
Universidade de Aveiro Departamento Electrónica e Telecomunicações
2004

Manuel Alberto Vaia dos Reis

Alguns Trilhos para Arqueologia Documental

Universidade de Aveiro Departamento Electrónica e Telecomunicações
2004

Manuel Alberto Vaia dos Reis

Alguns Trilhos para Arqueologia Documental

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do

Prof. Dr. Joaquim Arnaldo Carvalho Martins, professor associado com agregação do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro,

e

Prof. Dr. Joaquim Manuel Henriques de Sousa Pinto, professor auxiliar do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

o júri

presidente

Profª. Drª. Ana Maria Perfeito Tomé
professora associada
da Universidade de Aveiro

Prof. Dr. Joaquim Arnaldo Carvalho Martins
professor associado com agregação
da Universidade de Aveiro

Prof. Dr. Joaquim Manuel Henriques de Sousa Pinto
professor auxiliar
da Universidade de Aveiro

Prof. Dr. José Luís Brinquete Borbinha
professor auxiliar do Instituto Superior Técnico
da Universidade Técnica de Lisboa

agradecimentos

Ao longo da história, o registo dos factos através da palavra escrita, deixa-nos a memória das coisas e das pessoas. “A longa marcha para o esquecimento”, escrita por Jaime Gralheiro, peça de teatro encenada por José Carretas e levada à cena em 21 de Abril de 1989 pelo Círculo Experimental de Teatro de Aveiro – CETA, criou marcantes momentos de aprendizagem sobre a história e as gentes desta região. Uma parte da peça descreve os empolgantes momentos da abertura da Barra de Aveiro (1808) e tem um protagonista, o Engenheiro Luiz Gomes de Carvalho, que, com entusiasmo ou desalento, descreve os avanços e recuos das transformações liberais dos princípios do Séc. XIX.

Também aos estimados Profs. Drs. Arnaldo Martins e Joaquim Sousa Pinto, o pretexto que suscitaram para remexer nas memórias deste período.

Ao Eng. Pedro Almeida, pela amizade e apoio atento.

Ao Manuel, por alguns momentos que prescindiu da presença do seu pai.

À Rosa, minha companheira e amiga, pelo incentivo e compreensão.

resumo

O presente trabalho procura identificar instrumentos de processamento de linguagem natural, e a possibilidade da sua integração com o serviço de indexação da Microsoft. O processamento de linguagem natural é analisado na perspectiva da análise dos erros resultantes do OCR e na definição de um esquema de classificação de documentos.

abstract

This work aims to identify natural language processing tools and evaluates its integration capabilities with Microsoft's Indexing Service. The natural language processing faces the OCR's resulting errors problem and the classification schema's definition.

Sumário

Índice de Figuras

Índice de Tabelas

Introdução

A Assembleia da República disponibiliza actualmente os textos das actas parlamentares na Internet [ARDEBATESINTRO]. Essa "publicitação" tem decorrido em "modo faseado" em consequência da "grande quantidade de informação disponível" [ibid.]. "Numa primeira fase, desenvolvida entre meados de 2000 e o primeiro terço de 2002, foram tratadas as Actas Parlamentares pertencentes ao período histórico que medeia entre 1935 e a actualidade" [ibid.]. "Na segunda fase estão a ser tratadas as Actas Parlamentares que medeiam entre 1821 - data de início da actividade parlamentar em Portugal - e 1926, correspondendo aos períodos históricos designados por Monarquia Constitucional e 1ª República" [ibid.]. É no contexto desta segunda fase que o presente trabalho está inserido.

Tendência para uniformidade linguística

Tanto a Monarquia Constitucional (1821-1910) como a 1ª República (1910-1926), foram períodos históricos ricos em alterações convulsivas no sistema social e político. As actas parlamentares, testemunho dessas alterações e reflectindo também o sistema político vigente num determinado momento histórico, acompanham uma "espécie de Babel ortográfica" [ESTRELA93: 10], que caracterizou a ortografia da língua portuguesa até inícios do século passado (XX). Com um tal espólio, num período que atravessa a Reforma Ortográfica de 1911 [ibid.], seria tarefa interessante examinar se as actas parlamentares – dos períodos da Monarquia Constitucional e da 1ª República – contêm alguma uniformidade linguística, desde que confinadas a períodos temporais bem delimitados. Há um conjunto de argumentos que favorecem a hipótese de tal acontecer.

Na sua interessante "Breve Gramática do Português Contemporâneo" (Ed. João Sá da Costa, Lisboa, 2002), Celso Cunha e Lindley Cintra [CUNHA02], distinguem três tipos de diferenças ou variações: diatópicas, diastráticas e diafásicas[1]. As variações diatópicas estão relacionadas com o espaço geográfico: "falares locais, variantes regionais, e, até, intercontinentais". As variações diastráticas têm a ver com "diferenças entre camadas socioculturais". As variações diafásicas prendem-se a "modalidade expressiva", seja falada, escrita, literária ou outras.

Não serão de esperar, no contexto parlamentar português e nas épocas consideradas, grandes variações diatópicas ou diastráticas.

Os deputados escolhidos, se por um lado são originários, em regra, do espaço territorial português (continente e ilhas), por outro são educados ou convivem estreitamente com os centros de poder, de culto e académicos. O facto de conviverem entre si no decurso dos trabalhos

parlamentares, e de se verem obrigados a residir próximo da assembleia, poderá contribuir também para uma certa redução das variações diatópicas de que, porventura, sejam portadores.

A gama de estratos sociais de que são originários, também não é muito alargada. O próprio conceito de deputado surge, em alguma literatura, associado a um certo estatuto uniforme de erudição e destaque na sociedade portuguesa.

Estas considerações situam-se num contexto de actas parlamentares escritas, ou seja, discursos, relatos mais ou menos fiéis de intervenções orais, apreciações, projectos de lei, etc. Por conseguinte a modalidade escrita terá uma preponderância muito grande neste contexto. Eventualmente pode surgir uma ou outra modalidade especial que tem a ver com a especificidade do assunto sobre o qual se está a falar. Por exemplo, se uma acta apresentar uma discussão em torno de um regulamento de energia eléctrica é de esperar que contenha palavras como Watt e Volt ou expressões como kWh ou kVA.

Sendo escrita, notar-se-á, certamente, o efeito da "força centrípeta da conservação" [ibid.].

Esta noção, de uma certa uniformidade linguística interna dos documentos balizados no tempo e no espaço, é também testemunhada por autores empenhados no tratamento automático de documentos[2]. Pode-se considerar, portanto, que as actas parlamentares obedecem a um padrão de utilização da linguagem, muito embora não seja esse o objectivo do presente trabalho.

Objectivo da dissertação

O problema colocado consiste em avaliar a possibilidade de utilizar os índices do serviço de indexação da Microsoft com dois objectivos. O primeiro é a sua integração em ferramentas de classificação de documentos. O segundo visa encontrar formas de correcção erros. Naturalmente, está implícita a natureza automática ou semi-automática de qualquer um destes objectivos.

Esta formulação impõe necessariamente o exame das possibilidades oferecidas pelo *Indexing Service*. Adicionalmente é necessária uma visão ampla sobre as investigações e técnicas correntes situadas em torno daqueles objectivos. Finalmente é necessário procurar os mapeamentos possíveis entre as possibilidades oferecidas e as técnicas ou investigações identificadas.

Estrutura da dissertação

No capítulo 1 são descritos os conceitos de catalogação e classificação documental. A classificação aparece como o esquema de organização das diferentes classes. A catalogação como a materialização de um esquema de classificação tendo em conta os objectos existentes. A análise de clusters é apresentada como um instrumento metodológico de classificação

[1] Op. Cit., p. 2

documental. Identifica-se a noção de ruído dependente do contexto, como distinta do ruído esperado. Mostra-se, exemplificando, a necessidade de utilizar instrumentos de identificação de erros sintácticos e semânticos, que percorram os textos em detalhe. Neste capítulo distinguem-se duas grandes áreas, dos domínios da Inteligência Artificial e da "Information Retrieval".

No capítulo 2 são especificamente apresentados alguns conceitos sobre redes neuronais. Segue-se uma abordagem de trabalhos em torno do processamento de linguagem natural utilizando redes neuronais.

No capítulo 3, inserido no âmbito da *Information Retrieval*, expõem-se definições de conceitos fundamentais. Também mostra que é neste domínio que se podem situar soluções de classificação automática de documentos.

O *Indexing Service* é apresentado no capítulo 4. Começa-se por uma resenha histórica que visa situar no tempo a implementação deste sistema. Evidencia-se que o seu aparecimento se deveu, sobretudo, à necessidade de implementar uma solução de indexação para servidores Web da Microsoft. Descrevem-se também alguns aspectos de funcionamento e conceitos que lhe estão associados, bem como formas de acesso à informação residente na sua base de dados.

No capítulo 5 relacionam-se os métodos disponibilizados pelo *Indexing Service* com as técnicas de classificação de documentos. Discute-se também a sua aplicabilidade neste contexto.

No capítulo final, Resultados e Conclusões, expõem-se alguns aspectos determinantes nas opções tomadas, reporta-se uma experiência de análise de textos, estabelecem-se algumas conclusões e perspectivam-se algumas orientações para trabalho futuro.

[2] Cf. [SALTON88: 515], "(...), it became clear that most automatically derived term dependencies were valid only locally in the documents from which the dependent term groups were originally extracted".

