pelo que a sua exposição é sempre um exercício incompleto, tendo-se optado por abordar os grandes temas.

Como perspectivas de trabalho futuro, duas linhas emergem como continuação natural das propostas apresentadas nesta dissertação: *i)* a verificação da aplicabilidade das propostas noutros contextos de trabalho; e ii).a consolidação e investigação das propostas efectuadas.

A principal limitação do trabalho é a falta de comprovação de adequabilidade das propostas realizadas a contextos fora da criação de catálogos personalizados para produtos. Apesar deste trabalho não ter sido realizado, por se considerar fora do âmbito da dissertação, faz parte dos temas a desenvolver no futuro. A definição de novos contextos e a criação de novos *corpus,* permitirão, então, desencadear um novo conjunto de experiências e a validação da aplicabilidade das propostas a novos desafios.

A inversão de paradigma, proposto na **metodologia geral**, transporta a responsabilidade de construção dos catálogos para o utilizador, aumentando a capacidade de personalização e de contextualização temática. Todavia, a transferência de responsabilidade é, sem dúvida,

um dos problemas desta abordagem, devido ao crescimento da complexidade dos processos. Esta é uma área em que as expectativas de trabalho futuro são mais aliciantes, como seja tentar ultrapassar a complexidade das tarefas permitindo uma mais fácil adopção da metodologia proposta. Futuramente, os desenvolvimentos estarão centrados na derivação de sistemas particulares com menor intervenção do utilizador, aumentando a autonomia e explorando a migração do agente Tutor para plataformas externas. Este desiderato permitirá uma interacção remota, baseada na troca de mensagens, em que os utilizadores limitarão a sua intervenção ao fornecimento de dados, que são analisados e, posteriormente, traduzidos em novos SAD, alterando, em conformidade, o comportamento dos agentes. A migração do Tutor, ou a simples autonomização, obriga à identificação de soluções que permitam: *i)* o desencadear do processo de geração de SAD de forma autónoma; *ii)* a criação de *corpus* representativos tendo por base um conjunto limitado de exemplos; *iii)* ultrapassar limitações dos algoritmos que obrigam à intervenção de especialistas; e *iv)* a criação de um mecanismo automático de inferência de regras para a identificação e catalogação de conceitos. Esta evolução permitirá diminuir a necessidade de especialização dos utilizadores ao mínimo, contribuindo para a facilidade de operação do sistema proposto, numa «filosofia» de operação muito semelhante à utilizada pelos fornecedores de aplicações «anti-virus».

Relativamente à **arquitectura de referência** é possível identificar um conjunto de desafios que também lançam novas áreas de trabalho.

No que se refere ao sistema de multiagentes, existem diversas temáticas que podem ser abordadas com vista a melhoria do desempenho global do sistema.

A mobilidade dos agentes, i. e., a capacidade de navegação entre plataformas através das redes de comunicação, permitiria a recolha e catalogação de informação perto da fonte, diminuindo, assim, o tráfego de rede. É necessário contudo manter presente que associados à mobilidade estão os problemas de segurança e de administração, tais como a existência de superpopulações e servidores e/ou agentes maliciosos, o que obriga à definição de protocolos específicos que assegurem garantias, (entre outras de acesso e de terminação).

A criação de comportamentos de negociação de conhecimento que permitam a partilha de experiências e um enriquecimento mais rápido (o conhecimento seria cedido em troca de benefícios individuais ou de grupo) é outra área muito interessante. Este assunto podia ser estudado à luz de agentes colaborativos, e/ou competitivos, o que permitiria a construção de cenários em que o sistema evoluiria de forma independente e, por iniciativa dos proprietários, poderia iniciar processos de partilha de resultados.

Entre outras questões centrais, realça-se, ainda, a optimização da comunicação entre agentes, a validação de crenças, e a gestão de cargas que permita a administração dinâmica do número de agentes activos.

Todavia, é necessário ter em conta que, quanto maior for o grau de sofisticação das soluções identificadas, mais complexa é a tarefa de coordenação entre os diversos agentes, e consequentemente, maior o grau de dificuldade.

No que se refere a casos específicos, é possível indicar:

- **a validação dos dados armazenados:** as referências na Internet são difíceis de actualizar, uma vez que estão sempre a ser alteradas, neste sentido, espera-se utilizar uma estratégia de verificação constante dos dados recolhidos;
- **a evolução da interface de acesso às ontologias** permitindo a sua utilização mais intuitiva, evitando assim problemas importantes relacionados com a existência de sinónimos, conceitos que podem ter múltiplas identificações, abreviações, etc.;
- **a exploração de conhecimento** sobre a base de dados do catálogo. A periodicidade da pesquisa do Navegador permite produzir uma fonte de informação que pode ser explorada, no caso da aplicação concreta, identificação de produtos com preços reduzidos, usualmente, associados a promoções;
- **a criação de mecanismos de auxílio directo ao Navegador** permitindo aumentar a sua eficácia (e. g., quando a dúvida sobre a classificação a atribuir a uma observação for muito elevada).

No que se refere à **metodologia de derivação de sistemas particulares,** é possível identificar diversas áreas de progresso. Uma área especifica relaciona-se com a aprendizagem em texto, onde se destaca:

- a utilização de técnicas de compactação de características;
- a utilização de documentos não classificados, com recurso à classe de algoritmos EM, com vista a melhorar os classificadores sem ser necessário passar pelo «penoso» processo de classificação individual de cada observação;
- a utilização de PCA (Principal Component Analysis) para a selecção de características;
- a utilização de novos tipos de classificadores;
- a construção de SAD hierárquicos com graus de especialização crescentes, o que permitiria o encadeamento de classificadores generalistas com classificadores de especialidade crescente;
- o desencadear do processo de geração de novos SAD em resposta à avaliação real do desempenho dos agentes, realizado por realimentação;

- o estudo intensivo do algoritmo Fajé, aplicado a novos casos de estudo, que permitam validar a sua utilidade em novos contextos.

Finalmente, **relativamente ao protótipo,** para além de um vasto conjunto de melhorias de fiabilidade, não foi implementado o sistema de realimentação que permitiria o desencadear de processos automáticos de melhoria de desempenho global do sistema. A arquitectura de referência prevê que o Tutor possa realizar a avaliação do desempenho real dos agentes, tendo em conta que o agente interface de catálogo e o agente Catalogador estão regularmente em contacto com o Tutor.

6.3 Projectos futuros e ensino

Os resultados obtidos estão contextualizados ao estudo de caso que, apesar de ter utilizado dados não simulados, recolhidos de sítios Internet comerciais, não garante a sua extrapolação genérica. Todavia, apesar desta limitação, e após a realização de testes prospectivos noutras áreas, os resultados obtidos permitiram o lançamento de novos desafios tendo por base as propostas efectuadas nesta dissertação. Estes desafios concretizaram-se em novas propostas de projectos de investigação e desenvolvimento que estão, actualmente, em fase de execução ou aguardam, ainda, o processo de avaliação.

O primeiro grande teste, noutro contexto de trabalho está a ser realizado na área financeira, através do projecto CadEmIn (Cadastro de Empresas e Indivíduos). Este projecto tem como objectivo a construção de cadastros de empresas e indivíduos obtidos pelo processamento da informação disponível na Web. O enquadramento proposto está a ser integralmente utilizado e espera-se, no final deste projecto, atingir a fase de pré-produto permitindo, em paralelo, a consolidação dos métodos e processos aplicados nesta nova área.

Em simultâneo, estão em curso novos projectos de investigação aplicada que fazem uso parcial das aproximações propostas permitindo a sua validação.

O projecto à data mais avançado, iniciado em Setembro de 2003, está a ser realizado para a Marinha de Guerra Portuguesa e visa a obtenção de mecanismos de escrita distribuída e recuperação de falhas por negociação efectuada por agentes inteligentes. O objectivo é permitir ao SINGRAR (Sistema Integrado para a Gestão de Prioridades de Reparação e Afectação de Recursos) a criação de um sistema pericial distribuído, destinado a ser utilizado em navios militares, que integra diversas funcionalidades de apoio à decisão e também o aconselhamento relativamente à Gestão de Prioridades de Reparação dos equipamentos de bordo. Este projecto, pelo grau de exigência imposto e pela sua aplicação na área militar, permitirá consolidar conceptualmente, e de forma aplicada, a utilização de agentes e os processos de sincronização utilizados.

Igualmente aprovado, com data de início prevista para o primeiro semestre de 2005, encontra-se o projecto AURORA KES, financiado pela ESA (Agência Espacial Europeia). O objectivo é a utilização de um sistema de multiagentes para a gestão da carga, monitorização e diagnóstico do instrumento «Pasteur». A validação dos conceitos será realizada com dados reais provenientes da sonda espacial Beagle 2. Uma vez mais, este projecto permitirá a validação dos conceitos, especificamente a validação dos processos de selecção de características e de construção dos SAD, e a consolidação da abordagem suportada em agentes.

Espera o autor, com a realização destes projectos, terminar a fase de consubstanciação das propostas e conceitos apresentados e defendidos ao longo desta dissertação, criando alicerces sólidos para futuros trabalhos de investigação.

Ao longo da execução dos trabalhos foi efectuado um total de 12 publicações em Conferências Internacionais, um artigo de profundidade numa revista Internacional (a aguardar publicação) e foram gerados 22 relatórios técnico-científicos.

Os trabalhos efectuados contribuíram, igualmente, para fins educativos, tendo estado na origem de 4 trabalhos de fim de curso (terminados), e duas teses de Mestrado (actualmente em fase de conclusão) na Universidade Nova de Lisboa.

Em paralelo, foi possível construir uma fortíssima relação com a Universidade de São Paulo no Brasil que permitiu reforçar o tema da recuperação e extracção de informação no seu programa de pós-graduação (Dissertações de Mestrado), tendo estado na origem de seis trabalhos de mestrado, e seis publicações internacionais (ver anexo A.9).

Mais do que uma tarefa terminada, espera-se que os trabalhos apresentados sirvam de incentivo à criação de novos desafios e contribuam para o fortalecimento do grupo de investigação em que foram desenvolvidos.

Referências bibliográficas

[1] Chaudry, M.A., *Web surpasses one billion documents*, acesso em 2002/10/19, *http://lists.isb.sdnpk.org/pipermail/cyberclub-old/2000-February/001299.html*, 2000.

[2] Reach, G., *Global Internet Statistics: Source & References*, acesso em 2004/02/21, Global Reach, *http://glogal-reach.biz/globstats/refs.php3*, 2003.

[3] Consortium, I.S., *ISC DOmain Survey: Number of Internet Hosts*, acesso em 2004/02/21, *http://www.isc.org/index.pl?/ops/ds/host-count-history.php*, 2004.

[4] Reach, G., *Global Internet Statistics (by Language)*, acesso em 2004/02/21, Global Reach, *http://www.glreach.com/globstats/index.php3*, 2003.

[5] Berners-Lee, T., et al., *The World Wide Web,* Communications of the ACM, Vol. 37 (8), pp. 76-82, 1994.

[6] Bednarek, M., *An Introduction to HTML*, acesso em 2004/07/01, *http://tech.irt.org/articles/js095*, 1998.

[7] Press, T.a., *Google, Yahoo! Revving Up Search Engines*, acesso em 2004/03/08, abcNews, *http://abcnews.go.com/wire/Business/ap20040218_482.html*, 2004.

[8] Morrissey, B., *Search Guiding More Web Activity*, acesso em 2003/11/08, Internet Advertising Report, *http://www.internetnews.com/IAR/article.php/2108921*, 2003.

[9] Sherman, C., *Yahoo! Birth of a New Machine*, acesso em 2004/03/08, Jupitermedia Corporation, *http://searchenginewatch.com/searchday/article.php/3314171*, 2004.

[10] Looksmart, *Looksmart UK Help*, acesso em 2004/03/08, Looksmart, *http://www.looksmart.co.uk/help/index.jsp?pName=6*, 2004.

[11] Seo, *Top tem search engines*, acesso em 2004-03-13, SEO consultantes Directory, *http://www.seoconsultants.com/search-engines/*, 2004.

[12] Lawrence, S. and C.L. Giles, *Context and Page Analysis for Improved Web Search,* IEEE Internet Computing, Vol. 2 (4), pp. 38-46, 1998.

[13] Notess, G.R., *Internet Onesearch With the Mega Search Engines,* Online, Vol. 20 (6), pp. 36-39, 1996.

[14] Yankeegroup, *Mass Consolidation hits te Web-Search Market*, acesso em 2003/11/8, Yankeegroup, *http://tewhir.com/marketwatch/yan052003.cfm*, 2003.
[15] Chang, G., et al., *Mining the World Wide Web*, Information retrieval, editora Kluwer Academic Publishers, ISBN 0-7923-7349-9, numero de pág. 168, 2001.
[16] Cooley, R., B. Mobasher, and J. Srivastava, *Web Mining: Information and Pattern Discovery on the World Wide Web, 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Newport Beach, CADepartment of Computer Science University of Minnesota, ISBN 0-8186-8203-5, pp. 608, 03–08 November 1997, 1997.
[17] Simon, H., *Why should machines learn?*, in *Machine learning: An artificial intelligence approach*, J.G.C. R. S. Michalski, and T. M. Mitchell (Eds),Tioga, Palo Alto, CA, ed, J.G.C. R. S. Michalski, and T. M. Mitchell (Eds),Tioga, Palo Alto, CAa Morgan Kaufmann, ISBN 0935382054, pp. 25-38, 1983.
[18] Langley, P., *Elements of Machine Learning*, editora Morgan Kaufmann, 1996.
[19] Michalski, R., *A theory and methodology of inductive learning*, in *Machine Learning: An Artificial Intelligence Approach*, J.G.C. R. S. Michalski, and T. M. Mitchell (Eds),Tioga, Palo Alto, CA, ed, J.G.C. R. S. Michalski, and T. M. Mitchell (Eds),Tioga, Palo Alto, CAa Morgan Kaufmann, ISBN 0935382054, 1983.
[20] Mitchell, T.M., *Machine Learning*, editora McGraw-Hill International Editions, ISBN 0-07-042807-7, numero de pág. 1-414, 1997.
[21] Morik, K., et al., *Knowledge acquisition and machine learning: theory, methods and applications*, Knowledge-based systems, editora Academic Press, ISBN 0-12-506230-3, 1993.
[22] Schuurmans, D., *Machine Learning course notes*, acesso em 2001/03/13, University of Waterloo, *htttp://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/1_ml.html*, 1999.
[23] Domingos, P., *E4-Machine Learning*, in *Handbook of Data Mining and Knowledge Discovery*, New York:Oxford University Press, pp. 660-670, 2002.
[24] Thrun, S., et al., *Automated Learning and Discovery: State of the Art and Research Topics in a Rapidly Growing Field,* AIMagazine, Vol. 20 (3), pp. 78-82, 1999.
[25] Merz, C.J., *Classification and Regression by Combining Models*, University of California, numero de pág. 207, 1998.
[26] Brodley, C.E., *Automatic Selection of Split Criterion during Tree Growing Based on Node Location, ICML - International Conference on Machine Learning*, Tahoe City, California, USA, S.J.R. Armand Prieditisa Morgan Kaufmann, ISBN ISBN 1-55860-377-8, pp. 73-80, July 9-12, 1995, 1995.
[27] Tan, A.-H., *Text Mining: The state of the art and the challenges, PAKDD'99 workshop on Knowledge Disocovery from Advanced Databases*, Beijing, pp. 65-70, 1999.
[28] Grishman, R. and B. Sundheim, *Message understanding Conference - A brief history, 16th International Conference on Computational Linguistics*, Copenhagen, 1996/06, 1996.

[29] Eliassi-Rad, T., *Building Intelligent Agents that Learn to Retrieve and Extract Information*, Department of Computer Sciences, University of Wisconsin-Madison, numero de pág. 152, 2001.
[30] Cunningham, H., *Information Extraction - a User Guide (Second Edition)*, acesso em 2004/04/02, sheffieldnlp, *http://www.dcs.shef.ac.uk/~hamish/IE/userguide/main.html*, 2000.
[31] Callif, M.E. and R. Mooney, *Relational learning of pattern-match rules for information extraction, Sixteenth National Conference on Artificial Intelligence*, Orlando, Florida,USAAAAI Press, pp. 328-334, 1999.
[32] Cui, H., P.B. Heidorn, and H. Zhang, *An approach to automatic classification of text for information retrieval, ACM/IEEE CS Joint conference on Digital Libraries*, Portland, Oregon, USA, 2002/07/14-18, 2002.
[33] Nogueira, L. and E. Oliveira, *A Multi-Agent System for E-Insurance Brokering, NET.OBJECTDAYS 2002 Worshops Agent Technologies, Infrastructures, Tools and Applications for e-services*, Erfurt, Germany, R.M. R.Kowalczyk, H.Tianfield, R.Unlanda Springer-Verlag, ISBN 3-540-00742-3, pp. 354-372, October, 2002.
[34] Freitag, D., *Toward General-Purpose Learning for Information Extraction, Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, Montreal, Quebec, Canada, C. Boitet and P. Whitelocka Morgan Kaufmann Publishers, pp. 404-408, 10-14 August, 1998.
[35] Agichtein, E. and L. Gravano, *Querying Text Databases for Efficient Information Extraction, 19th IEEE International Conference on Data Engineering*, Bangalore, IndiaColumbia University, ISBN 078037665X, pp. 113- 124, 5-8 March 2003, 2003.
[36] Agichtein, E. and L. Gravano, *Snowball: Extracting relations from large plain-text collections, 5th ACM International Conference on Digital Libraries*, 2000/06, 2000.
[37] Grishman, R., S. Huttunen, and R. Yangarber, *Real-time event extraction for infectious disease outbreaks, Human language Technology Conference (HTL)*, 2002.
[38] Yangarber, R. and R. Grishman, *NYU: Description of the Proteus/PET System as Used for MUC-7 ST, Message Understanding Conference Proceedings MUC-7*NIST, 1998.
[39] Ray, S. and M. Craven, *Representing sentence structure in hidden Markov models for information extraction, Seventeenth International Joint conference on Artificial Intelligence*, Seattle, WA, USAAAAI Press, pp. 1273-1279, 2001.
[40] Wolpert, D.H., *Stacked Generalization, Neural Networks*Pergamon Press, pp. 241-259, 1992.
[41] Rijsbergen, C.J.v., *Information Retrieval*, Second ed., editora Butterworths, London, 1979.
[42] Carroll, J., et al., *Sparkle Work Package 1 Specification of Phrasal parsing*, acesso em 2003/02/01, *http://www.ilc.pi.cnr.it/*, 1997.

[43] Hirschman, L., et al., *Chapter 13 Evaluation*, in *Survey of the State of the Art in Human Language Technology*, R.A. Cole, et al., eds, R.A. Cole, et al.a, 1996.
[44] Yang, Y. and X. Liu, *A re-examination of text categorization methods, SIGIR'99*, 1999.
[45] Rognvaldsson, D., *Generalization*, acesso em 2003/03/15, Denni Rognvaldsson, 2001.
[46] Sarle, W., *What are cross-validation and bootstrapping?*, *http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html*, 2002.
[47] Zhang, P., *On the distributional properties of model selection criteria,* Journal of the American Statistical Association, Vol. 87 (419), pp. 732-737, 1992.
[48] Shao, J., *Linear model selection by cross-validation,* Journal of the American Statiscal Association, Vol. 88 (422), pp. 486-494, June 1993, 1993.
[49] Kohavi, R., *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, IJCAI Fourteenth International Joint Conference on Artificial Intelligence*, Montréal, Québec, Canada, M. Kaufmanna, pp. 1137-1145, August 20-25 1995, 1995.
[50] Fresno, V. and A. Ribeiro, *Features selection and dimensionality reduction in Web pages representation, CIMA'2001 Computational Intelligence: Methods & Applications*, Bangor, Wales, United Kingdom, L.L. Kunchevaa ICSC Academic Press, ISBN 3-906454-26-6, pp. 416-421, 2001.
[51] Zavrel, J., P. Berck, and W. Lavrijssen, *Information Extraction by Text Classification: Corpus Mining for Features, Workshop Information Extraction meets Corpus Linguistics*, Athens,Greece, 2000.
[52] Yang, Y., S. Slattery, and R. Ghani, *A study of approaches to hypertext categorization,* Journal of Intelligent Information Systems, Vol. 18 (2), 2002.
[53] Ghani, R., S. Slattery, and Y. Yang, *Hypertext Categorization using Hyperlink Patterns and Meta Data, ICML'01 - The Eighteenth International Conference on Machine Learning*, pp. 178-185, 2001.
[54] Liu, H. and H. Motoda, *Less is more*, in *Feature extraction construction and selection: A data mining perspective*, H. Liu and H. Motoda, eds, H. Liu and H. Motodaa Kluwer Academic, pp. 3-12, 2001.
[55] Wang, K. and S. Sundaresh, *Selecting features by vertical compactness of data*, in *Feature extraction construction and selection: A data mining perspective*, H. Liu and H. Motoda, eds, H. Liu and H. Motodaa Kluwer Academic, pp. 71-84, 2001.
[56] Marques, J.S., *Reconhecimento de Padrões: métodos estatísticos e neuronais*, editora IST Press, ISBN 972-8469-08-X, numero de pág. 284, 1999.
[57] Siedlecki, W., *On Automatic Feature Extraction,* International Journal of Pattern Recognition and Artificial Intelligence, Vol. II, pp. 197-220, 1988.
[58] Yang, Y. and J.P. Pedersen, *A Comparative Study on Feature Selection in Text Categorization, ICML'97 - Fourteenth International Conference on Machine Learning*, San Francisco, USA, D.H. Fishera Morgan Kaufmann Publishers, pp. 412-420, 1997.

[59] Dunning, T.E., *Accurate methods for the statistics of surprise and coincidence,* Computational Linguistics, Vol. 19 (1), pp. 61-74, 1993.
[60] Kira, K. and L. Rendell, *A Practical approach to feature selection, Ninth International Conf. Machine Learning*, pp. 249-256, 1992.
[61] Kononenko, I., *Estimating attributes: Analysis and extensions of Relief, Seventh European Conf. Machine Learning*, pp. 171-182, 1994.
[62] Hall, M.A. and G. Holmes, *Benchmarking Attribute Selection Techniques for Discrete Class Data Mining,* IEEE Transactions on knowledge and data engineering, Vol. 15 (6), pp. 1437-1447, November/December, 2003.
[63] Sikonja, M. and I. Kononenko, *An adaptation of Relief for attribute Estimation in regression, 14th International Conf. Machine Learning*, pp. 296-304, 1997.
[64] Liu, H. and R. Setiono, *A Probabilistic Approach to feature selection: A filter solution, 13th International Conf. Machine Learning*, pp. 319-327, 1996.
[65] Gama, J.M.P.d., *Combining Classification Algorithms*, Departamento de Ciências de Computadores, Faculdade de Ciências da Universidade do Porto, numero de pág. 193, 1999.
[66] Joachims, T., *Text Categorization with Support Vector Machines: Learning with Many Relevant Features, 10th European Conference on Machine Learning*, C. Nedellec and C. Rouveirola Springer Verlag, Heidelberg, DE, pp. 137-142, 1998.
[67] Apte, C., F. Damerau, and S. Weiss, *Text Mining with Decision Rules and Decision Trees, Automated Learning and Discovery: Workshop on Learning from Text and the Web*, Pittsburgh, PA, 1998.
[68] Pazzani, M. and D. Billsus, *Learning and Revising User Profiles: The identification of interesting web sites.,* Machine Learning, Vol. 27 (3), pp. 313-331, June, 1997.
[69] Webb, G. and M. Pazzani, *Adjusted Probability Naive Bayesian Induction, 11th Autralian Joint Conference on Artificial Intelligence*, Brisbane, Australia, 1998.
[70] Hastie, T., R. Tibshirani, and J. Friedman, *The elements of statistic learning, Data Mining, Inference, and prediction*, Springer Series in statistic, editora Springer, ISBN ISBN 0387952845, 2001.
[71] Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.
[72] Quinlan, J.R., *Induction of decision trees machine Learning*, Machine Learning, numero de pág. 81-106, 1986.
[73] Quinlan, J., *C4.5: Programs for Machine Learning*, editora Morgan Kaufmann, San Mateo, 1993.
[74] Shannon, C.E., *Mathematical theory of communication*, in *Claude Elwood Shannon: Collected Papers*, N.J.A. Sloane and A.D. Wyner, eds, N.J.A. Sloane and A.D. Wynera IEEE Press