Capítulo 1. Tratamento documental

Sob este título examina-se o significado de alguns conceitos relacionados com o tratamento de documentos. Começa-se por precisar o entendimento genérico em torno dos conceitos de classificação e catalogação. Tomando como referência os textos das Actas Parlamentares, aponta-se uma metodologia de classificação e estabelecem-se algumas considerações em torno do significado de ruído e palavras ruidosas. A análise da linguagem e do problema dos erros, por outro lado, evidencia a necessidade de ter em conta e prestar a devida atenção aos detalhes.

1.1. Classificação

Como o nome sugere, classificação exprime definição de classes. Ou seja, a classificação de documentos visa associar cada documento a uma classe integrada num esquema de classificação.

O que se pretende classificar é o conhecimento contido nos documentos em "formato electrónico" resultantes da "extracção do texto através da sua imagem digitalizada" [3]. Há, portanto, que criar classes de conceitos a utilizar na construção do esquema de classificação de documentos inseridos na biblioteca digital.

1.1.1. Esquema de classificação

Um esquema de classificação é uma estrutura que organiza a disposição de classes, relacionando-as ou ordenando-as entre si. Naturalmente, perante uma determinada espécie de objectos, classificar exige a identificação de características que os objectos têm em comum, nos mais variados níveis de detalhe, seguindo as necessidades que a classificação visa resolver.

Se essa identificação prévia não for efectuada, qualquer esquema de classificação assim estabelecido terá sempre uma natureza precária e limitada. Precária porque facilita o suscitar de novas características, com o que isso implica de alterações ao esquema de classificação; limitada porque, não tendo sido identificadas características diferenciadoras em número suficiente, fica limitado o número de níveis de desagregação dos objectos.

Habitualmente as classes de conceitos são sucessivamente subdivididas, criando-se uma estrutura hierárquica com classes e subclasses de tal modo que, tomando a classe como um conjunto, as subclasses serão seus subconjuntos.

[3] Da Proposta de Dissertação.

1.1.2. Exemplos de esquemas de classificação

O sistema de classificação de documentos adoptado da Biblioteca da Universidade de Aveiro é um exemplo de um esquema hierárquico designado por Classificação Decimal Universal – CDU, que os Serviços de Documentação da UA disponibilizam, e se encontra acessível na Web [SDUA].

O pórtico da Assembleia da República [AR], apresenta a informação dividida segundo um critério de classificação próprio e adequado à informação que disponibiliza. Também aqui há uma estrutura hierárquica, oferecida aos seus visitantes ou utilizadores, destinada a simplificar as acções de pesquisa de informação.

1.2. Diferença entre classificação e catalogação

Enquanto o objectivo da classificação de documentos é a criação das designações das classes necessárias ao centro de documentação, a catalogação trata da componente física que reúne a informação sobre os documentos. Numa biblioteca, por exemplo, o catálogo é constituído por fichas onde estão registados os assuntos, os autores, os títulos, bem como outras referências. Por sua vez estas fichas estão arrumadas em ficheiros distintos, e ordenadas segundo aqueles critérios ou outros que se revelem como necessários.

Com a informatização a tendência é reunir toda esta informação numa única base de dados. Nestas circunstâncias os catálogos são materializados pela disponibilidade de casos de utilização de acesso à informação.

Tomando o servidor Web como um centro de documentação, são disponibilizados diversos instrumentos de catalogação. O mais simples e imediato é proceder à arrumação dos documentos em pastas ou directórios criados no disco para os receber. Neste caso o catálogo é materializado pela listagem dos diferentes directórios. A regra adoptada para a nomenclatura e distribuição das diversas pastas e ficheiros pode constituir uma classificação.

Perante a tendência de dispersão de ficheiros pelo disco e pela Web, foram sendo também introduzidos mecanismos intrínsecos aos documentos, que permitem a utilização de ferramentas simples de localização e catalogação. A criação de metadados visa precisamente permitir e agilizar a catalogação de documentos por parte dos seus autores ou proprietários. Os dispositivos de visualização ou alteração de documentos separam o respectivo conteúdo das propriedades inscritas pelos metadados. O catálogo constituído desta forma, embora com a sua informação distribuída pelos diferentes documentos, é materializado pela listagem da informação associada aos metadados estabelecidos. Essa informação pode ser estabelecida com base em esquemas de classificação.

1.3. O Tratamento das Actas Parlamentares

Num contexto de *Actas dos Debates Parlamentares das Cortes Geraes e Extraordinárias da Nação Portugueza (1821)*[4], não seria muito complicado criar um esquema de classificação simples, apenas baseado em referências cronológicas associadas a acontecimentos históricos. De facto, o catálogo que está implementado apresenta casos de utilização que se aproximam muito desta descrição. São exemplos disso a possibilidade de consultar as actas por data e, filtrando por ano, obter um índice que facilita essa consulta.

Numa amostra observada, de ficheiros resultantes do OCR das actas, verificou-se que estão distribuídos na base de uma estrutura de directórios com, pelo menos, três camadas em árvore invertida correspondentes ao ano, mês e dia. O nome de cada ficheiro, para além desta informação, é complementado com o número da folha. Por sua vez estes ficheiros, em formato HTML, contêm metadados que reiteram a informação relativa à data e número de folha.

1.3.1. Classificação por análise de clusters

Conhecer, com alguma precisão, que assuntos foram efectivamente debatidos e quais as suas relações constitui uma dificuldade. Trata-se de classificar, por assunto, um volume muito grande de actas, isto é, identificar quais os documentos que abordam este ou aquele tema. Na identificação de qualquer classe interessa associar apenas os termos extraídos dos documentos, que efectivamente a caracterizam.

As características dos documentos sob a forma de texto armazenados em disco, são as palavras, termos, expressões ou, em última análise, simples caracteres. Assim, é necessário começar por reunir ou agregar as expressões características dos documentos e determinar possíveis relações entre elas. A agregação de palavras e expressões é comparável à anotação, efectuada recorrendo a uma leitura rápida sem atender a detalhes, de algumas palavras ou expressões significativas para a caracterização do documento. Sobre este assunto, e corroborando estas considerações, pode ler-se [WILLETT88]:

> "Normalmente, classificação significa a associação de objectos a classes pré-definidas, enquanto que a análise de agregados exige a identificação destas classes; assim, a agregação deve preceder a classificação (...) " [5]

Com os textos armazenados num disco de computador e aproveitando o seu poder de cálculo, tendencialmente tais tarefas podem e devem ser desempenhadas automaticamente.

A análise de agregados é habitualmente conhecida na literatura como *cluster analysis* e, entre nós, há quem a designe por "Análise de Clusters" [REIS01: 287]. Esta última designação

[4] As considerações sobre Actas Parlamentares expostas ao longo deste capítulo, referem-se exclusivamente às aplicações de consulta disponibilizadas em linha [ARDEBATESMC].
[5] Orign. "Classification normally refers to the assignment of objects to predefined classes whereas cluster analysis requires the identification of these classes; thus, clustering must precede classification(...)" [WILLET88:577]

será adoptada ao longo deste trabalho. A análise de clusters é um conjunto de procedimentos estatísticos, da área da estatística multivariada, que "podem ser usados para classificar objectos e pessoas sem preconceitos" [ibid.]. Estas metodologias serão abordadas com mais detalhe no capítulo 3.

1.3.2. O ruído e o contexto ruidoso

O agente que executa os procedimentos de classificação, para além de ignorar as características inerentes ao contexto, também ignora elementos mais detalhados que não têm qualquer significado para a distinção entre documentos. Se essa leitura for efectuada por um especialista humano, naturalmente conhece o contexto dos documentos ignorando-o nas suas anotações. Se este agente for um programa de computador é necessário que disponha dessa informação, no sentido de a não utilizar, considerando-a como ruído sempre que ocorra.

As actas parlamentares estão recheadas com elementos que caracterizam o contexto. Considere-se o exemplo do ficheiro A1822M06D10-0396.htm, reproduzido no Anexo III. Trata-se da folha 396 referente à acta da sessão do dia 10 de Junho de 1822. Analisando o código do ficheiro, encontra-se a seguinte linha:

<meta name="LIMITS" content="0392-0412">.

Esta informação delimita a acta daquela sessão, ou seja, esta página está inserida num conjunto de páginas que se estende desde a 392 até à 412.

Sendo uma página intermédia, não surpreende que seja omissa relativamente às palavras "ACTA" ou "SESSÃO". Contudo, sem atender ao código ou à nomenclatura utilizada para designar esse ficheiro, apenas examinando o texto, a sua disposição e conteúdo, é possível inferir que se trata de um fragmento com número 396 de uma ACTA de reunião ou, como já se sabe, SESSÃO. Lendo o texto com alguma atenção a pormenores, saltam à vista expressões como "reino", "Membros do Congresso", "Commissão", "Sr. Ferreira Borges" e a frase "Interrompeu o Sr. Presidente o debate, a fim de participar as Cortes que (...)". Estas expressões, para um leitor com informação em múltiplas áreas do conhecimento, nomeadamente o humanístico incluindo a história portuguesa, já é suficiente para caracterizar, aproximadamente, o enquadramento deste texto.

Nestas expressões podem distinguir-se as características que definem o contexto, das outras que, estando omissas, se mostram desnecessárias. O conjunto de palavras {"reino", "membros", "congresso", "commissão", "ferreira", "borges", "interrompeu", "presidente", "debate", "fim", "participar", "cortes"}, é um conjunto de características que indiciam um contexto, i. e, um determinado género de documentos no conjunto dos residentes no servidor[6] .

[6] Um exemplo idêntico, e exprimindo o mesmo conceito, está exposto em [RAJMAN99].

Por outro lado o conjunto {“do”, “sr.”, “o”, ”a”, ”de”, “as”, “que”} não proporciona qualquer indicação de caracterização. Se, na visualização de documentos apenas fossem apresentados elementos pertencentes a este conjunto, dir-se-ia, sem qualquer espécie de escrúpulo, que se estaria a observar lixo, ou melhor, *ruído*. Estaria comprometida qualquer possibilidade de distinção útil entre documentos.

As considerações, estabelecidas nos dois últimos parágrafos, sugerem duas naturezas de palavras, ou características, ruidosas. Em primeiro lugar aquelas que dependem do contexto. Neste caso a delimitação e o número de elementos serão estabelecidos por aproximação. Em segundo lugar as que são inerentes à linguagem. O conjunto destas palavras será bem delimitado tendo, em consequência, cardinalidade fixa.

1.3.3. A linguagem e os erros

Uma outra ordem de problemas, que a análise das actas das Cortes Gerais coloca, é o da linguagem, por um lado tomada no seu conjunto, e por outro nas particularidades da sua ortografia quando comparada com a contemporânea. Este último problema prende-se com a criação de instrumentos de detecção de erros.

Tomando novamente como exemplo a folha 396 da acta do dia 10 Junho 1822, um leitor atento aos detalhes localizaria certamente algumas irregularidades. A cerca de um quarto do texto pode ler-se: “como provedores elles arrecadão as terras reaes” (sic). O mesmo trecho na imagem[7] do original é: “como provedores elles arrecadão as terças reaes”. Os substantivos “terras” e “terças”, isoladamente, pertencem ao léxico da língua portuguesa. Com o adjectivo “reaes”, ambas as expressões, “terras reaes” ou “terças reaes”, são válidas.

Contudo há dois aspectos que alertam para a possibilidade da ocorrência de erro. O primeiro consiste na presença do verbo “arrecadar”, na conjugação da terceira pessoa no plural, “elles arrecadão”. As “terras“ não serão, propriamente, um tipo de objecto que se possa “arrecadar“. Em segundo lugar, examinando o contexto em que esta frase se insere, verifica-se que se trata da “cobrança” de “imposições”, “rendimentos de commendas”, “dizimarias pertencentes ao thesouro”, “prestimonios” e por aí adiante. A combinação destes duas irregularidades semânticas, conduziu à suspeita e, por sua vez, à comparação com o fac-símile do original.

Mais ou menos a meio do texto pode ler-se “depois do terem falado os ultimo. Preopinantes;” (sic). Este trecho apresenta duas irregularidades sintácticas evidentes. A primeira consiste em não fazer sentido a contracção da preposição “de” com o artigo definido “o” antes do verbo “ter”. A segunda irregularidade é a discordância em número entre o artigo definido “os” e a palavra “ultimo”. Esta derradeira palavra está seguida de um ponto final e parece um objecto directo e, por esse motivo, um substantivo. Mas a frase seguinte começa com uma oração bizarra

por ser constituída apenas por um substantivo, “Preopinantes”. Perante esta sucessão de erros sensíveis, a inevitável comparação com o original esclarece o leitor. O trecho original na imagem do texto é “depois de terem falado os ultimos Preopinantes;”.

A análise efectuada indicia a necessidade de se encontrar ferramentas que *leiam* em detalhe os textos e confiram a sua obediência a padrões sintácticos e semânticos. Estes exemplos ilustram erros resultantes da aquisição e conversão por OCR. O caso da substituição da palavra “terças” por “terras” ou da preposição “de” por “do”, testemunham seguinte a afirmação: “A principal causa dos erros de reconhecimento é a semelhança gráfica”[8] [TUMMARATTANANONT02].

1.4. Técnicas recentes de escavação ou mineração de texto

Têm sido diversas as abordagens relacionadas com a análise documental, tomando designações sugestivas como “Text Mining”, “Natural Language Processing”, “Knowledge Discovery”, “Automatic Text Retrieval”, etc. Estas e outras designações reflectem a escola ou domínios de conhecimento dos investigadores, bem como os meios de que dispõem. No decurso desta investigação, surgiram abordagens tanto nas vertentes da recuperação de informação (“Information Retrieval”) como da inteligência artificial (“Artificial Inteligence”).

À partida não se pretendeu optar por uma ou outra vertente de investigação. Esta ideia coincide com a visão que alguns investigadores procuram manter. Por exemplo, Dhillon e Modha [DHILLON00], cujos trabalhos estão inseridos num contexto de “Information Retrieval”, referem também como de interesse prático os algoritmos baseados em “machine learning”.

Todavia, ver-se-á, no capítulo 3 e seguintes, que a abordagem pela vertente da “Information Retrieval” se revela mais ajustada à filosofia de funcionamento do Indexing Service. Adicionalmente, o conjunto de soluções relativas à classificação automática de documentos, designadamente recorrendo à análise de clusters, também está claramente enquadrado neste domínio.

Na secção 1.3.3, “A linguagem e os erros”, referiu-se a necessidade de encontrar instrumentos que, procedendo a uma leitura detalhada dos documentos, identificassem erros de natureza sintáctica e semântica. As investigações em torno de soluções baseadas em redes neuronais tendem a corresponder rigorosamente a essa especificação. Assim, e por se tratar de uma matéria intrigante, inicia-se o próximo capítulo, examinando conceitos, arquitecturas e investigações que lhes estão associadas.

[7] V. Anexo III.
[8] Orign. “The main cause of recognition errors is graphical similarity.”

1.5. Conclusões do capítulo 1

A classificação de documentos e a correcção de erros são dois problemas radicalmente diferentes no que se refere ao detalhe com que os documentos são percepcionados. Portanto, perante os documentos, identificam-se duas atitudes diferentes a adoptar nas acções de classificação ou na correcção de erros. Esta diferença de atitude determina as características dos agentes que executarão quer uma quer outra actividade.

É também consequência e testemunho desta atitude dual face ao tratamento dos documentos, o desenvolvimento de duas ou mais vertentes de investigação. As técnicas que utilizam conceitos de redes neuronais estarão mais próximas da atitude do leitor atento aos detalhes. Por sua vez, as que utilizam técnicas de análise de clusters, estarão mais próximas de uma leitura rápida sem considerar os detalhes.

Capítulo 2. Técnicas de NLP[9] com redes neuronais

No contexto do presente trabalho e doravante, o significado de "processamento de linguagem natural", refere-se, *exclusivamente*, ao tratamento da linguagem na sua modalidade *escrita*. Está a falar-se de escavação ou mineração de texto. A expressão "processamento de linguagem natural" poderá ser, e é também legitimamente, utilizada com outro significado, designadamente o do tratamento da linguagem na sua modalidade falada ou sonora. Não é o caso.

Na secção 1.3.3 foi identificada "a necessidade de se encontrar ferramentas que *leiam* em detalhe os textos e confiram a sua obediência a padrões sintácticos e semânticos". Mais adiante, na secção 1.4, afirmou-se que as "investigações em torno de soluções baseadas em redes neuronais tendem a corresponder rigorosamente a essa especificação". Mais precisamente, aqui são procurados caminhos para soluções que visem a localização e correcção de erros ortográficos de forma mais ou menos automatizada.

As redes neuronais, como adiante se verá[10], podem ser empregues como solução para uma variedade muito ampla de problemas. A classificação de documentos, escritos em linguagem natural, poderá estar incluída nesse conjunto. Todavia tal aspecto, sendo importante, não ressaltou tão vincadamente para o fim que se pretende. Como se verá nos capítulos seguintes, há uma estreita ligação entre conceitos incluídos no domínio da "Information Retrieval" e as propriedades e métodos que os objectos de acesso ao "Indexing Service" disponibilizam. Esta conexão condicionou fortemente a investigação.

Simplificadamente, o processamento neuronal procura simular o funcionamento cerebral em alternativa ao processamento sequencial "introduzido por von Neumann" [HERTZ91: 1]. O processamento paralelo, a capacidade de adaptação, a tolerância nas falhas e as possibilidades de tratamento de informação probabilística e ruidosa, são algumas das importantes características cerebrais que se procura implementar com as redes neuronais [ibid.]. Este "campo" de investigação "é também conhecido como *neural networks*, *neurocomputation*, *associative networks*, *collective computation*, *connectionism*" [op. cit. p. 2].

Antes de prosseguir com a identificação e breve descrição de algumas investigações orientadas para o processamento de linguagem natural com redes neuronais, é necessário conhecer e precisar o significado de alguma terminologia empregue neste contexto. Esta atitude é necessária para a compreensão mínima de algumas obras incontornáveis sobre o processamento de linguagem natural. Na secção seguinte descrevem-se, de uma forma muito resumida, alguns aspectos e conceitos sobre redes neuronais, que se revelaram como essenciais.

[9] NLP significa "Natural Language Processing".
[10] Cf. Sec. 2.1.4

2.1. Redes neuronais, alguns breves apontamentos

Os neurónios cerebrais (Figura 1) são células nervosas constituídas por um *corpo* ou *soma*[11] que contém o *núcleo* [ibid.]. Da célula sai um único filamento ou axónio[12], que se prolonga e ramifica em labirintos de ligações a outras células [ibid.]. Por outro lado, da célula nervosa também saem *dendrites*[13], filamentos que igualmente se ramificam, funcionando como extensões do corpo da célula [ibid.].