John Wiley & Sons, ISBN 0-7803-0434-9, 1993.
[75] Mingers, J., *An empirical comparison of pruning methods for decision tree induction,* Machine Learning, Vol. 4, pp. 227-243, 1989.
[76] Quinlan, J.R., *Rule induction with statistical data - a comparasion with multiple regression.,* Journal of the Operational Research Society, Vol. 38, pp. 347-352, 1987.
[77] Windeatt, T. and G. Ardeshir, *Tree pruning for output coded ensembles, 16th International Conference on Pattern Recognition*, Quebec, Canada, ISBN 0-7695-1695-X, pp. 92-95, 11–15 August, 2002.
[78] Bishop, C.M., *Bayesian Methods for Neural Networks*, 1995, Aston University, pp. 1-18, *www.ncrg.aston.ac.uk*, 1995.
[79] Domingos, P., Michael, and J. Pazzani, *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss,* Machine Learning, Vol. 29 (2-3), pp. 103-130, 1997.
[80] Alkoot, F. and J. Kittler, *Experimental Evaluation of Expert Fusion Strategies,* Pattern Recognition Letters, Vol. 20, pp. 1361-1369, 1999.
[81] Kuncheva, L., J. Bezdek, and R. Duin, *Decision Templates for Multiple Classifier Fusion: An Experimental Comparision,* Pattern Recognition, Vol. 34 (2), pp. 299-314, 2001.
[82] Turmer, K. and J. Ghosh, *Linear and Order Statistics Combiners for Pattern Classifiers*, in *Combining Artificial Neural Nets*, A. Sharkey, ed, A. Sharkeya, pp. 127-161, 1999.
[83] Kittler, J., et al., *On Combining Classifiers,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20 (3), pp. 226-239, 1998.
[84] Kuncheva, L.I., *A Theoretical Study on Six Classifier Fusion Strategies,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24 (2), pp. 281-286, 2002.
[85] Al-Kofahi, K., et al., *Combining multiple classifiers for text categorization, Tenth international conference on Information and knowledge management*, Atlanta, Georgia, USA, N. ACM Press New York, USAa, ISBN 1-58113-436-3, pp. 97-104, 2001.
[86] Larkey, L.S. and W.B. Croft, *Combining classifiers in text categorization, 19th annual international ACM SIGIR Conference on Research and development in information retrieval*, Zurich, SwitzerlandACM Press New York, NY, USA, ISBN 0-89791-792-8, pp. 289-297, 1996.
[87] Yang, Y., T. Ault, and T. Pierce, *Combining multiple learning strategies for effective cross validation, International Conference on machine Learning*, pp. 1167-1182, 2000.
[88] Z. H. Zhou, W.T., *Selective Ensemble of Decision Trees*, Lecture Notes in Artificial Intelligence 2639, editora Springer, Berlin, numero de pág. 476-483, 2003.
[89] Lima, L.R.S.d., A.H.F. Laender, and B.A. Ribeiro-Neto, *A hierarchical approach to the automatic categorization of medical documents, Seventh international conference on Information and knowledge management*, Bethesda, Maryland, United StatesACM Press New York, NY, USA, ISBN 1-58113-061-9, pp. 132-139, 1998.
[90] Breiman, L., *Stacked Regressions,* Machine Learning, Vol. 24 (2), pp. 49-64, 1996.

[91] Ting, K.M. and I.H. Witten, *Issues in Stacked Generalization,* Journal of Artificial Intelligence Research, Vol. 10, editora AI Access Foundation and Morgan Kaufmann Publishers, pp. 271-289, 99-05, 1999.
[92] Bennett, P.N., S.T. Dumais, and E. Horvitz, *Probabilistic combination of text classifiers using reliability indicators: models and results, 25th annual international ACM SIGIR conference on Research and development in information retrieval*, Tampere, FinlandACM Press New York, NY, USA, ISBN 1-58113-561-0, pp. 207-214, 2002.
[93] Breiman, L., *Bagging predictors,* Machine Learning, Vol. 24 (2), pp. 123-140, 1996.
[94] Schapire, R.E., *The Strength of Weak Learnability,* Machine Learning, Vol. 5, pp. 197-227, 1990/06, 1990.
[95] Schapire, R.E., et al., *Boosting the margin: A new explanation for the effectiveness of voting methods,* Annals of Statistics, Vol. 26 (5), pp. 1651-1686, 1998.
[96] Fensel, D., *Ontologies: A silver bullet for knowledge Management and Electronic Commerce*, editora Springer-Verlag, Berlin Heidelberg, ISBN 3-540-41602-1, 2001.
[97] Britannica, E., *Encyclopædia Britannica Premium Service*, em, 2004.
[98] Gruber, T.R., *Toward principles for the design of ontologies used for knowledge sharing,* IJHCS, Vol. 43 (5/6), pp. 907-928, November 1995, 1995.
[99] Berners-Lee, T., J. Hendler, and O. Lassila, *The semantic web,* Scientific American, May, 2001.
[100] Gruber, T.R., *A translation approach to portable ontology specifications,* Knowledge Acquisition, Vol. 5 (2), pp. 21-66, 1998, 1998.
[101] Quillian, M.R., *A notation for Representing Conceptual Information: An Application to Semantics and Mechanical English Paraphrasing*, October 1963, Dept. of Energy technology, num. SP-1395, 1963.
[102] Quillian, M.R., *Semantic Memory*, em *Semantic Information Processing*, pp. 227-270, 1968.
[103] Meersman, R., *Ontologies and Databases: More than a Fleeting Resemblance*, 2002/10/22, Vrije Universiteit Brussel, num. STAR-2001-03, pp. 8, 2001.
[104] Meersman, R.A., *The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems., CODAS*, Wollongong, Australia, Y. Zhanga Springer Verlag, Berlin, pp. 1-14, 1999.
[105] Nations, U., *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport*, acesso em 2003/10/14, *http://www.unece.org/trade/untdid/*, 2003.
[106] McCarthy, J. and P.J. Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, in *Machine Intelligence 4*, B.M.a.D. Michie, ed, B.M.a.D. Michiea Edinburgh University Press, pp. 463--502, 1969.

[107] Brakel, J.v., *The Complete Description of the Frame Problem,* PSYCOLOQUY, editora J. van Brakel, 1992.
[108] Borgida, A. and R.J. Brachman, *Description Logic Handbook*, ed. F. Baader, et al., editora Cambridge University Press, ISBN 0521781760, numero de pág. 574, 2003.
[109] Calvanese, D., et al., *Handbook of Automated Reasoning*, ed. J.A. Robinson and A. Voronkov, Vol. II, editora Elsevier Science Publishers (North-Holland), Amsterdam, ISBN 0-262-18223-8, numero de pág. 2150, 2001.
[110] Nardi, D., et al., *The Description Logic Handbook: Theory, Implementation and Applications*, ed. F. Baader, et al., editora Cambridge University Press, ISBN 0521781760, numero de pág. 574, 2003.
[111] Lambrix, P., *Description Logics*, acesso em 2003/11/01, Patrick Lambrix, *http://www.ida.liu.se/labs/iislab/people/patla/DL/*, 2003.
[112] Horrocks, I. and U. Sattler, *Description Logics - Basic, Applications, and More*, acesso em 2003/11/01, Ulrike Sattler, *http://www.cs.man.ac.uk/~horrocks/Slides/*, 2001.
[113] Randell, B. and A. Skonnard, *A Guide to XML and Its Technologies*, acesso em 2004/02/03, *http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/xmlguide.asp*, 1999.
[114] W3C, *Resource Description Framework (RDF): Concepts and Abstract Syntax*, acesso em 2003/11/10, *http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/*, 2002.
[115] DAML, *The DARPA Agent Markup Language Homepage*, acesso em 2003/11/15, DARPA, *www.daml.org*, 2003.
[116] Hendler, J., *Announcing DAML+OIL*, acesso em 2003/11/01, Jim Hendler, *http://lists.w3.org/Archives/Public/www-rdf-logic/2001Jan/0041.html*, 2001.
[117] W3C, *DAML+OIL (March 2001) Reference Description*, acesso em 2003/11/10, W3C, *http://www.w3.org/TR/daml+oil-reference/*, 2001.
[118] W3C, *Annotated DAML+ OIL Ontology Markup*, acesso em 2003/11/10, W3C, *http://www.w3.org/TR/daml+oil-walkthru/*, 2001.
[119] Greaves, M., *2004 DAML Program Directions*, acesso em 2003/11/26, DAML Program Manager, *http://www.daml.org/listarchive/daml-all/att-0301/01-2004_DAML_Program_Directions.doc*, 2003.
[120] Genesereth, M.R., *Knowledge Interchange Format (KIF) ANSI Draft NCITS.T2/98-004.*, acesso em 2003/11/01, *http://logic.stanford.edu/kif/kif.html*, 1998.
[121] Cycorp, *Cycorp it's just common sense*, acesso em 2003/10/17, Cycorp, *http://www.cyc.com/*, 2003.
[122] W3C, *RDF Vocabulary Description Language 1.0: RDF Schema*, acesso em 2004/05/08, *http://www.w3.org/TR/rdf-schema/*, 2004.
[123] W3C, *OWL Web Ontology Language Overview*, acesso em 2003/11/20, W3C, *http://www.w3.org/TR/owl-features/*, 2003.

[124] DCMI, *Dublin Core Metadata Initiative*, acesso em 2004/01/03, *http://dublincore.org/*, 2004.
[125] Reem Al-Halimi, R.C.B., J. F. M. Burg, Martin Chodorow, Christiane Fellbaum, Joachim Grabowski, Sanda Harabagiu, Marti A. Hearst, Graeme Hirst, Douglas A. Jones, Rick Kazman, Karen T. Kohl, Shari Landes, Claudia Leacock, George A. Miller, Katherine J. Miller, Dan Moldovan, Naoyuki Nomura, Uta Priss, Philip Resnik, David St-Onge, Randee Tengi, Reind P. van de Riet, Ellen Voorhees., *WordNet - An Electronic Lexical Database*, ed. C. Fellbaum, editora MIT Press, ISBN ISBN 0-262-06197-X, numero de pág. 423, 1998.
[126] Fox, M.S. and M. Gruninger, *Enterprise Modelling,* AI Magazine, pp. 109-121, Fall 1998, 1998.
[127] Uschold, M., *The Enterprise Ontology*, *http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html*, 2003.
[128] One, C., *xCBL.org*, acesso em 2003/10/31, Commerce One, *htt://www.xcbl.org*, 2003.
[129] cXML, *Commerce XML resources*, acesso em 2003/10/31, cXML, *http://www.cxml.org*, 2003.
[130] DCMI, *Dublin Core Metadata Initiative*, acesso em 2003, DCMI, *http://dublincore.org*, 2003.
[131] Farquhar, A., R. Fikes, and J. Rice, *The Ontolingua Server: a Tool for Collaborative Ontology Construction, Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Alberta, Canada, November, 1996.
[132] Hovy, E., K. Knight, and M. Junk, *Large Resources Ontologies (SENSUS) and Lexicons*, *http://www.isi.edu/natural-language/projects/nlg-publications.html*, 2001.
[133] Wiederhold, G., et al., *Scalable Knowledge Composition (SKC)*, *http://www-db.stanford.edu/SKC/*, 2000.
[134] Schreiber, G., et al., *Knowledge Engineering and Management The CommonKADS Methodology*, editora The Mit Press, ISBN ISBN 0-262-19300-0, 1999.
[135] Parunak, V.D., *Practical and Industrial Applications of Agent-Based Systems*, Industrial Technology Institute, pp. 1-41, *gunther.smeal.psu.edu/7913.html*, 1998.
[136] Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., editora Prentice Hall, ISBN 0137903952, numero de pág. 1132, 2002.
[137] Genesereth, M.R. and S.P. Ketchpel, *Software agents,* Communications of the ACM,Special Issue on Intelligent Agents, Vol. 37 (7), pp. 48-53, 1994/07, 1994.
[138] Maes, P., *On Software Agents: Humanizing the global computer,* IEEE Internet computing, Vol. I, pp. 10-19, July/August, 1997.
[139] Coen, M.H., *SodaBot: A Software Agent Environment and Construction System*, Junho, 1994, MIT, num. AI Technical Report 1493, pp. 78, 1994.
[140] Wooldridge, N.R.J.M.J., *Intelligent agents:theory and practice,* The knowledge Engineering Review, Vol. 10 (2), pp. 115-152, 1995.