As extremidades das ramificações do axónio de uma célula nervosa encontram-se com os corpos ou com as dendrites de outras células [op. cit. p. 3]. Estas áreas de encontro designam-se por *sinapses*[14] ou junções sinápticas [op. cit. p. 2].

A passagem seguinte descreve o processo de transmissão de sinal e prenuncia tanto um modelo de neurónio como as restrições temporais a considerar em qualquer desenho ou implementação [op. cit. p. 3]:

> «A transmissão de sinal de uma célula para outra, numa sinapse, é um processo químico complexo pelo qual são libertadas substâncias transmissoras específicas a partir do lado emissor da junção. O efeito é elevar ou baixar o potencial eléctrico no interior do corpo da célula receptora. Se este potencial atinge um limite, através do axónio é enviado um impulso ou *acção potencial* de intensidade e duração fixas. » (...) «a célula ”disparou“. O impulso alastra-se através da ramificação axónica até às junções sinápticas com outras células. Após o disparo, a célula terá que esperar algum tempo chamado *período refractário* antes de poder disparar novamente.»[15]

Esta descrição é baseada no modelo de neurónio, proposto em 1943 por McCullock e Pitts, um processador simples cuja saída apresenta dois estados, activo ou inactivo [ibid.]. A passagem de um estado a outro depende da soma do estado das entradas face a um limiar de activação. As entradas, correspondentes às junções sinápticas, têm ponderações associadas que podem ser activadoras se positivas, ou inibidoras se negativas [ibid.].

[11] Orign. “soma”, Soma em citologia é o mesmo que “CORPO CELULAR” [HOUAISS03b].
[12] Orign. “axon”, em português Axónio “ (anatomia) prolongamento único de uma célula nervosa (...) do Grego áksõn, Sinónimo cilindro-eixo, neuroaxónio” [HOUAISS02a].
[13] Orign. “dendrite”, Dendrite, dendrito em biologia, citologia e histologia “prolongamento dos neurónios especializado na recepção de estímulos” [HOUAISS03a].
[14] Orign. “synapses”, Sinapse, (da fisiologia) “local de contacto entre neurónios onde ocorre a transmissão de impulsos nervosos de uma célula para outra” [HOUAISS03].
[15] Orign. «The transmission of a signal from one cell to another at a synapse is a complex chemical process in which specific transmitter substances are released from the sending side of the junction. The effect is to raise or lower the electrical potential inside the body of the receiving cell. If this potential reaches a threshold, a pulse or action potential of fixed strength and duration is sent down the axon.» (…) «the cell has “fired”. The pulse branches out through the axonal arborization to synaptic junctions to other cells. After firing, the cell has to wait for a time called the refractory period before it can fire again.» [HERTZ91: 3].

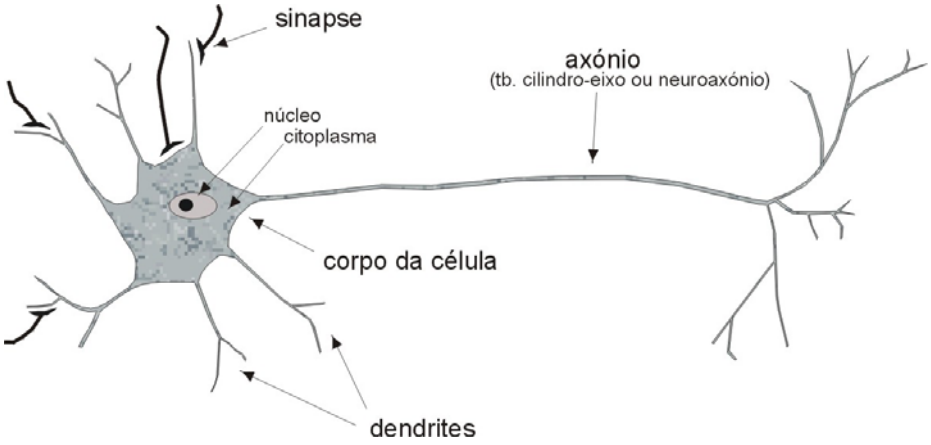

Figura 1 - Neurónio cerebral[16]

2.1.1. Um Modelo de neurónio

A Figura 2 representa um modelo de neurónio [HAYKIN: fig. 1.4]. Nesta representação distinguem-se três elementos essenciais: um conjunto de sinapses, com as ponderações ou pesos associados representando a sua importância relativa; um dispositivo somador; e, finalmente, uma função de activação destinada a limitar a amplitude da saída do neurónio [Ibid.]. No modelo da Figura 2, está representada uma entrada auxiliar Θ_k, também designada por "threshold"[17], que se destina a reduzir a entrada total na função de activação [Ibid.]. O "threshold" pode ser visto, alternativamente, como o negativo de "bias", como adiante se descreverá [ibid.].

Matematicamente, o neurónio *k* representado na Figura 2, pode ser descrito utilizando duas fórmulas relativamente simples [ibid.]:

$$u_k = \sum_{j=1}^{p} w_{kj} x_j$$

e

$$y_k = \varphi(u_k - \Theta_k),$$

em que x_1, x_2,..., x_p são os sinais de entrada; w_{k1}, w_{k2},..., w_{kp} são os pesos sinápticos do neurónio *k*; u_k é a saída que resulta da combinação linear; Θ_k é o limiar; $\varphi(\cdot)$ é a função de activação; e y_k é o sinal de saída do neurónio.

[16] Figura adaptada a partir da FIGURE 1.1 in [HERTZ91: 2].
[17] Tr. Limiar. O conceito de limiar está habitualmente muito associado à função degrau. No contexto do modelo de neurónio poderá não ter exactamente esse significado.

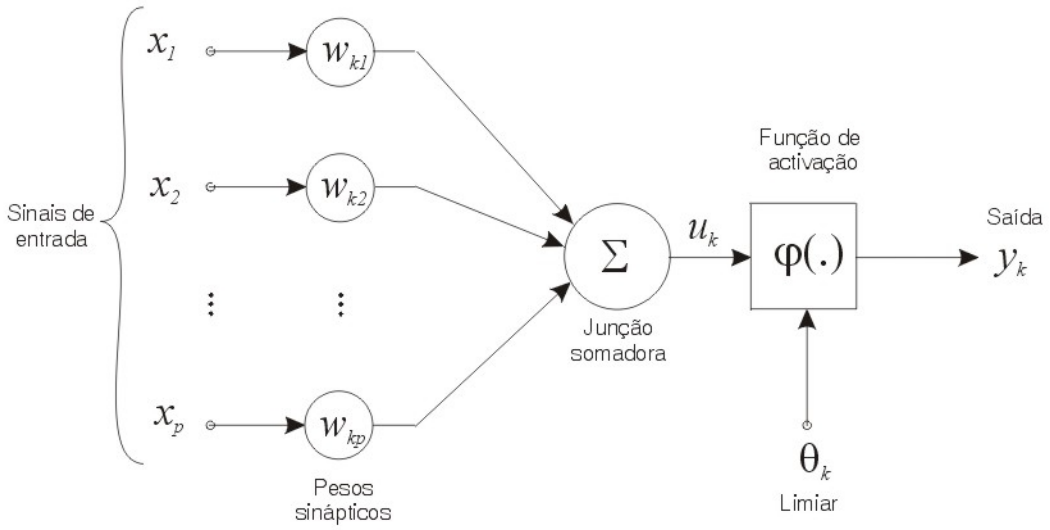

Figura 2 - Modelo de neurónio

Este modelo básico acomoda o limiar ou "threshold" como uma entrada bem distinta dos restantes sinais de entrada. Todavia [HAYKIN: 9-10] mostra que esta entrada pode ser agregada ao conjunto dos restantes de sinais de entrada, verificando-se as seguintes condições: o seu sinal de entrada é constante e igual a –1 e o peso associado é igual a Θ_k. Ou seja, removendo a entrada Θ_k, ao conjunto dos sinais de entrada representado na Figura 2, acrescentar-se-ia um sinal de entrada $x_0 = -1$ com peso sináptico$w_{k0} = \Theta_k$. Conforme atrás se referiu, o conceito de "bias" surge como o negativo de limiar ou "threshold" externo. Neste caso o sinal de entrada seria $x_0 = +1$ com um peso sináptico $w_{k0} = b_k$, em que b_k é a alimentação ou "bias" .

Um outro aspecto não menos importante prende-se com o significado e a forma da função de activação. A função de activação $\varphi(\cdot)$ define a saída do neurónio face ao nível de actividade presente na sua entrada [op. cit. p. 10]. Por exemplo, a função de activação $\varphi(\cdot)$ no modelo de McCullock-Pitts, conforme atrás se descreveu, é a função degrau unitário. Isto significa que a saída é representada por dois estados: activo ou inactivo. Este modelo é uma idealização muito simplificadora [HERTZ: 3]. Recebe esse nome como tributo à actividade pioneira, nesta área, desses investigadores.

No conjunto dos tipos básicos de função de activação, para além da função degrau unitário, são mais frequentemente consideradas as funções parcialmente linear e a sigmóide [HAYKIN: fig. 1.7]. Para ilustrar simplificadamente, a forma de cada uma destas funções é esboçada na Figura 3.