[141] Michael Luck, P.M., Chris Preist, *Agent Technology: Enabling next generation Computing A roadmap for Agent Based computing*, ISBN 0854 327886, numero de pág. 94, 2003.
[142] Wooldridge, M. and P. Ciancarini, *Agent-Oriented Software Engineering: The State of the Art, Agent-Oriented Software Engineering, First International Workshop*, Limerick, IrelandSpringer-Verlag, ISBN 3-540-41594-7, pp. 1-28, June 10, 2000, 2000.
[143] Oliveira, E.d., J.M. Fonseca, and A. Steiger-Garção, *MACIV: A DAI Based Resource Management System,* Applied Artificial Intelligence, Vol. 11 (6), editora Taylor & Francis, pp. 525-550, September, 1997.
[144] Fonseca, S.P., M.L. Griss, and R. Letsinger, *Agent Behavior Architectures. A MAS Framework Comparison*, 2001/12/19, HP Laboratories Palo Alto, pp. 1-8, 2001.
[145] Wooldridge, M., N.R. Jennings, and D. Kinny, *The Gaia Methodology For Agent-Oriented Analysis And Design,* Autonomous Agents and Multi-Agent Systems, Vol. 3 (3), editora Kluwer Academic publishers, pp. 285--312, 2000.
[146] Kinny, D., M. Georgeff, and A. Rao, *A Methodology and Modeling technique for systems of BDI agents., Seventh European workshop on modeling autonomous agents in a multi-agent world*, W.V.d. Velde and J.W. Perrama Springer-Verlag, Berlin Germany, pp. 56-71, 1996.
[147] Collinot, A. and A. Drogoul, *Using the Cassiopeia Method to Design a Soccer Robot Team,* Applied Articial Intelligence (AAI) Journal, Vol. 12 (2-3), pp. 127-147, 1998.
[148] Brazier, F., et al., *Formal specification of multi-agent systems: a real-world case, First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USAAAAI Press/MIT Press, pp. 25-32, June, 1995.
[149] Spivey, J., *The Z Notation*, editora Prentice Hall, ISBN 013983768X, numero de pág. 155, 1992.
[150] Iglesias, C.A., M. Garijo, and J. Centeno-González, *A Survey of Agent-Oriented Methodologies, 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, M. J.P., S. M.P., and A. Raoa Springer-Verlag London, UK, ISBN 3-540-65713-4, pp. 317-330, 1998/07/04-07, 1998.
[151] Wooldridge, M. and N.R. Jennings, *Intelligent agents: Theory and practice,* The Kwowledge Engineering Review, Vol. 10 (2), editora Cambridge University Press, pp. 115-152, June, 1995.
[152] Wooldridge, M., *Reasoning about rational agents*, editora Bradford Books, ISBN 0-262-23213-8, numero de pág. 241, 2000.
[153] Dennett, D.C., *The intentional Stance*, Bradford Book, editora MIT Press: Cambridge,MA, ISBN 0-262-04093-X, numero de pág. 380, 1987.

[154] Rao, A.S. and M.P. Georgeff, *BDI Agents: from theory to practice, Proceedings of the First International Conference on Multi-agent Systems ICMAS-95*, San Francisco, USA, V. Lesser and L. Gassera AAAI Press / MIT Press, pp. 312-319, 12-14 June, 1995.
[155] Cohen, P.R. and H.J. Levesque, *Intention is choice with commitment,* Artificial Intelligence, Vol. 42 (3), pp. 213-261, 1990.
[156] Fisher, M., *A survey of Concurrent METATEM - the language and its applications, First International Conference on Temporal Logics*, Boom, Germany, D.M. Gabbay and H.J. Ohlbacha Springer-Verlag: Berlin, Germany, pp. 480-505, July, 1994.
[157] Rosenschein, S.J. and L.P. Kaelbling, *A Situated View of Representation and Control,* Artificial Intelligence, Vol. 73, pp. 515-540, 1995.
[158] Manna, Z. and A. Pnueli, *Temporal verification of reactive Systems - Safety*, editora Springer-Verlag, New York, 1995.
[159] Brooks, R.A., *How To Build Complete Creatures Rather Than Isolated Cognitive Simulators,* Arquitecture for Intelligence, K. VanLehn ed, editora Lawrence Erlbaum Assosiates, Hillsdale,NJ, pp. 225-239, 1989.
[160] Jack, *Intelligent Agent Reaoning*, Agent Oriented Software Group, *http://www.agent-software.com/shared/home/reasoning.html*, 2004.
[161] Acronymics, *Agent Builder*, acesso em 2003/12/29, Acronymics, *www.agentbuilder.com/AgentTools*, 2004.
[162] CSWL, *UPnp, Jini and Salutation - A look at some popular coordination frameworks for future networked devices*, California Software Labs, *http://www.cswl.com/whiteppr/tech/upnp.html*, 2003.
[163] Prado, J.E. and P.R. Wurman, *Non-cooperative Planning in Multi-Agent, Resource-Constrained Environments with Markets for Reservations, Edmonton*, The AAAI-02 Workshop on Planning with and for Multiagent Systems, pp. 60-66, 28 July, 2002.
[164] Muguda, N., P.R. Wurman, and R.M. Young, *Experiments with Planning and Markets in Multiagent Systems,* SIGecom Exchanges, Vol. 5 (1), pp. 34-47, 2004.
[165] Cox, J.S. and E.H. Durfee, *Discovering and Exploiting Synergy Between Hierarchical Planning Agents, Edmonton*, The AAAI-02 Workshop on Planning with and for Multiagent SystemsAAAI Press, ISBN ISBN 1-57735-165-7, pp. 15-23, 28 July, 2002.
[166] Finin, T., et al., *Specification of the KQML Agent-Communication Language*, 15 July, Enterprise Integration Technologies, num. EIT TR 92-04, pp. 34, *http://www-ksl.stanford.edu/knowledge-sharing/papers/*, 1993.
[167] Finin, T., et al., *KQML: An Information and Knowledge Exchange Protocol*, in *Knowledge Building and Knowledge Sharing*, K. Fuchi and T. Yokoi, eds, K. Fuchi and T. Yokoia Ohmsha and IOS Press, pp. 338, 1994.
[168] FIPA00061, *FIPA ACL Message Structure Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.

[169] Tacla, C.A. and J.-P. Barthès, *A Multi-Agent Architecture for Knowledge Acquisition, AAAI Spring Symposium*, Menlo Park, L.v. Elst, V. Dignum, and A. Abeckera AAAI Press, ISBN 1-57735-178-9, pp. 159-166, 24-26 March, 2003.
[170] Pease, A. and J. Li, *Agent-Mediated Knowledge Engineering Collaboration, Agent Mediated Knowledge Management, International Symposium*, Stanford, CA, USA, L.v. Elst, V. Dignum, and A. Abeckera Springer, ISBN 3-540-20868-2, pp. 405-415, 24-26 March, 2004.
[171] Nunes, L. and E. Oliveira, *Learning from Multiple Sources, Third International Joint Conference on Autonomous Agents and Multiagent Systems*, New YorkACM Press, pp. 1106-1113, 19 - 23 July, 2004.
[172] Prasad, M.V.N., V.R. Lesser, and S.E. Lander, *Cooperative Learning over Composite Search Spaces: Experiences with a Multi-agent Design System., Thirteenth National Conference on Artificial Intelligence*, pp. 68-73, January, 1996.
[173] Oliveira, E., J.M. Fonseca, and N.R. Jennings, *Learning to be Competitive in the Market, AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, Orlando, FL, USA, S. Sena AAAI Press, ISBN 1-57735-096-0, pp. 30-37, 1999.
[174] Vidal, J.M. and E.H. Durfee, *Predicting the expected behavior of agents that learn about agents: the CLRI framework,* Autonomous Agents and Multi-Agent Systems, Vol. 6 (1), editora Kluwer Academic Publishers, ISSN 1387-2532, pp. 77-107, December, 2003.
[175] Rouchier, J. and S. Thoyer, *Modelling a European decision making process with heterogeneous public opinion and lobbying: the case of the authorization procedure for placing Genetically Modified Organisms on the market, AAMAS 2003 - 2nd Joint Conference on Autonomous Agents and Multi-Agent Systems - The 4th Workshop on Multi-Agent Based Simulation*, Melbourne, Australia, D. Hales, et al.a Springer-Verlag, ISBN 3-540-20736-8, pp. 149-167, 14th July, 2003.
[176] Oliveira, E. and L. Sarmento, *Emotional Advantage for Adaptability and Autonomy, AAMAS 2003 - 2nd Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Victoria, Australia, J. Rosenschein, et al.a ACM Press, ISBN 1-58113-683-8, pp. 305-312, 14-18 July, 2003.
[177] Shah, A.P. and A.R. Pritchett, *Work Environment Analysis: Environment Centric Multi-Agent Simulation for Design of Socio-technical Systems, Third International Joint Conference on Autonomous Agents and Multiagent Systems - Joint Workshop on Multi-Agent and Multi-Agent-Based Simulation*, Columbia University in New York City, USA, 19 July 19, 2004.
[178] Izumi, K., T. Yamashita, and K. Kurumatani, *Analysis of Learning Types in an Artificial Market, Third International Joint Conference on Autonomous Agents and Multiagent Systems - Joint Workshop on Multi-Agent and Multi-Agent-Based Simulation*, Columbia University in New York City, USA, 19 July, 2004.

[179] Becu, N., et al., *A Methodology for Eliciting and Modelling Stakeholders' Representations with Agent Based Modelling, AAMAS 2003 - 2nd Joint Conference on Autonomous Agents and Multi-Agent Systems - The 4th Workshop on Multi-Agent Based Simulation*, Melbourne, Australia, D. Hales, et al.a Springer-Verlag, ISBN 3-540-20736-8, pp. 131-149, 14th July, 2003.
[180] IBM, *Aglets*, IBM, *http://www.trl.ibm.com/aglets/*, 2002.
[181] IBM, *Autonomic Computing*, IBM, *http://www.research.ibm.com/autonomic/index.html*, 2003.
[182] Fuller, I.J., *An overview of the HP NewWave environment,* Hewlett-Packard Journal, August, 1989.
[183] Patil, R., *Coordination of Evolving Conventions Enabling Sharing of Knowledge*, acesso em 2004-03-13, The Knowledge Sharing Effort, *http://www.isi.edu/isd/KRSharing/*, 2001.
[184] Group, A.W., *OMG Agent Platfrom Special Interest Group*, acesso em 2004-03-13, Agent Working Group, *http://www.objs.com/agent/agent-psig-mission.html*, 2001.
[185] FIPA, *FIPA*, acesso em 2003/01/01, FIPA, *www.fipa.org*, 2002.
[186] Vidal, J.M., *FIPA Introduction*, acesso em 2002/12/01, José M. Vidal, *http://jmvidal.cse.sc.edu/talks/fipaintro/*, 2002.
[187] Dale, J., *Open Standards for Interoperating Agent-Based Systems*, in *Software Focus*Wiley, 2001.
[188] FIPA00001, *FIPA Abstract Architecture Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[189] FIPA00067, *FIPA Agent Message Transport Service Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[190] FIPA00075, *FIPA Agent Message Transport Protocol for IIOP Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[191] FIPA00084, *FIPA Agent Message Transport Protocol for HTTP Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[192] FIPA00076, *FIPA Agent Message Transport Protocol for WAP Specification*, em *Foundation for Intelligent Physical Agents*, 2000.
[193] FIPA00085, *FIPA Agent Message Transport Envelope Representation in XML Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[194] FIPA00088, *FIPA Agent Message Transport Envelope Representation in Bit-Efficient Encoding Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[195] FIPA00071, *FIPA ACL Message Representation in XML Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[196] FIPA00070, *FIPA ACL Message Representation in String Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.

[197] FIPA00069, *FIPA ACL Message Representation in Bit-Efficient Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[198] FIPA00008, *FIPA SL Content Language Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[199] FIPA00009, *FIPA CCL Content Language Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[200] FIPA00010, *FIPA KIF Content Language Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[201] FIPA00011, *FIPA RDF Content Language Specification.*, em *Foundation for Intelligent Physical Agents*, 2000.
[202] Mladenic, D., *Text-learning and related intelligent agents: A survey,* IEEE Intelligent Systems, Vol. 14, pp. 44-54, July-Aug.1999, 1999.
[203] Rucker, J. and M.J. Polanco, *Siteseer: personalized navigation for the Web,* Communications of the ACM, Vol. 40 (3), editora ACM Press New York, NY, USA, ISSN 0001-0782, pp. 73-76, March, 1997.
[204] Konstan, J.A., et al., *GroupLense: Applying Filtering to Usenet News,* Communications of the ACM, Vol. 40 (3), pp. 77-87, March, 1997.
[205] Maes, P., *Agents that Reduce Work and Information Overload,* Communications of the ACM, Vol. 37 (7), editora ACM, pp. 31-40, July, 1994.
[206] Krulwich, B., *The bargainfinder agent: Comparison price shopping on the internet*, in *Agents, Bots, and other Internet Beasties*, J. Williams, ed, J. Williamsa Macmillan Publishing, pp. 257-263, 1996.
[207] Joachims, T., D. Freitag, and T. Mitchell, *WebWatcher: A Tour Guide for the World Wide Web, IJCAI97-International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997.
[208] Goldman, C.V., A. Langer, and J.S. Rosenschein, *Musag: An Agent that Learns What You Mean,* Journal of Applied Artificial Intelligence, Vol. 11 (5-6), 1997.
[209] Lieberman, H., *Letizia: An Agent That Assists Web Browsing, Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, C.S. Mellisha Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, ISBN 1-55860-363-8, 1995.
[210] Lieberman, H., C. Fry, and L. Weitzman, *Why Surf Alone?: Exploring the Web with Reconnaissance Agents,* Communications of the ACM, pp. 69-75, August, 2001.
[211] Kamba, T., H. Sakagami, and Y. Koseki, *Anatagonomy: A personalized news-paper on the World Wide Web,* Human-Computer Studies, Vol. 46 (6), editora Academic Press, Inc. Duluth, MN, USA, ISSN 1071-5819, pp. 789-803, June, 1997.
[212] Ault, T. and Y. Yanq, *kNN, Rocchio and Metrics for Information Filtering at TREC-10*, Nov, 2001.