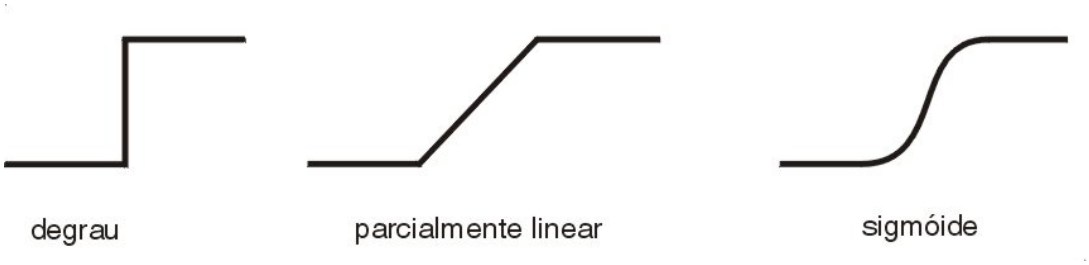

Figura 3 – Forma das funções de activação mais conhecidas

Qualquer uma destas duas últimas funções pode tender para uma função degrau [HAYKIN: 12]. Especialmente no caso da função sigmóide, a sua importância reside no facto de ser facilmente diferenciável, aspecto importante na teoria das redes neuronais [ibid.].

Para simplificar ainda mais o aspecto gráfico, é também usual substituir a representação do modelo sob a forma de diagrama de blocos semelhantes ao da Figura 2, por uma representação de grafos de fluxo de sinal [HAYKIN: 13]. Esta representação obedece a um conjunto de regras bem definidas, assegurando todos os detalhes funcionais do modelo [op. cit. sec. 1.4].

A descrição da dinâmica do neurónio considerando quer a implícita componente temporal quer a natureza do que se considera variável nas entradas e saídas, pode ser muito complexa [HERTZ91: 4]. Descrevem-se [ibid.] alguns aspectos que podem influir na maior ou menor complexidade dos modelos. No contexto do presente trabalho, essa variabilidade não é considerada. A simplicidade deste modelo deve ser, pois, considerada como uma longínqua aproximação à realidade biológica [op. cit. p. 5]. No entanto, e por isso mesmo, adapta-se bem ao desenho e implementação de redes com variadas topologias.

2.1.2. Algoritmos e paradigmas de aprendizagem

As redes neuronais aprendem ou desenvolvem representações de “conhecimento” a partir do meio que lhes é oferecido [HAYKIN: 45]. Uma rede neuronal faz a sua aprendizagem por um processo iterativo de ajustes dos pesos sinápticos [ibid.]. Após cada iteração, a rede torna-se mais “conhecedora” [ibid.].

O processo de aprendizagem é constituído por dois factores: um é a regra de ajuste dos pesos sinápticos; o outro, a forma como a rede se relaciona com o meio envolvente [HAYKIN: 46]. A regra de ajuste é também designada como *algoritmo (ou regra) de aprendizagem* [ibid.]. A forma como a rede se relaciona com o meio é conhecida por *paradigma de aprendizagem* [ibid.].

[HAYKIN] identifica quatro regras básicas de aprendizagem: “error-correction learning”, “Hebbian learning”, “competitive learning” e “Boltzman learning” [ibid.]. Por outro lado apresenta três paradigmas: “supervised learning”, "Reinforcement learning” e “Self-organized (unsupervised) learning” [ibid.].

Do conjunto de algoritmos de aprendizagem referidos, escolhe-se o "error-correction learning" como exemplo para ilustrar os conceitos relacionados com a natureza dos dados empregues no treino de redes neuronais. Além disso, é no "error-correction learning" que se baseia o "popular algoritmo conhecido como *error back-propagation*" [op. cit. p.138], que, mais adiante, será referido.

Em cada iteração *n* é apresentado um conjunto de dados de treino. Este conjunto é constituído por um vector de dados de entrada $\vec{x}(n)$ e um conjunto de dados de saída desejada, um vector $\vec{d}(n)$. Para o neurónio *k* a saída desejada seria $d_k(n)$. A diferença entre a saída resultante $Y_k(n)$ e o valor da saída desejada $d_k(n)$, permite calcular o erro ou seja $e_k(n) = d_k(n) - y_k(n)$ [op. cit. p. 47]. O erro, por sua vez, é utilizado para recalcular os parâmetros da rede, em regra os pesos ou ponderações sinápticas. O objectivo é minimizar uma "função de custo" baseada no sinal de erro $e_k(n)$, de tal modo que a saída do neurónio se aproxime da desejada [ibid.].

Cada conjunto de dados de entrada e de saída desejada é um exemplo que pode ser representado como um par ordenado $\left\{\vec{x}, \vec{d}\right\}$ [op. cit. p. 178 passim.]. Para treinar uma rede é-lhe apresentado um conjunto de exemplos assim constituído. O treino pode ser refinado, aplicando procedimentos de validação cruzada. O conjunto de exemplos é, por escolha aleatória, subdividido em dois: um conjunto de teste de outro de treino. O conjunto de treino, por sua vez, também subdividido em dois: um conjunto destinado a estimar (treinar) o modelo e outro (de validação) para validar o desempenho do modelo. Em regra, o conjunto de validação representa cerca de 10 a 20% do conjunto de treino [op. cit. p. 180]. Estes dois últimos conjuntos destinam-se a ajudar a escolher o modelo que apresenta o melhor desempenho. Uma vez escolhido o modelo nessas condições, utiliza-se o conjunto de treino completo para treinar a rede. Finalmente mede-se o desempenho em termos de generalização, submetendo a rede ao conjunto de teste [ibid.].

Neste contexto, define-se "época" como a apresentação completa do conjunto de dados de treino durante o processo de aprendizagem [HAYKIN: 151]. [JAIN00] refere-se a "uma travessia completa de dados de treino" [18] [n. 5 in op. cit. p. 20]. No mesmo sentido a este termo se referem [MAYBERRY03: 4; MAYBERRY94: 3; MIIKKULAINEN96: 4; LAWRENCE00: 13 passim.; BERG92: 5; BRYANT01: 6].

No que se refere ao paradigma de aprendizagem "supervised learning" ou "learning with a teacher" [HERTZ91: 89], pode afirmar-se que o "teacher" é alguém que dispõe do conhecimento sobre o ambiente representado por um conjunto de exemplos de entrada e saída [HAYKIN: 57]. Estando o "teacher" e a rede expostos ao mesmo vector de treino ou exemplo, é possível determinar o erro em cada iteração [ibid.]. A rede vai sendo ajustada, passo-a-passo, com o

[18] Orign. "One epoch means going through the entire training data once."

objectivo de emular o "teacher" [ibid.]. São exemplos de algoritmos de aprendizagem supervisionada a "lest-mean-square" (LMS) e o "back-propagation" (BP) [op. cit. p. 58].

Este algoritmo de aprendizagem supervisionada pode ser implementado de duas maneiras, conhecidas como "off-line" e "on-line" [op. cit. p. 59]. No caso da modalidade "off-line" a rede é afinada à parte, até executar o desempenho que se pretende. Concluído este ajuste o "design" é congelado e a rede lançada em funcionamento real em modo "static" [ibid.]. Na modalidade "on-line" a rede aprende em modo de operação real. Neste caso diz-se que a rede é dinâmica [ibid.].

O paradigma "reinforcement learning" procura ultrapassar algumas limitações do "supervised learning", designadamente no que se refere à adaptação a novas situações. A teoria do "reforço" é originada a partir de estudos experimentais da psicologia da aprendizagem animal [ibid.]. A ideia básica do "reinforcement learning" resume-se (in [HAYKIN: 59])[19] :

> «Se uma acção executada por sistema aprendiz é seguida por um estado geral de satisfação, então é reforçada a tendência do sistema para produzir essa acção em particular. No caso contrário, a tendência do sistema para produzir essa acção, é enfraquecida».

Um sistema implementando o paradigma "reinforcement learning" utiliza um agente crítico para obter a informação de reforço do ambiente [op. cit. p. 61]. Implementa também mecanismos que permitem o ajuste dos seus parâmetros por tentativa e erro [op. cit. p. 64].

No paradigma de aprendizagem "unsupervised" ou "self-organized", não há professor ou agente crítico externo para supervisionar o processo de aprendizagem [op. cit. p. 65]. Em substituição dos conjuntos de exemplos de treino acima referidos, é estabelecido um critério independente da tarefa, perante o qual se afere a qualidade da representação [ibid.]. Pode ser empregue uma regra de aprendizagem competitiva [ibid.]. Neste caso pode haver, por exemplo, uma camada de neurónios que competem uns com os outros (com regras bem determinadas) no sentido ganhar a oportunidade de corresponder às características representadas no padrão de entrada [ibid.].

2.1.3. Algumas arquitecturas de redes neuronais

As redes neuronais são essencialmente organizações de complexidade variável, compostas por unidades de processamento simples. A forma utilizada para estruturar os neurónios numa rede neuronal, está intimamente relacionada com o algoritmo de aprendizagem utilizado [HAYKIN: 18]. "As redes neuronais podem ser vistas como sistemas maciços de processamento paralelo constituído por um número extremamente elevado de processadores simples com muitas

[19] Haykin neste caso cita SUTTON, R. S.; et al. (1991) – "Reinforcement learning is direct adaptative optimal control."; **Proceedings of the American Control Conference**; pp. 2143-2146; Boston; M.A., e também BARTO, A.G. (1992) - "Reinforcement learning and adaptative critic methods." In **Handbook of Intelligent Control** (D.A. WHITE and D.A. SOFGE, eds.), pp. 469-491; Nova Iorque: Van Nostrand-Reinhold.

interligações” [20] [JAIN00: 6]. Dito de outra forma, o *poder* das redes neuronais assenta e varia com as diferentes arquitecturas com que são dispostas e acedidas as unidades de processamento. [ELMAN: 16-17] exerce claramente essa possibilidade quando, não satisfeito com a análise dos resultados das saídas, regista e analisa a evolução das representações internas da rede. Em resumo, tais arquitecturas podem depender do problema específico que se pretende resolver.