[213] Sousa, P., *Um sistema de apoio ao desenvolvimento de interfaces de conversão de dados, baseado na norma ISO10303*, Departamento de Engenharia Informática, Universidade Nova de Lisboa, numero de pág. 129, 1999.
[214] Bokma, A., *CogNet: Integrated Information and knowledge management and its use in virtual organizations, PROVE2000 - E-business and Virtual Enterprises Managing Business-to-Business cooperation*, Florianopolis, Brazil, L. Camarinha-Matos, H. Afsarmanesh, and R. Rabeloa Kluwer Academic, ISBN 0-7923-7205-0, 2000.
[215] Standford, *Protégé*, em, 2004.
[216] CSELT, *Introduction More Info*, acesso em 2002/12/23, CSELT, *http://sharon.csel.it/projects/jade/doc/html/intro.htm*, 2002.
[217] Bellifemine, F., A. Poggi, and G. Rimassa, *JADE - A FIPA-compliant agent framework, 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, London, UK, 1999, 1999.
[218] Eckel, B., *Thinking in JAVA*, editora NJ. Prentice Hall, Upper Sandle River, 1998.
[219] Wolpert, D.H., *The relationship between PAC, the Statistical Physics framework, the Bayesian framework and the VC framework*, The Santa Fe Institute, pp. 96, 1994.
[220] Schaffer, C., *A conservation law for generalization performance, Eleventh International Conference on Machine Learning*, Rutgers University, New Brunswick, H.H. W. Cohena NJ. Morgan Kaufmann, pp. 259-265, 1994.
[221] NIST, *Text Retrieval Conference (TREC)*, acesso em 2002/05/04, NIST, *http://trec.nist.gov/*, 2004.
[222] Larsen, J. and C. Goutte, *On Optimal Data Split for Generalization Estimation and Model Selection, IEEE Nueral Networks for Signal Processing*, Madison, Wisconsin,USA, pp. 225-234, 23-25 August, 1999, 1999.
[223] Whitley, D., *A Genetic Algorithm Tutorial,* Statistics and Computing, Vol. 4, pp. 65-85, 1994.
[224] Whitley, D., *An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls,* Journal of Information and Software Technology, Vol. 43 (14), pp. 817-831, 15 December 2001, 2001.
[225] Fingar, P., H. Kumar, and T. Sharma, *The Death of "e" and the Birth of the Real New Economy : Business Models, Technologies and Strategies for the 21st Century*, editora Meghan-kiffer Press, ISBN 0-929652-20-7, numero de pág. 360, 2001.
[226] Porter, M.E., *Competitive Strategy; techniques for analysing industries and competitors*, editora Free Press, New York, ISBN 0684841460, 1980.
[227] Burman, D., *E-procurement: Purchasing for the Internet based economy*, Butler Group, 2000.
[228] Evason, M., *Calculating e-procurement benefits*, acesso em 2002/05/23, *www.ebusinessforum.com*, 2001.

[229] Knowledgestorm, *Knowledgestorm software directory*, acesso em June, Knowledgestorm, *www.knowledgestorm.com*, 2003.
[230] Fielder, S., *New State-of-the-Art*, 2002/09, University of Sunderland, pp. 51, *www.deeepsia.com*, 2002.
[231] Sullivan, D., *2003&2002 Statistics*, acesso em 2003/11/08, SearchEngineWatch.com, *http://searchenginewatch.com/reports/article.php/2156471*, 2003.
[232] ECCMA, *UNSPSC Code*, acesso em Outubro de 2002, *http://www.eccma.org/unspsc/*, 2001.
[233] W3C, *OWL Web Ontology Language Guide*, acesso em 2003/11/20, *http://www.w3.org/TR/2003/CR-owl-guide-20030818/*, 2003.
[234] W3C, *XSL Transformations (XSLT) Version 1.0*, acesso em 2003/11/10, *http://www.w3.org/TR/1999/REC-xslt-19991116*, 1999.

Glossário

O domínio da informática é fortemente marcado pela utilização intensiva de siglas e termos, estabelecidos no meio académico e industrial. Visando simplificar o seu entendimento, aos leitores menos familiarizados, elaborou-se este glossário de siglas e termos. Neste glossário são apresentadas as traduções de termos realizadas ao longo da dissertação e o significado das siglas adoptadas. Naturalmente, evitou-se a inclusão de traduções e siglas demasiado óbvias. Os elementos aqui listados foram apresentados ao longo do texto na sua primeira ocorrência.

Termo	Descrição
ACL	Agent Communication Language
AFOSR	Air Force Office of Scientific Research
ARPA	US Advanced Reserach Projects Agency
ASP	Application Service Provider
B2B	Business-to-business
B2C	Business-to-customer
BDC	Base de dados de conhecimento
BDI	Belif-Desire-Intention
DARPA	US Defense Advanced Reserach Projects Agency
EDI	Electronic Data Interchange
FIPA	Foundation for Intelligent Physical Agents
HMI	Human machine Interface
HTML	Hyper Text Markup Language
IA	Inteligência Artificail
IIOP	Internet Inter-Orb Protocol
JADE	Java Agent Development Framework
KQML	Knowledge Query Manipulation Language
MIME	Multipurpose Internet Mail Extensions
MRO	Maintenance, repairs, operations com tradução para Manutenção, Reparação, Operação
NRI	National Research Initiative
ODBC	Open DataBase Connectivity
OIL	Ontology Interchange Language
OMG	Object Management Group
OWL	Web Ontology Language
PME	Pequena e Média empresa
ROI	Return of Investment; com tradução para Retorno de investimento;
SAD	Sistema(s) de Apoio à Decisão
SGBD	Sistema de Gestão de Base de Dados
SGML	Standard Genreralised Markup Language
SVM	Support Vector Machines
TIC	Tecnologias de Informação e Comunicação
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locators
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
IMC	Informação Mútua Condicional
IE	Extracção de Informação
IR	Recuperação de Informação

Tabela 22 - Glossário de siglas e termos

Anexos

A.1 Incerteza e entropia

Seja $\{A_i\},\ i=1,2,\ldots,N$ um sistema completo de acontecimentos (i. e., tais que cada prova de uma experiência aleatória ξ ocorre um e um só destes acontecimentos) e considera-se o seguinte esquema finito *A*.

$$A \begin{cases} A_1 & A_2 & \ldots & A_N \\ p_1 & p_2 & \ldots & p_N \end{cases} ; \quad \begin{matrix} p_i = P(A_i) \geq 0 \\ \sum_{i=1}^{N} p_i = 1 \end{matrix} \qquad (91)$$

Este esquema é, do ponto de vista formal, análogo ao esquema de uma variável aleatória tomando um número finito de valores. Interprete-se então o esquema *A* como a lei de probabilidade de uma variável qualitativa *X*, tomando os valores A_i como as probabilidades $p_i,\ i=1,2,\ldots,N$.

Cada esquema finito descreve um estado de incerteza.

Tomem-se, por exemplo, os esquemas

$$A \begin{cases} A_1 & A_2 \\ 0.5 & 0.5 \end{cases} \qquad e \quad B \begin{cases} B_1 & B_2 \\ 0.1 & 0.9 \end{cases}$$

Intuitivamente, é natural afirmar que a «quantidade de incerteza» do esquema *A* é maior do que a do esquema *B*.

Por outras palavras e interpretando *A* e *B* como distribuições de probabilidade das variáveis categoria *X* e *Y*, respectivamente, pode dizer-se que a variável *X* é mais dispersa que a variável *Y*. Para a variável *X* associada ao esquema finito (91) o único parâmetro de tendência central que se pode definir é, obviamente, a moda.

Se existir um único $p_j = max\ (p_1, p_2, \ldots, p_N)$ então a moda de *X* é A_j .

Para medir a «quantidade de incerteza» (ou dispersão) introduz-se a seguinte função:

$$H(p_1,\ldots,p_N) = -\sum_{i=1}^{N} p_i \log(p_i) \tag{92}$$

Na base 2, usual em Teria de Informação, as unidades são designadas por unidades binárias ou *bits*.

A função H diz-se a *entropia* do esquema finito (91). Neste esquema a ocorrência de A_j, por exemplo, dá uma certa quantidade de informação. Põe-se o problema de medir esta quantidade de informação.

A quantidade de informação está associada com a «incerteza» contida no esquema finito em consideração.

Parece natural admitir que quanto mais incerta for a ocorrência de A_j maior é a quantidade de informação associada à ocorrência efectiva de A_j. Então se $P(A_i)<(A_j)$, a realização de A_i dá uma quantidade de informação superior à realização de $A_{j\,j}$. Shannon propôs como medida de quantidade de informação, ligada à realização de A_j o valor $I_j = -\log P(A_j)$ que satisfaz aos requisitos postos.

De facto, se $P(A_i)<(A_j)$ então $\log P(A_i) < \log P(A_j)$, $\log P(A_i) > \log P(A_j)$; isto é $I_i > I_j$ (I_i diz-se a informação própria mútua de A_j).

A.2 Enquadramento probabilísticos de base

A.2.1 Axioma 1

A probabilidade de um evento está compreendida entre 0 e 1.

$$0 \leq P(E) \leq 1 \tag{93}$$

A.2.2 Axioma 2

A probabilidade do espaço de eventos é sempre igual a 1.

$$P(S) = 1 \tag{94}$$

A.2.3 Axioma 3

Para cada sequência de eventos mutuamente exclusivos $E_i, i = 1,2,\ldots,N$ tal que $E_i E_j = \phi$ para $i \neq j$, a probabilidade da união de eventos mutuamente exclusivos é o somatório da probabilidade dos eventos.

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i) \tag{95}$$

A.2.4 Probabilidade dos eventos complementares

A probabilidade de um evento complementar é igual a 1 menos o valor do evento.

$$P(E^c) = 1 - P(E) \tag{96}$$

A.2.5 Probabilidade da reunião de dois eventos

A probabilidade da reunião de dois eventos é igual à soma das probabilidades dos eventos menos a probabilidade da sua intersecção.

$$P(E \cup F) = P(E) + P(F) - P(EF) \tag{97}$$

A.2.6 Ocorrências equiprováveis

No caso de eventos que podem ser considerados equiprováveis a probabilidade desse evento é calculado:

$$P(E) = \frac{\left(número \quad de \quad pontos \quad no \quad evento \quad E\right)}{\left(número \quad de \quad pontos \quad no \quad espaço \quad S\right)} \tag{98}$$

A.2.7 Probabilidade condicionada

Sempre que a probabilidade de um evento E altera a probabilidade de ocorrência do evento F, estamos perante o caso de probabilidade condicionada, $P(E / F)$.

A relação entre a intersecção das probabilidade, a probabilidade condicionada e a probabilidade do evento condicionador pode ser escrita da seguinte forma:

$$P(EF) = P(E) * P(F / E) = P(F) * P(E / F) \tag{99}$$

A probabilidade de ocorrência de dois eventos é igual ao produto da probabilidade de saída de um dos eventos e da probabilidade de ocorrência do outro evento após a saída no anterior.

Dois eventos, E e F, dizem-se independentes, sempre que a probabilidade condicionada de E, $P(E/F)$ é igual à *P(E)*. Neste caso é imediato que:

$$P(EF) = P(E) * P(F) \tag{100}$$

A.2.8 Regra de Bayes

Tendo em conta o cálculo da probabilidade da intersecção de dois eventos através da probabilidade condicionada e do particionamento de espaço temos,

$$P(EF) = P(E) * P(F / E) = P(F) * P(E / F) \tag{101}$$

$$P(F / E) = \frac{P(EF)}{P(E)} = \frac{P(E / F) * P(F)}{P(E)} \tag{102}$$

A.3 Linguagem OWL

O OWL Lite é uma versão restrita do OWL onde somente um conjunto das características do OWL é utilizado. Para além do exemplo da cardinalidade já referido, uma das restrições mais relevantes está relacionada com a necessidade de definir classes em termos de superclasses nomeadas que, por definição, não podem ser expressas de forma arbitrária. A equivalência de classes, assim como todo o tipo de relações, está igualmente limitada a classes nomeadas, reduzindo a possibilidade de definir relações com classes arbitrárias.

As características-base do Schema RDF do OWL Lite incluem as definições de «Class», para descrever um grupo de indivíduos que possuem as mesmas características; «SubclassOf» para a construção de hierarquias; «Property» para descrever uma classe em função de relações com outras classes, ou através de relações com tipos de dados; «Subproperty», que define mecanismos de hierarquia entre propriedades; «Domain» e «Range», restrições globais, que limitam o domínio das propriedade permitindo mecanismos de inferência; e indivíduos instâncias de classes.