Algumas famílias de redes neuronais, mais frequentemente referidas na literatura - “Perceptron”, “Multi-Layer-Perceptron”, “Recurrent Back-Propagation”, “Hopfield” e “Kohonen Feature Map” - podem considerar-se exemplos clássicos, porque, no seu conjunto, reúnem conceitos essenciais. Estas famílias de redes distinguem-se pela forma como se caracterizam os conceitos de entrada e saída, o modo de propagação da activação, a sua topologia básica e a forma de aprendizagem [FRÖHLICH97].

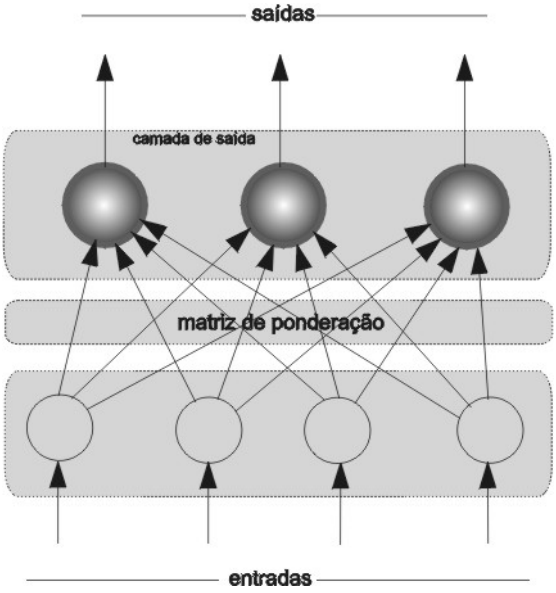

Figura 4 - Simple Perceptron[21]

O perceptron, de um nível, proposto por Rosenblatt em princípios da década de 60 [HERTZ91: 90], é a representação mais simples de uma rede neuronal (Figura 4). Também conhecido por *simple perceptron* [HERTZ91: cap. 5] só tem uma camada neuronal por definição. Mais precisamente, o processamento é executado apenas na camada de saída [HAYKIN: 18].

O sinal, ou activação, propaga-se num só sentido (*feed-forward*). Os neurónios de entrada têm ligações a todos os neurónios da saída. O paradigma de aprendizagem utilizado é o “supervised learning”.

O perceptron multi-layer representado na Figura 5, apresenta pelo menos mais uma camada intermédia. Pelo facto de este nível não estar em contacto directo com as entradas e

[20] Orign. “Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections.”
[21] Adaptação de diagrama idêntico patente em [FRÖLICH97].

saídas, é usual chamar-lhe camada escondida (“hidden-layer”) [HERTZ: 90]. O modo de aprendizagem torna-se mais complexo do que o anterior porque será necessário ajustar mais do que uma matriz de pesos sinápticos. No caso da Figura 5, seria necessário ajustar duas matrizes. É neste contexto que surge o conceito de “back-propagation”.

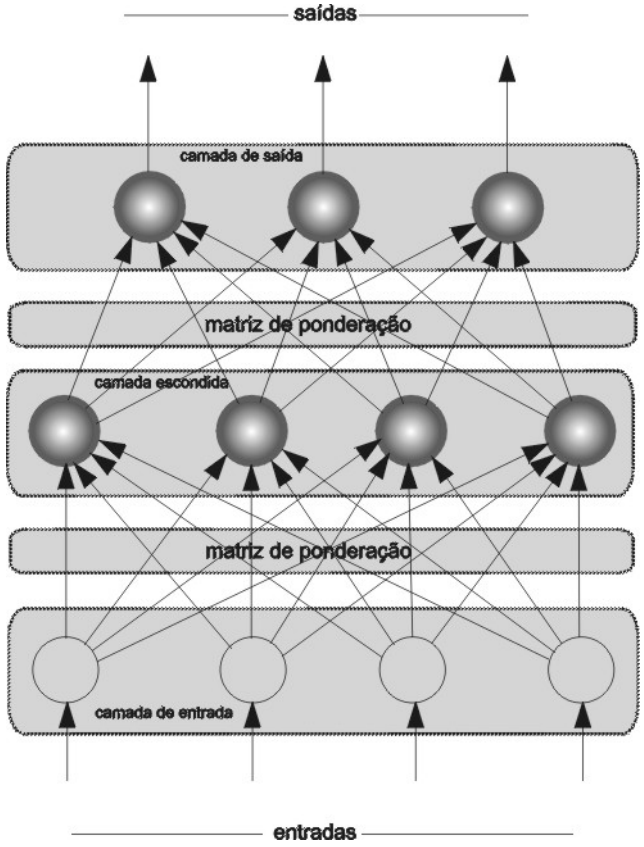

Figura 5 - Multi-Layer Perceptron[22]

As redes com *back-propagation* caracterizam-se por proporcionarem o cálculo do erro utilizando o diferencial da saída relativamente ao valor esperado em cada unidade de processamento. Determinado esse erro, é necessário encontrar uma forma de o fazer propagar para trás (“back-propagate”). Assim, o algoritmo “back-propagation” significa o processo encontrado que assegura as correcções necessárias em cada uma das matrizes de pesos sinápticos [HERTZ91: cap. 6]. As linhas a tracejado na Figura 6 ilustram essa ideia.

No contexto do algoritmo “back-propagation”, realça-se a importância de os neurónios apresentarem uma função de activação diferenciável, em regra do tipo sigmóide [HAYKIN: 138]. De resto, nos “perceptrons”, a necessidade de a função de activação “suavemente não linear”, i. e., diferenciável [ibid.], é evidenciada em oposição ao modelo originalmente proposto por Rosenblatt, que utiliza a função de activação do tipo degrau [op. cit. p. 118-120]. Destas

[22] Idem.

considerações resulta a mais habitual utilização de funções de activação do tipo parcialmente linear ou sigmóide em perceptrons [Cf. op. cit. caps. 5, 6 e passim.].

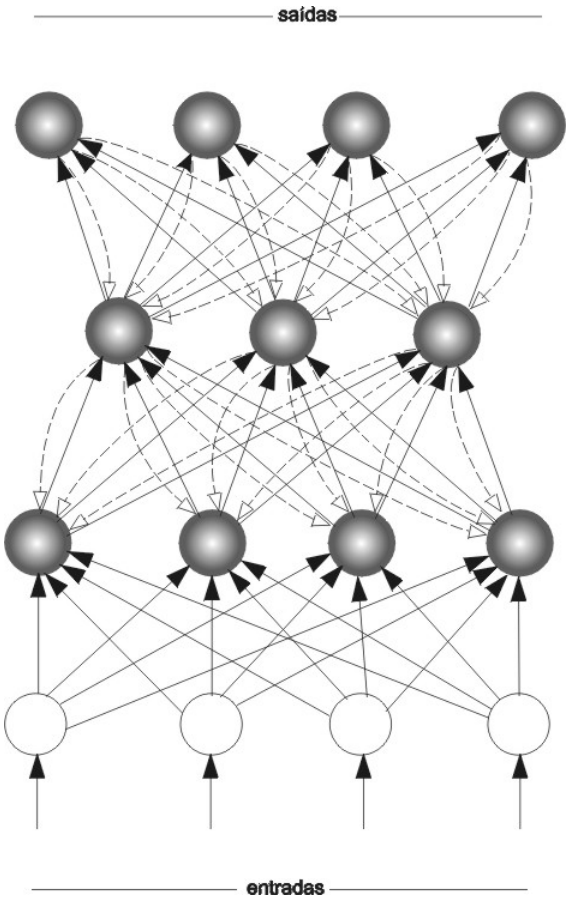

Figura 6 - Rede Multi-Layer com Back-Propagation[23]

Quando as unidades de processamento têm ligações simétricas ou bidireccionais e até mesmo lacetes de realimentação (*feedback*), diz-se que a rede é *recurrent* [HERTZ91: 163]. Quando "(…) a retropropagação pode ser alargada a redes arbitrárias, desde que convirjam para estados estáveis (...) o algoritmo é habitualmente intitulado de retropropagação recorrente" [24] [HERTZ91: 172]. Se as linhas representadas a tracejado na Figura 6 forem a implementação da simetria nas ligações existentes, então essa rede será designada por "recurrent back-propagation"[25].

Igualmente recorrentes são as redes Hopfield. A rede Hopfield procura "corporizar" um princípio da física que consiste em "guardar informação numa configuração dinamicamente estável" [HAYKIN: 285]. A «ideia», de Hopfield, «é guardar cada padrão no fundo de um "vale" de

[23] Idem.
[24] Orign. "(...) back-propagation can be extended to arbitrary networks as long as they converge to stable states (…) the algorithm is usually called recurrent back-propagation".

um campo de energia e seguidamente permitir que um procedimento dinâmico minimize a energia da rede de tal modo que o vale se torne o seu assento» [ibid.].

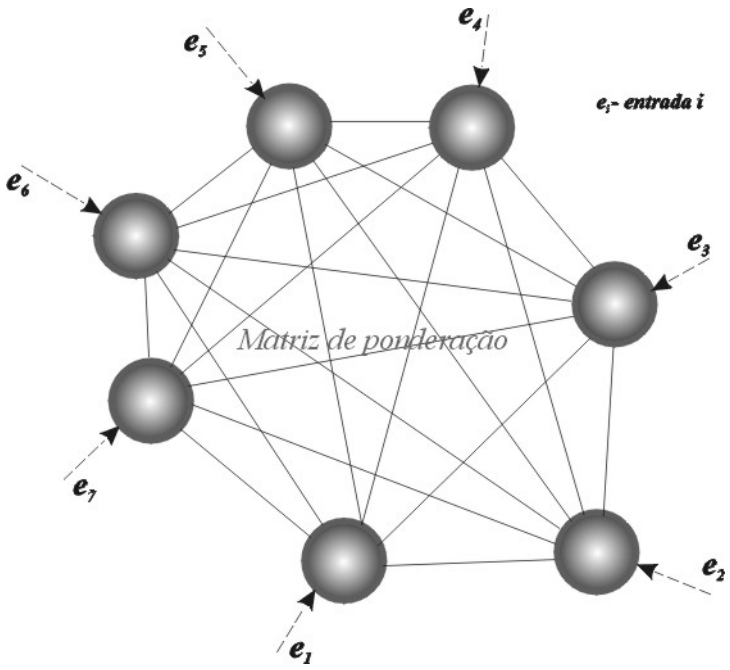

Figura 7 - Rede Hopfield[26]

As redes Hopfield são constituídas por unidades de processamento que funcionam como entradas e saídas (Figura 7). Cada unidade de processamento está ligada simetricamente a todas as outras. O método de aprendizagem não é supervisionado.

"A função de energia *E* é uma função monótona e decrescente do estado da rede $\{x_j | j=1, 2,..., N\}$" [HAYKIN: 287]. Define-se como espaço de estados o conjunto de estados que a rede pode assumir [op. cit. p. 288]. Os mínimos locais da função de energia representam os pontos de estabilidade no espaço de estados [ibid.]. Ou seja, este sistema converge: "No contexto do espaço (de configuração) os padrões armazenados (...) são atractores" [27] [HERTZ91: 12].

A rede Kohonen[28], representada na Figura 8, caracteriza-se essencialmente por proporcionar um mapa de características (*feature map*), que corresponde a uma espécie de auto-organização. Os neurónios desta rede estão dispostos nos nós de uma malha ou "lattice" [HAYKIN: 397]. Competindo entre si, os neurónios alteram as suas coordenadas na malha por forma a corresponderem às características do padrão de entrada [ibid.].

---

[25] Neste caso conserva-se a designação original para manter mais clara a identificação do conceito; todavia poderá ser eventualmente traduzida por "retropropagação recorrente".
[26] Adaptação de diagrama idêntico patente em [FRÖLICH97].
[27] Orign. "Within that (configuration) space the stored patterns (…) are attractors"
[28] Num simpático site intitulado "Neural Networks with Java", [FRÖLICH97] oferece uma síntese objectiva e acessível sobre redes neuronais. Também apresenta um "applet" que ilustra uma rede Kohonen, com alguma qualidade gráfica, o qual permite experimentar e intuir os conceitos envolvidos em torno desta temática.

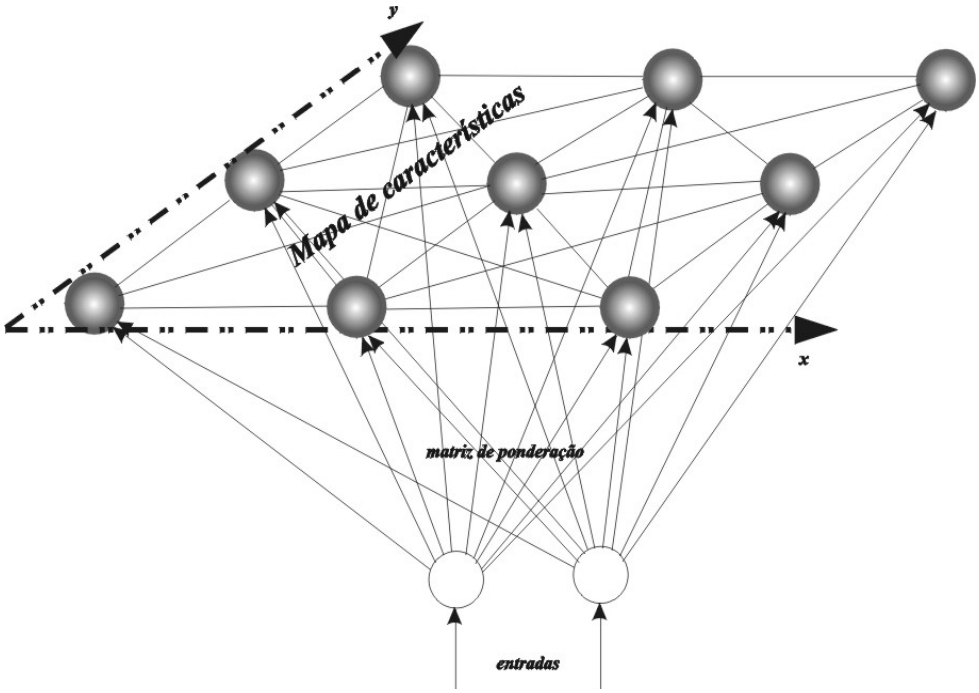

Figura 8 - Rede Kohonen

2.1.4. Aplicações habituais das redes neuronais

Topologia	Descrição da natureza da aplicação
simple perceptron	lógica simples
multi-layer perceptron	operações lógicas mais complexas classificação de padrões
recurrent back-propagation	operações lógicas mais complexas classificação de padrões análise de voz
Hopfield	associação de padrões problemas de optimização
Kohonen	associação de padrões problemas de optimização

Tabela 1- Aplicações de redes neuronais (Cf. [FRÖLICH97])

As redes neuronais têm sido aplicadas em variadas situações e especialmente quando os problemas não podem ser resolvidos por algoritmos convencionais [FRÖLICH97]. São habitualmente empregues na solução de problemas de classificação e optimização [ibid.] Os problemas endereçados pelas redes neuronais, habitualmente são [ibid.]:

> "associação de padrões, classificação de padrões, detecção de regularidades, processamento de imagem, análise de voz, problemas de optimização, controlo de robots, processamento de dados incorrectos ou incompletos, garantia de qualidade, previsão de mercados accionistas, simulação".

Cada topologia tem as suas aplicações principais. Na Tabela 1- Aplicações de redes neuronais, reúnem-se alguns dos campos de aplicação das topologias descritas [ibid.].

2.2. Modelos com representação de linguagem natural

Nestes modelos clássicos, as redes neuronais são essencialmente instrumentos de processamento numérico. Como acima se descreveu, cada exemplo de treino é, no caso geral, um par ordenado constituído por dois vectores. Além disso, a rede dispõe de matrizes de pesos sinápticos. Quer isto dizer que todos os elementos referidos são representações ou conjuntos de representações numéricas sejam internas ou externas.

Por outro lado, a arquitectura das redes neuronais escolhida apresenta um número de entradas fixo, dificultando a representação das palavras com comprimento variável. É necessário encontrar formas de confinar a riqueza e a elasticidade da linguagem natural, à rígida formatação de entrada exigida pelas redes neuronais, nas topologias exemplificadas.

2.2.1. Simple Recurrent Network

A SRN de Elman surge num contexto de pesquisa de uma representação do tempo em redes neuronais, partindo da reconhecida ideia de que "o tempo é claramente importante na cognição" [29] [ELMAN90: 2]. A linguagem é um comportamento que se exprime como uma sequência temporal [ibid.]. Mas, por um lado, os teóricos da linguística "talvez" tendam a "preocupar-se menos com a representação e processamento dos aspectos temporais das expressões" [ibid.]. Além disso, e por outro lado, os "modelos de processamento paralelo" tentam representar o tempo como se de uma grandeza espacial se tratasse [ibid.]. Elman procura, portanto, uma representação implícita do tempo, i. e., o efeito que o tempo tem sobre o processamento, em oposição à sua representação explícita, i. e., uma dimensão adicional da entrada [ibid.].

Partindo do requisito de que "a rede deve ser provida de memória" [30] [op. cit. p. 3], propõe a "Simple Recurrent Network", abreviadamente SRN. Considera uma rede de três camadas e coloca uma segunda camada escondida ao nível da entrada, intitulada "context units" (Figura 9). O facto de se localizar ao nível da camada de entrada, não significa que deixe de ser uma camada escondida no contexto da disposição das camadas, considerando o fluxo da activação. A SRN é

[29] Orign. "time is clearly important in cognition".
[30] Orign. "network must be given memory".

uma rede “multi-layer”, semelhante à da Figura 5, tipicamente “feed-forward”, em que a camada escondida acaba por ter como entradas, a camada de entrada e a “de contexto”.