As correspondências de igualdade e desigualdade do OWL Lite são descritas através da «equivalentClass», o que permite definir a equivalência entre duas classes; «equivalentProperty» para a igualdade de propriedades; «sameAs» para a igualdade entre indivíduos; «differentFrom» para declarar a diferença entre indivíduos, «allDifferent» que permite definir que um grupo de indivíduos são mutuamente distintos.

As características das propriedades e dos seus valores em OWL Lite são realizadas através: do «inverseOf» que permite definir propriedades inversas, do «TransitiveProperty» para a descrição da transitividade entre propriedades; do «SymmetricProperty» para a descrição de propriedades simétricas, do «FunctionalProperty» e da «InverseFunctionalProperty» para descrição de funções entre propriedades, (entre outras o «UNIQUE»).

Para a descrição de restrições existem dois mecanismos-base: de restrição de quais os valores possíveis, realizado através de «allValuesFrom» e «SomeValuesFrom»; e de cardinalidade, «minCardinality», «maxCardinality», e «cardinality» que impõe relações locais entre uma propriedade e uma classe, mas que são fortemente limitadas uma vez que o OWL lite só admite valores de zero ou um. Existe, igualmente, a possibilidade de definir a relação entre classes através da intersecção, «intersectOf», e de definir tipos de dados através de «datatypes».

Quanto ao Owl DL e Owl Full, partilham o mesmo vocabulário. Todavia, no caso do DL, são impostas restrições que podem ser descritas de forma genérica como a imposição da separação de tipos, impedindo que uma classe possa ser um indivíduo ou uma propriedade, e, também, que uma propriedade possa ser descrita como do tipo objecto ou do tipo de dados. Não existem, igualmente, restrições à cardinalidade. O vocabulário que estende o

OWL Lite acrescenta entre outros, o «OneOf» para a descrição de classes enumeradas, onde o conjunto de indivíduos que podem ser instanciados está previamente definido; o «hasValue» que força a existência de valor para propriedades; o «disjointWith» para descrever classes disjuntas; os «unionOf», «complementOf» e «intersectionOf» para criação de combinações booleanas [233].

O OWL é parte integrante do conjunto de ferramentas incrementais recomendadas pelo W3C, com vista à obtenção da Web semântica:

- XML disponibiliza uma sintaxe para a criação de documentos estruturados, mas não impõe restrições semânticas aos significados dos documentos;
- Esquema XML (DTD) oferece a possibilidade de definir restrições à estrutura dos documentos XML, assim como capacidade de descrição semântica do seu conteúdo;
- Norma XSL-T [234] tem como objectivo realizar o mapeamento entre diversas terminologias;
- RDF permite a descrição de recursos e das suas inter-relações através de um modelo de dados orientado aos objectos [114];
- Esquema RDF permite a descrição de propriedades e classes de recursos RDF com uma semântica de generalização de hierarquias para propriedades e classes.

OWL que adiciona mais vocabulário para a descrição de propriedades e de classes, entre outras: relações entre classes (e. g., disjunção), cardinalidade (e. g., estritamente um), igualdade, características de propriedades (e. g., simetria) e classes enumeradas.

A.4 A plataforma JADE

Este anexo apresenta, sumariamente, a plataforma JADE, mais especificamente, a sua arquitectura, os mecanismos de comunicação, os modelos de execução dos agentes, o tipo de programação de agentes e algumas ferramentas de desenvolvimento.

A arquitectura da plataforma de desenvolvimento de Agentes está em conformidade com as especificações FIPA97 disponibilizando os agentes obrigatórios: o agente de interface de comunicação ACC, o agente de Gestão de Sistema, e o agente Facilitador de Directorias, e estando a comunicação entre agentes implementada com o recurso a mensagens FIPA ACL.

Genericamente a plataforma foi desenvolvida com o recurso à linguagem JAVA, necessitando, para ser instalada, somente da pré-instalação de uma máquina virtual JAVA. A plataforma baseia-se assim na existência de uma rede de servidores, com máquinas virtuais JAVA instaladas, onde podem existir um ou mais contentores de Agentes, que oferecem um ambiente de execução em tempo real concorrencial aos agentes neles residentes. Cada contentor de Agente é ambiente de execução multiprocesso, sendo atribuído pela plataforma um processo a cada Agente, a alguns módulos de gestão e a algumas acções de execução do sistema (e. g., envio de mensagens). Desta forma, fica assegurada a distribuição real da arquitectura, não só pela utilização de diversos servidores, como, igualmente, pelo recurso aos mecanismos de paralelismo oferecidos pela ambiente JAVA.

Existe um contentor onde estão instalados os agentes de gestão da plataforma que têm o papel de representação plataforma para o mundo exterior. A implementação de cada contentor é realizada pelo recurso a um objecto servidor de RMI, que gere localmente um conjunto de Agentes, assegurando o seu ciclo de vida (criação, activação, suspensão e destruição) e comunicação. A comunicação é assegurada pela capacidade que os contentores possuem em receber mensagens e fazer a sua condução para as filas de mensagem privadas de cada agente, assim como pela capacidade inversa de recepção de pedidos de envio de mensagens e selecção do processo de transmissão mais adequado dependendo da localização dos intervenientes.

Os mecanismos de comunicação na plataforma de agentes são realizados através dos contentores de agentes. De uma forma genérica cabe aos contentores de Agente identificar quais os processos mais eficientes para assegurar a comunicação entre os agentes e encapsular o processo de transmissão de mensagens, respeitando as directivas de ocultação dos mecanismos de comunicação impostas pela FIPA. A comunicação entre contentores é realizada pela evocação remota de métodos implementados pelo JAVA (RMI).

No momento de arranque de uma plataforma é necessário assegurar que o primeiro contentor a ser activado é o contentor de representação da plataforma (contentor principal), pois é responsável pela manutenção de um registo de RMI interno, que é utilizado pelos restantes contentores da plataforma sempre que são activados. No arranque do contentor principal é criado um registo RMI interno no servidor local em escuta no porto TCP/IP predefinido, sendo imediatamente activado o contentor principal onde estão residentes os agentes de sistema ACC, AMS e DF.

A partir deste momento, cabe ao contentor principal manter a tabela de contentores com as respectivas referências de objecto RMI, assim como uma tabela descritora global de Agentes (AGDT), onde são relacionados os nomes de todos os Agentes e os seus dados AMS e o seu contentor.

A consistência destas tabelas é assegurada dado que sempre que um contentor é activado, consulta os seus dados/registos locais para identificar o contentor principal, (referência do objecto RMI) ao qual se quer juntar, para passar a fazer parte da plataforma. Utilizando essa informação faz o seu registo na plataforma, sendo adicionado na tabela de contentores, e passa a assegurar o registo na tabela descritora global de Agentes de todos os Agentes que nele são criados e/ou destruídos.

Cada contentor mantém uma representação local das duas tabelas, evitando a sua consulta em cada comunicação. Na prática, as tabelas são actualizadas localmente sempre que uma nova mensagem é recebida, pelo registo da referência do objecto do contentor emissor e sempre que é desencadeada uma excepção na evocação de métodos remotos (por falta de página local ou por referência corrupta), momento em que toda a tabela é carregada. Com este protocolo, é possível criar um registo global de contentores e de Agentes que lida de forma eficiente com situações dinâmicas (i. e., activação/destruição de novos contentores e agentes).

No envio de mensagens, pode acontecer uma das seguintes situações dependendo da localização relativa do agente destinatário:

- **reside no mesmo contentor** – este é o caso mais favorável, no qual cabe ao contentor, fazer uma cópia do objecto mensagem ACL e fazer a sua passagem por evento para o agente destinatário. Não existe tradução da mensagem nem evocações remotas, estando os custos de comunicação somente associados à evocação do método *clone*() do objecto mensagem ACL e desencadear o evento;
- **reside na mesma plataforma mas noutro contentor** – neste caso é necessário fazer passar o objecto mensagem ACL através da evocação remota de métodos implementados pelo JAVA para utilização de objectos distribuídos. Neste caso existem duas possibilidades de custo de comunicação: ou a imagem local das

tabelas está actualizada, o que obriga somente à execução de um RMI, sendo o objecto mensagem ACL «serializado» e «desserializado» em tempo real na evocação do método, ou pelo contrário a imagem da tabela não está actualizada o que força a uma execução extra de um RMI para actualização das tabelas;

- **reside noutra plataforma de agentes JAVA –** neste caso é necessário fazer o envio da mensagem por protocolo IIOP e interface OMG IDL de acordo com as especificações FIPA. Estamos na presença de dois processos de empacotamento de mensagem de dupla tradução, no envio de objecto mensagem ACL, para cadeia de bytes IIOP passando por formato FIPA ACL (formato de cadeia de caracteres) e na recepção em processo inverso (cadeia IIOP, para FIPA ACL e finalmente objecto mensagem ACL). O envio da mensagem é realizado por intermédio do Agente ACC, instalado no contentor principal que serve de porta de saída da plataforma. Na plataforma destino, a mensagem é finalmente encaminhada para o contentor final. Os custos de comunicação estão associados à evocação remota de CORBA e a quatro processos de tradução, (dois no envio e dois na recepção), aos quais acresce o encaminhamento local à plataforma via RMI ou evento, dependendo da localização do contentor destino;
- **reside noutra plataforma de agentes não JAVA –** este é o caso mais complexo que é em tudo semelhante ao anterior até ao envio da mensagem, i. e., faz recurso ao protocolo IIOP e interface OMG IDL de acordo com as especificações FIPA, todavia, na plataforma destino, a mensagem é encaminhada pelos seus mecanismos internos.

A interface da plataforma com o exterior é única e é implementada pelo agente ACC, que é um objecto servidor IIOP CORBA, que está à escuta de evocações remotas. Cabe a este Agente a conversão entre o objecto mensagem ACL e o formato FIPA ACL e vice-versa, dependendo de corresponder a um envio ou recepção de mensagem, respectivamente.

O modelo de execução do JADE implementa a autonomia dos agentes pela utilização de um processo por agente. De uma forma mais precisa, do ponto de vista de programação distribuída, um agente é um objecto activo que contém um processo de controlo. Esta aproximação permite construir agentes com autonomia, que para além de apresentarem reacção a mensagens, podem executar múltiplas tarefas em simultâneo. Cada comportamento de um agente é capturado pela utilização do modelo de abstracção «Behaviour». Neste sentido um agente não é mais do que um processo de controlo ao qual são adicionados comportamentos através da adição de novos objectos «Behaviour».

No JADE a cada Agente é atribuído um único processo. É assim utilizado o modelo concorrencial de um processo por Agente *versus* o modelo concorrencial de um processo por comportamento de Agente. Esta opção permite assegurar a execução dos Agentes num

ambiente multiprocesso preemptivo, enquanto que os comportamentos de cada Agente são escalonados cooperativamente. A escolha de um processo por agente visa a obtenção de um sistema global mais eficiente, não sendo necessário gravar o ambiente de trabalho de cada comportamento, por exemplo sempre que é suspenso. O escalonamento dos comportamentos é implementado pela classe-base de agentes, o que permite ocultar do programador todos os problemas inerentes à selecção dos comportamentos para execução. O escalonador da classe-base implementa uma política de «round-robin» não preempitivo entre todos os comportamentos activos do Agente. O escalonador não implementa a suspensão da execução de comportamentos, pelo que é deixado ao programador a responsabilidade de gerir a filosofia colaborativa dos comportamentos de cada Agente. No caso de um Agente ser interrompido antes do processo de controlo ter terminado a sua execução, i. e., todos os comportamentos estarem terminados, será reescalonado para a próxima rodada. Um agente pode ficar suspenso, caso esteja somente à espera de uma mensagem. Neste caso o processo de controlo não fica activo e não é seleccionado pelo escalonador até que uma mensagem que lhe seja destinada seja recepcionada pelo sistema. Nessa altura, o processo de controlo do Agente é activado, e será incluído na lista de processos activos. Esta abordagem permite evitar a espera activa de mensagens, evitando assim o consumo desnecessário de tempo de CPU.

A classe-base assegura ao agente, para além do mecanismo de controlo e gestão de comportamentos, um conjunto de funcionalidades-base de interacção com a plataforma e desenvolvimento de Agentes. Entre as funcionalidades de interacção com a plataforma realça-se, a título de exemplo, o registo, activação, configuração, suspensão e a manutenção remota. Entre as funcionalidades-base de desenvolvimento de Agentes destacam-se os mecanismos de comunicação (envio e recepção de mensagens), o registo em domínios e as primitivas de protocolos de interacção normalizados.

A programação de Agentes no JADE é realizada primordialmente pelo desenvolvimento de comportamentos. Na perspectiva do programador, o seu agente é uma classe filha da classe «Agent», à qual são adicionados os comportamentos que permitem o desempenho das suas tarefas. As classes de comportamentos devem ser desenvolvidas como filhas das classes que fazem parte da hierarquia da classe «Behaviour». A hierarquia de classes «Behaviour» compreende a existência de comportamentos simples «SimpleBehaviour» e complexos «ComplexBehaviour».

Os comportamentos simples são utilizados pelo utilizador para a execução de tarefas atómicas, estando detalhados em classes de comportamentos cíclicos «CyclicBehaviour» ou únicos «OneShotBehaviour».

Os comportamentos complexos permitem a criação de Agentes com comportamentos não atómicos, logo compostos por diversos subcomportamentos. A classe-base

«ComplexBehaviour» assegura dois métodos, «addBehaviour(Behaviour)» e «removeBehaviour(Behaviour)», que permite ao programador, respectivamente, acrescentar e remover novos comportamentos à Fila de comportamentos Agentes. O detalhe da classe «ComplexBehaviour» é realizado pelas subclasses «SequentialBehaviour», «NonDeterministicBehaviour», «ParallelBehaviour» e «FiniteStateMachineBehaviour».

A criação de comportamentos complexos para um Agente implica a extensão de uma ou várias classes filhas de «ComplexBehaviour» e da sua adição à fila de comportamentos. A selecção da classe pai está dependente, principalmente, do tipo de execução que se quer ter para os subcomportamentos. Tendo em conta que «ComplexBehaviour» é filha de «Behaviour», um comportamento complexo pode incluir comportamentos simples ou complexos, sendo possível a construção de comportamentos em árvore de profundidade variável, em que o escalonador só entra em conta com os subcomportamentos de nível mais elevado.

A caracterização das funcionalidades de cada uma das seguintes classe abstractas de definição de comportamento é a seguinte:

- «Behaviour» define as funcionalidades-base de todos os comportamentos. São disponibilizados dois métodos fundamentais: i) o método «action» onde é definido o comportamento do Agente; e ii) o método «done» que permite informar o escalonador de que o comportamento está terminado pelo retorno do valor verdade. (se o valor retornado for falso, o comportamento mantêm-se na fila de comportamentos activo e será reactivado na próxima chamada);
- «OneShotBehaviour» deve ser utilizado para comportamento atómicos, que só serão executados uma única vez, e. g., a libertação de um recurso único;
- «CyclicBehaviour» deve ser utilizado para comportamento atómicos, mas que não terminam;
- «SimpleBehaviour» deve ser utilizado para comportamentos simples que possa ser capturado num comportamento único e que se podem repetir, e. g., resposta imediata a um estimulo exterior. A repetição não deve ser cíclica infinita e, nesse caso, o comportamento «CyclicBehaviour» deve ser utilizado;
- «SequentialBehaviour» deve ser utilizado para definir comportamentos que são compostos por subcomportamentos que têm que ser executados sequencialmente. O comportamento bloqueia sempre o que subcomportamento corrente bloqueia e termina quando o último comportamento estiver terminado;
- «NonDeterministicBehaviour» deve ser utilizado para definir comportamentos que são compostos por subcomportamentos que podem ser executados de forma não determinística. O fim do comportamento é determinado pela verificação de uma condição prévia que pode ser uma das seguintes: todos os subcomportamentos

terminados; um terminado; ou um subconjunto. O bloqueio do comportamento só acontece se todos os subcomportamentos activos estiverem bloqueados. O algoritmo de selecção de comportamentos utilizado é «round-robin»;

- «ParallelBehaviour» deve ser utilizado para definir comportamentos que são compostos por subcomportamentos que podem que ser executados em paralelo;
- «FiniteStateMachineBehaviour» deve ser utilizado para definir comportamentos que são compostos por subcomportamentos que por sua vez tem que ser executados respeitando o comportamento de uma máquina de estados finita. Neste caso, é permitida a definição de estados do comportamento, sequências de estados e condições de transição.

A plataforma JADE **disponibiliza como ferramentas de desenvolvimento** de agentes uma interface gráfica para a gestão da plataforma, e a monitorização e controlo do estado dos agentes. Esta interface gráfica não é mais do que um agente extra, chamado RMA (Remote Monitoring Agent) comunicando, naturalmente, com os restantes agentes por mensagens ACL, num protocolo proprietário de extensão ao protocolo ontológico de gestão de Agentes. A interface permite o controlo do ciclo de vida dos agentes (criação, suspensão, activação e destruição de agentes) no servidor local ou em servidores remotos.

A.5 Soluções comerciais para compras electrónicas

Nome	Plataforma	Características	Serviços
Achilles www.achilles.co.uk	Achilles Trading Exchange Markit NOTiCE UVDB CAPALIST	Mercado electrónico B2B para sector público do Reino Unido. Base de dados: de fornecedores para a indústria ferroviária; de oportunidades de contratos; para o sector público industrial utilities industry; para sector da Educação; para sector da Polícia;	Serviços e Ferramentas para gestão de catálogos electrónicos. Ambiente seguro com recurso a técnicas de cifra. Catálogo electrónico com uma estrutura não hierárquica unificada Não requer integração total para todos os participantes pelo que se adequa a PME Sistema escalável para novos sectores de mercado.
ANZ www.anzebiz.com		Fácil utilização Interface única de acesso Acesso imediato ao conteúdo dos catálogos por selecção de fornecedor	Selecção de fornecedores Criação de catálogos virtuais privados Acesso restrito
Ariba	Ariba PunchOut Ariba Supplier Network	Permite aos compradores um acesso rápido ao sítio Web eCommerce do fornecedor Ideal para catálogos dinâmicos e de elevada dimensão que têm que ser personalizados para cada comprador Ligações e transacções com um grande número de fornecedores Canais de distribuição e de parceiros através de ligação única via Ariba global eCommerce	Não existe a necessidade de manutenção dos conteúdos do catálogo por parte do comprador Os compradores podem activar a rede de fornecedores Ariba que permite o acesso global a uma directoria de milhares de fornecedores com processos de negociação de privacidade, segurança, visibilidade. Os fornecedores têm que ser certificados pelo Ariba com o Ariba Ready Status para serem aceites, o que assegura a sua capacidade de escalabilidade, de assegurar transacções, de disponibilizar informações aos compradores e ao mercado.

Cataloga www.cataloga.com	Channel Manager	Ferramentas para todos os aspectos relacionados com a gestão de catálogos electrónicos para os fornecedores e compradores	Assegura informação sobre o produto de elevada qualidade, precisa, rica e tecnicamente válida. Permite o carregamento de informação a partir de diversas fontes fazendo o seu mapeamento para categorias hierárquicas. Permite a definição de catálogos específicos para utilizadores Ferramentas de leilão e para promoção de quantidade e no tempo. Acesso limitado a fornecedores certificados Ariba Ready
CommercOne	CommerceOne Buy	Solução integral para o processo de aquisição para grandes empresas	Acesso à rede de fornecedores da Commerce.net e da Global Trading Web. Utiliza xCBL, facilmente integrável com sistemas legados, permite sistemas múltiplos de taxas, língua, e moeda.
Exostar www.exostar.com	Procure Pass	Acesso à base de dados de fornecedores Exostar (industria da Defesa e Aeroespacial). Acesso às PME a um conjunto de soluções estado da arte para compras electrónicas com baixo investimento, escaláveis e facilmente utilizáveis.	Permite a visualização e selecção de bens directos e indirectos. Permite a definição personalidade de fornecedores e o catálogo permite pesquisas multicritério.
First Índex www.firstindex.com	FindFASTPRO FindFAST	Controlo das relações entre fornecedores e do processo de fornecimento em sectores de engenharia e manufactura. Apresenta os melhores preços para os melhores fornecedores	Nova gama de produtos para a área das compras electrónicas. Disponibiliza uma lista completa de fornecedores, permite o registo livre, garante a colocação de pedidos de cotação, recebe as cotações, aceita as ofertas, disponibiliza meios de comunicação com os fornecedores. Taxa anual para os compradores.
GlobalSpec www.globalspec.com	SpecSearch	Ferramenta dedicada ao fornecimento de componentes.	Tirando recurso intensivo da Internet, de tecnologias de pesquisa em base de dados disponibiliza a engenheiros a identificação de componentes através de parâmetros técnicos. Definição de parâmetros de pesquisa baseados num profundo conhecimento do domínio. A utilização é livre por parte dos compradores sendo cobrada uma taxa única aos fornecedores.

I2	I2Procurement	Soluções de aquisição electrónica	Solução de gestão compreensiva e integrada de materiais MRO
ManufacturingQuote.Inc	MFGQUOTE Thomas Publishing Company Gardner Publications	Solução de compras electrónicas para serviços de manufactura	Acesso livre para compradores e uma taxa anual única de subscrição para fornecedores.
Open Market www.openmarket.com	LiveCommerce2.0	Solução informática que permite às empresas a criação intuitiva de catálogos electrónicos personalizados.	Entradas dinâmicas, multilíngua com ligação a sistemas ERP e com mecanismos de monitoração do comportamento do utilziador. Disponível em versão para instalação com o sistema Transact que faz a gestão do comércio na Internet.
Oracle	iProcurement Solution	Acesso a catálogos de fornecedores	Manutenção de conteúdos de um catálogo de fornecedores, Acesso através de directório de fornecedores ao catálogo do fornecedor.
Requisite Technology www.requisite.com	REQUISITE ProductPac BugsEye Catalog Finding Engine	Solução de fácil utilização que permite aos utilizadores encontrarem os itens catalogados de forma eficaz.	Pesquisa com o recurso a linguagem natural, interface intuitiva, com múltiplos métodos de pesquisa. Refinamento paramétrico. Personalização do catálogo e agentes de pesquisa. Catálogo com sistema de segurança avançada. Aplicável a grandes empresas.
SAQQARA	SAQQARA Catalog Management Service	Cria, gere e aloja catálogos Web privados e personalizados.	Manutenção de catálogos de múltiplos fornecedores. Importação de dados de diversos formatos com mecanismos de auxilio, e. g., filtros, ferramentas de importação e gestão de taxinomias. Disponibilização de uma vista agregada dos diversos catálogos. Pesquisa de produto unificada multifornecedor e ferramentas de auxilio à comparação dos resultados obtidos.
Simplytrading www.simplytrading.com		Portal de negócios baseado numa perspectiva ASP. Público-alvo são as PMEs, escalável e de baixo custo.	Solução incluí um mercado electrónico, gestão de catálogos, e de cadeias de fornecimento. Permite a construção de soluções Web: sítios Web, catálogos, registo de domínios. Tecnologia Microsoft.

Slingshot www.slingshotecity.com	Ebuy	Componente do sistema eCity Enterprise-Business Supply Chain Management incluí um catálogo electrónico para compradores	Integração com as aplicações dos clientes via XML. Descrição de produtos numa hierarquia de produtos. Motores de pesquisa baseados na hierarquia ou por descrição do produto. Público-alvo são as empresas de média dimensão, pelo que é adequado a PME de grande dimensão.
Zycus	Catsummit	Solução de gestão automática de catálogos de empresas. Flexível permite a gestão autónoma de catálogos para grandes organizações e mercados electrónicos.	Tecnologia Web pode ser acedido e configurado através de navegadores Web comuns. Suporta diversos formatos de dados, mapeamentos flexíveis, normalização automática baseada em regras predefinidas, classificação automática na taxinomia UNSPCC, Escalável a múltiplos domínios, permite o suporte a múltiplos catálogos. Notificação automática a mudanças de preço, mecanismos de autorização, múltiplos formatos xCBL,CXML, etc.

A.6 Os sítios Internet do corpus

Nome do sítio	Endereço URL
confetti	store.europe.yahoo.com\confetti-uk
furniture123	store.europe.yahoo.com\furniture123-uk
microwarehouse	store.europe.yahoo.com\microwarehouse
pharmacy2u	store.europe.yahoo.com\pharmacy2u-uk
Qed-uk	store.europe.yahoo.com\qed-uk
sharperimage	store.europe.yahoo.com\sharperimage-uk
auctionworks	www.auctionworks.com
comet	www.comet.co.uk
interflora	www.interflora.co.uk
kinddeals	www.kinddeals.com
pcline	www.pcline.com
posternow	www.posternow.com
tesco	www.tesco.com
thorntons	www.thorntons.co.uk
usanotebook	www.usanotebook.com
carphonewarehouse	www1.carphonewarehouse.com
larktoys	www.larktoys.com
mandmsports	http://www.mandmsports.com/
gbposters	http://www.gbposters.co.uk/
outdoorclothingonline	http://www.outdoorclothingonline.com/pages/store.html
annova	http://www.annova.biz/
audiovisual	http://www.audiovisual.co.uk/
alfaglade	http://www.alfaglade.co.uk/home.asp
find-me-a-gift	http://www.find-me-a-gift.co.uk/
jmldirect	http://www.jmldirect.co.uk/
aarons	http://www.aarons-books.co.uk/
champagne-and-flowers	http://www.champagne-and-flowers.com/acatalog/
artdiscount	http://www.artdiscount.co.uk/
10kplus	http://www.10kplus.com
acmediscount	http://www.bswebengine.com/acme7699/
cheapstationery	http://www.cheapstationery.com/
101cd	http://www.101cd.com/
avery	http://www.avery.com/
bbcshop	http://www.bbcshop.com/
abbeymusic	http://www.abbeymusic.com/
a1books	http://www.a1books.com
anotherbookshop	http://www.anotherbookshop.com/
early-childhood	http://www.early-childhood.net/
kidsonthemove	http://www.kidsonthemove.co.uk/
babylove	http://www.babylove.com
polo	http://www.polo.com/
luxestyle	http://store.yahoo.com/luxestyle/
brooksbrothers	http://www.brooksbrothers.com/

starmount	http://www.starmount.co.uk/
buyitallnow	http://www.buyitallnow.com/erol.html
cheap-computer-parts	http://www.cheap-computer-parts.co.uk/
1coolpc	http://www.1coolpc.com/
surfpc	http://www.surfpc.co.uk/
planetparts	http://www.planetparts.co.uk/
ncsales	http://www.ncsales.com
allamericanelectronics	http://www.allamericanelectronics.com/
unbeatable	http://www.unbeatable.co.uk/
bennetts-online	http://www.bennettsonline.co.uk/
encoredirect	http://www.encoredirect.co.uk/
electronics-online	http://www.electronics-online.co.uk/acatalog/
currys	http://www.currys.co.uk/
iportables	http://www.iportables.com/
go2camcorder	http://www.go2camcorder.co.uk/
hotkit	http://www.hotkit.co.uk/
simplyradios	http://www.simplyradios.com/
shasonic	http://www.shasonic.co.uk/
rswww	http://rswww.com/
ustronics	http://www.ustronics.com/
techtronics	http://www.techtronics.com
tatumelectronics	http://www.tatumelectronics.com
4everflowers	http://www.4everflowers.com/
passion-food.co.uk	http://www.passion-food.co.uk/frame.htm
floydonwine	http://www.floydonwine.com/fw_shop/
hotwines	http://www.hotwines.co.uk/
huggables	http://www.huggables.com/
creaturetoys	http://www.creaturetoys.co.uk/
partridges	http://www.partridges.co.uk/
marksandspencer	http://www.marksandspencer.com/
energy-music	http://www.energy-music.co.uk/
cd-wow	http://www2.cd-wow.com/
albatross	http://www.albatross-music.demon.co.uk/
play	http://www.playserver2.com/play247.asp?
virgin	http://www.virginmega.co.uk/pwsvm/application/pwsvm
tqgames	http://www.tqgames.co.uk/index.cfm
brain-builders	http://www.brain-builders.com/
compendiumonline	http://www.compendiumonline.co.uk/
mvc	http://www.mvc.co.uk/
hoylesonline	http://www.hoylesonline.com/
games-web	http://www.games-web.co.uk/
hobbyworkshop	http://www.hobbyworkshop.com/
sportsgameshop	http://www.sportsgameshop.com/
cduniverse	http://www.cduniverse.com/
streetsonline	http://www.streetsonline.co.uk
whsmith	http://www.whsmith.co.uk/whs/Go.asp
allmyhome	http://www.allmyhome.com/

bombayduck	http://www.bombayduck.co.uk/
chinacraft	http://www.chinacraft.co.uk/home_frameset.html
detail-online	http://www.detail-online.co.uk/asp/default.asp
foundat	http://www.foundat.co.uk/acatalog/index3.html
lillianvernon	http://www.lillianvernon.com/
lakelandlimited	http://www.lakelandlimited.com
Jeelee	http://www.jeelee.com/
loft-sf	http://www.loft-sf.com/
Cartoongallery	http://www.cartoongallery.co.uk/
Health4all	http://www.health4all.co.uk/
Perfumania	http://www.perfumania.com/default.aspx

A.7 Lista de paragem de palavras inglesas

A-Cons	Cont-Go	H-Mad	Man-Pl	Po-sta	Sti-Wan	Was-Your
A	contrariwise	had	Many	possible	still	was
About	Cos	halves	May	present	studies	we
above	could	hardly	Maybe	presented	such	week
according	crd	has	me	presents	supposing	well
across	cu	hast	meantime	provide	tested	were
after	day	hath	meanwhile	provided	than	what
afterwards	described	have	might	provides	that	whatever
against	describes	he	more	quite	the	whatsoever
albeit	designed	hence	moreover	rather	thee	when
All	determine	henceforth	most	really	their	whence
almost	determined	her	mostly	related	them	whenever
alone	different	here	more	report	themselves	whensoever
along	discussed	hereabouts	mr	required	then	where
already	do	hereafter	mrs	results	thence	whereabouts
also	does	hereby	ms	round	thenceforth	whereafter
although	doesn't	herein	much	said	there	whereas
always	doing	hereto	must	sake	thereabout	whereat
among	dost	hereupon	my	same	thereabouts	whereby
amongst	doth	hers	myself	sang	thereafter	wherefore
An	double	herself	namely	save	thereby	wherefrom
And	down	him	need	saw	therefore	wherein
another	dual	himself	neither	see	therein	whereinto
Any	due	hindmost	never	seeing	thereof	whereof
anybody	during	his	nevertheless	seem	thereon	whereon
anyhow	each	hither	next	seemed	thereto	wheresoever
anyone	either	hitherto	no	seeming	thereupon	whereto
Anything	else	how	nobody	seems	these	whereunto
Anyway	elsewhere	however	none	seen	they	whereupon
Anywhere	enough	howsoever	nonetheless	seldom	this	wherever
Apart	et	I	noone	selected	those	wherewith
Are	etc	ie	nope	selves	thou	whether
Around	even	if	nor	sent	though	whew
As	ever	in	not	several	thrice	which
At	every	inasmuch	nothing	sfrd	through	whichever
Author	everybody	inc	notwithstanding	shalt	throughout	whichsoever
Av	everyone	include	now	she	thru	while
Available	everything	included	nowadays	should	thus	whilst
Be	everywhere	including	nowhere	shown	thy	whither
Became	except	indeed	obtained	sideways	thyself	who
Because	excepted	indoors	of	significant	till	whoa
Become	excepting	inside	off	since	to	whoever
Becomes	exception	insomuch	often	slept	together	whole

Becoming	exclude	instead	ok	slew	too	whom
Been	excluding	into	on	slung	toward	whomever
Before	exclusive	investigated	once	slunk	types	whomsoever
Beforehand	far	inward	one	smote	towards	whose
Behind	farther	inwards	only	so	unable	whosoever
Being	farthest	is	onto	some	under	why
Below	few	it	or	somebody	underneath	will
Beside	ff	its	other	somehow	unless	wilt
besides	first	itself	others	someone	unlike	with
between	for	just	otherwise	something	until	within
beyond	formerly	kind	ought	sometime	up	without
Both	forth	kg	our	sometimes	upon	worse
But	forward	km	ours	somewhat	upward	worst
By	found	last	ourselves	somewhere	upwards	would
Can	from	latter	out	spake	us	wow
cannot	front	latterly	outside	spat	use	ye
Canst	further	less	over	spoke	used	yet
Certain	furthermore	lest	own	spoken	using	year
Cf	furthest	let	per	sprang	various	yippee
Cfrd	general	like	performance	sprung	very	you
choose	given	little	performed	srd	via	your
conducted	get	ltd	perhaps	stave	vs	yours
considered	go	made	plenty	staves	want	yourself
						yourselves

A.8 Frequência das características, por intervalos de selecção

A.8.1 Ordenadas por Informação Mútua

	<=0	<=1	<=2	<=3	<=4	<=5	<=6	<=7	<=8	<=9	<=10
100	0	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0	0	0
400	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
600	0	0	0	0	0	0	0	0	0	0	0
700	0	0	0	0	0	0	0	0	0	0	5
800	0	0	0	0	0	0	0	0	0	6	11
900	0	0	0	0	0	0	0	0	0	16	21
1000	0	0	0	0	0	0	0	0	11	27	35
1100	0	0	0	0	0	0	0	0	11	27	35
1200	0	0	0	0	0	0	0	20	31	58	66
1300	0	0	0	0	0	0	0	20	31	58	72
1400	0	0	0	0	0	0	55	75	86	113	127
1500	0	0	0	0	0	0	55	75	99	126	140
1600	0	0	0	0	0	0	55	75	99	131	147
1700	0	0	0	0	0	27	82	102	126	158	174
1800	0	0	0	0	0	58	113	149	173	205	221
1900	0	0	0	0	0	58	113	149	173	205	221
2000	0	0	0	0	0	58	113	149	186	222	239

A.8.2 Ordenadas por Qui-quadrado

	<=0	<=1	<=2	<=3	<=4	<=5	<=6	<=7	<=8	<=9	<=10
100	0	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0	0	0
400	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
600	0	0	0	0	0	0	0	0	0	0	5
700	0	0	0	0	0	0	0	0	0	16	24
800	0	0	0	0	0	0	0	0	11	38	46
900	0	0	0	0	0	0	0	20	31	58	72
1000	0	0	0	0	0	0	0	20	44	71	85
1100	0	0	0	0	0	0	33	53	77	109	123
1200	0	0	0	0	0	0	55	75	99	131	147
1300	0	0	0	0	0	0	55	91	115	147	163
1400	0	0	0	0	0	0	55	91	128	164	180
1500	0	0	0	0	0	58	113	149	186	222	239
1600	0	0	0	0	0	58	113	149	186	222	239
1700	0	0	0	0	0	58	199	235	272	308	325
1800	0	0	0	0	0	58	243	279	316	352	369
1900	0	0	0	0	0	58	243	290	337	379	399
2000	0	0	0	0	63	121	306	353	400	442	462

A.9 Cooperação com Universidade de São Paulo

Os trabalhos apresentados nesta dissertação estiveram na origem dos seguintes trabalhos de mestrado:

- Jorge Felix Herrera: Uso de Data Warehousing e Data Mining na Busca de Relações e Conhecimento em um Ambiente de Comércio Eletrônico.

- José Martins Junior: Classificação de Páginas na Internet

- Fernando Aires de Oliveira: Extração de Informação sobre Produtos Comercializados em Páginas Brasileiras para Alimentação de Catálogos Eletrônicos.

- Cláudio Policastro: Manutenção de Casos num Sistema Híbrido de Raciocínio.

- Alfredo de Aragão Lanari: Utilização de Redes Neurais em Hipermídia Adaptativa.

- Antonio Jose Brandão: Uso de Ontologias para Classificação de Vulnerabilidades em Sistemas Computacionais.

Estiveram, igualmente, na origem das seguintes publicações:

1 - A, Herrera J F, Martins J. R,, Moreira, E. S. A *Model for Data Manipulation and Ontology Navigation in DEEPSIA project* na First Seminar on Advanced Research in Eletronic Business, Proceedings of the First Seminar on Advanced Research in Eletronic Business, p.139 – 145, 2002

2 - G, Milagres F, Moreira, E. S., P, Pimentão J, Sousa P A, *Dealing with Security within DEEPSIA project* em ICIS 2002, 2002 publicado nos Proceedings of the WSEAS international Conference on Information Security, p.2431 – 2439, 2002.

3 - M, Odeh M, Moreira, E. S., L, Madeira T, *Eletronic Commerce and Its Socio-Economic Implications in Brazilian Small and Medium Enterprised* em IEEE 2002 International Symposium on Technology and SocietyRaleigh. Proceedings of the International Symposium on Technology and Society, p.45 - 50, 2002.

4 - Milagres, Francisco Gomes, Moreira, E. S., Pimentão, João Paulo, Sousa, Pedro Alexandre Costa, Garção, A S, *Security Analysis of a Multi-Agent System in DEEPSIA project* na First Seminar on Advanced Research in Electronic Business, Rio de Janeiro. Proceedings of the First Seminar on Advanced Research in Electronic Business, p.155 – 162, 2002.

5 - Eulalia, Luiz Antonio, Moreira, E. S., Ferreira, Andre Ponce Leon *Using Ontologies for Intelligent Agent Training and Information Retrieval In: European Conference on Product and Process Modeling*, Portoroz. E-Commerce Application Case Study in ECCPPM, p.277 – 284, 2002.

6 - Junior, José Martins, Moreira, E. S., *Using Support Vector Machines to recognize products for sale in e-Commerce pages*, na Artifitial Intelligence and Applications, AIA Innsbruck, Austria, Fev/2004.