Esta camada “de contexto” recebe, em cada iteração, a informação de activação da camada escondida e conserva-a até ao início da iteração seguinte. Há uma correspondência de um para um no sentido da camada escondida para a camada “de contexto”. No sentido contrário todas as unidades “de contexto” activam todas as unidades escondidas. Nestas ligações as ponderações são fixas e de valor unitário. O método de treino utilizado é supervisionado ou “with a teacher” e usa-se “back-propagation” para ajustar incrementalmente as ponderações das ligações [op. cit. p. 5].

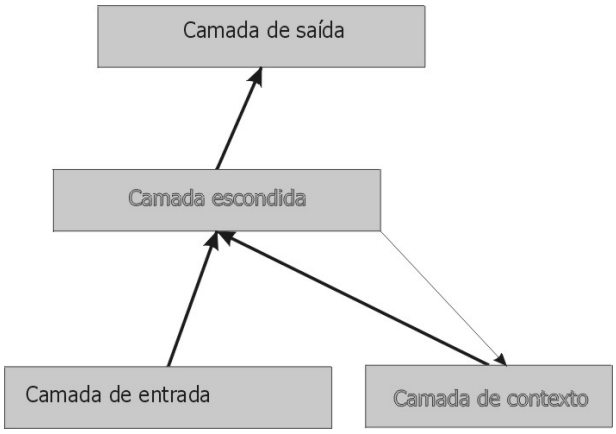

Figura 9 - Simple Recurrent Network (SRN)

A camada escondida terá, portanto, que calcular o resultado correcto tendo em conta quer o valor do padrão de entrada, quer o estado anterior, residente na camada “de contexto”. É assim que “estas unidades de contexto fornecem a memória à rede” [31] [ibid.].

Com este modelo básico [op. cit. p. 4], apresenta algumas simulações e os respectivos resultados [op. cit. p. 5 sgg.]. As variações do modelo dependem do que se considera ser, em cada caso, a unidade de padrão de entrada.

O caso da simulação do problema “OU exclusivo” ou XOR, ajuda a ilustrar o funcionamento do modelo e a metodologia empregue quer na construção dos conjuntos de treino, quer na análise que o autor faz dos resultados conseguidos. Como é sabido, o resultado da operação XOR sobre os pares (0, 0) e (1, 1) é igual a 0, e sobre os pares (1, 0) e (0, 1) é igual a 1. Mostra-se [HAYKIN: 157 sgg.], que este problema pode ser resolvido utilizando uma rede de duas camadas e com o algoritmo de “back-propagation” [32]. Verifica-se, portanto, uma diferença evidente entre esta solução e a que emprega a SRN: o padrão de entrada é constituído por dois bits em cada iteração utilizando a rede “multi-layer”, enquanto que, utilizando a SRN, apenas por um.

[31] Orign. ”“these context units thus provide the network with memory”.
[32] Elman refere-se a este aspecto, dizendo que a solução deste problema “requires at least three-layers”. Interpreta-se esta diferença de três para duas camadas, como proveniente de diferentes critérios de contagem das camadas neuronais.

No caso da simulação do problema XOR, a rede SRN de Elman é constituída por uma unidade de processamento (um bit) nas camadas de entrada e saída, e duas unidades de processamento (2 bits) nas camadas escondidas e de contexto. Depois de treinada, a rede procurará entregar na saída, em cada iteração, o resultado do XOR sobre os bits anterior e actual.

Construiu-se uma sequência de bits [ELMAN: 5]: 1 0 1 0 0 0 0 1 1 1 1 0 1 0 1 ... . A construção desta sequência tem uma regra: aplicando-se a operação XOR sobre o primeiro e o segundo bits, e obtém-se o terceiro; volta a aplicar-se a operação XOR sobre o quarto e o quinto, obtendo-se o sexto; e por aí fora [ibid.]. A sequência criada desta forma tinha 3000 bits. A rede foi treinada no sentido de prever o bit seguinte na sequência [ibid.]. Ou seja, o conjunto de exemplos sob a forma do par (x,d), assemelha-se ao conjunto [op. cit. p. 6]:

$$\{(1,0),(0,1),(1,0),(0,0),(0,0),(0,0),(0,1),(1,1),(1,1),(1,1),(1,0),(0,1),(1,0),(0,1),(1,?),\cdots\}.$$

Ao fim de 600 passagens desta sequência de 3000 bits, a resposta da rede aproximou-se da desejada [ibid.].

Finalmente, é a partir da análise da evolução do erro médio quadrático, resultante da diferença entre a saída desejada e a efectiva, que Elman retira algumas conclusões. Neste caso particular mostra que o erro é mínimo a intervalos de 3 iterações, ou seja, quando é possível fazer uma previsão correcta da entrada seguinte [ibid.].

E é essencialmente dentro desta sequência de acontecimentos, que se desenvolvem as restantes simulações: construção de um padrão de entrada e um conjunto de treino com regras bem definidas; treino da rede com um conjunto de exemplos criado na perspectiva de que a saída desejada represente a previsão do padrão seguinte na entrada; teste e análise da evolução do erro resultante.

Nas regras de construção dos padrões de entrada e dos conjuntos de treino, Elman procura reflectir diferentes aspectos que se colocam na abordagem do processamento de linguagem natural, e na perspectiva do desenrolar dos acontecimentos linguísticos no tempo. A SRN é submetida a simulações, num crescendo de complexidade de padrões sequenciais [op. cit. p. 7 sgg.]. Todavia reconhecem-se fortes limitações relacionadas com a dimensão das sequências de entrada e a sua variabilidade [ibid.].

No caso anterior, o padrão de entrada e de saída desejada era constituído apenas por um bit. Para representar letras e palavras, i. e., estruturas mais complexas, foram constituídos padrões ou vectores de bits a apresentar em cada iteração. Ou seja, cada elemento do par (x,d) passou a ser constituído por um vector. O número de unidades de processamento nas camadas de entrada e de saída, acompanharam o número de bits em cada vector. Também variou o número de unidades de processamento nas camadas escondida e "de contexto". No parágrafo seguinte referem-se padrões com 31 bits. A rede utilizada, neste caso, tinha 31 unidades de

processamento nas camadas de entrada e de saída, e 150 nas camadas escondida e "de contexto".

A construção destes conjuntos de vectores obedeceu a regras muito precisas e limitadas, por forma a assegurar o cumprimento das premissas de cada ensaio. Também limitado foi o número de letras ou palavras admitidas. Por exemplo, foi criado um pequeno dicionário de 29 palavras, em que cada palavra estava associada a um bit diferente num vector de 31 bits, sobrando duas posições. Desta forma procurou-se assegurar a ortonormalidade dos padrões entre si [op. cit. p. 15]. Neste curto glossário também se categorizou sintacticamente cada termo [op. cit. p. 14]. Nestas condições, os padrões de teste apresentados foram necessariamente limitados ao problema específico cuja sequência temporal se procurou estudar.

2.2.2. Memória Auto Associativa e Recursiva - RAAM[33]

Um dos problemas que se tem colocado frequentemente na aplicação de redes neuronais a actividades cognitivas de mais "alto nível", tais como o Processamento de Linguagem Natural, tem sido a "inadequação das suas representações" [POLLACK90: 2]. As "estruturas simbólicas tradicionalmente utilizadas em Inteligência Artificial", que apresentam "dimensão variável" e necessitam de "alocação dinâmica" [ibid.], não são directamente representadas em redes neuronais [ibid.]. A dificuldade em representar a combinação de elementos em estruturas sintácticas que, por sua vez, resultem num todo com algum valor significativo ou semântico, tem sido utilizada como um dos argumentos invocados contra a utilização de redes neuronais [ibid.].

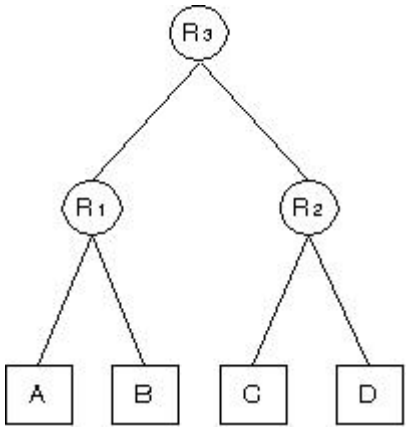

Figura 10 - Exemplo de uma árvore binária

É neste contexto que se apresenta [POLLACK90: 6 sgg.] um modelo em que a representação de sequências simbólicas de dimensão variável é reduzida a uma forma numérica de dimensão fixa. Este modelo é constituído por dois componentes básicos, o compressor e o

reconstrutor. O compressor reduz grupos de pequenos conjuntos de padrões de dimensão fixa codificando-os num só padrão com a mesma dimensão. O reconstrutor executa o processo inverso. Se os padrões a agrupar forem constituídos por *m* bits e estiverem dispostos segundo os ramos de uma "binary tree", o compressor pode ser implementado com uma rede de um nível com *2m* entradas e *m* saídas. Disposição inversa terá o reconstrutor.

Exemplificando o conceito [op. cit. p. 6.], considere-se a árvore binária representada na Figura 10, em que os terminais A, B, C e D são padrões de dimensão fixa [ibid.]. Estes padrões são codificados recursivamente em três etapas. Em primeiro lugar R_1 é o resultado da compressão de A e B. Seguidamente R_2 é o resultado da compressão de C e D. Finalmente R_1 e R_2, após a compressão, resultam em R_3 [ibid.]. Em sentido contrário, a reconstrução, consistirá em reproduzir recursivamente os padrões originais a partir do padrão R_3, invertendo o sentido dos passos descritos [ibid.].

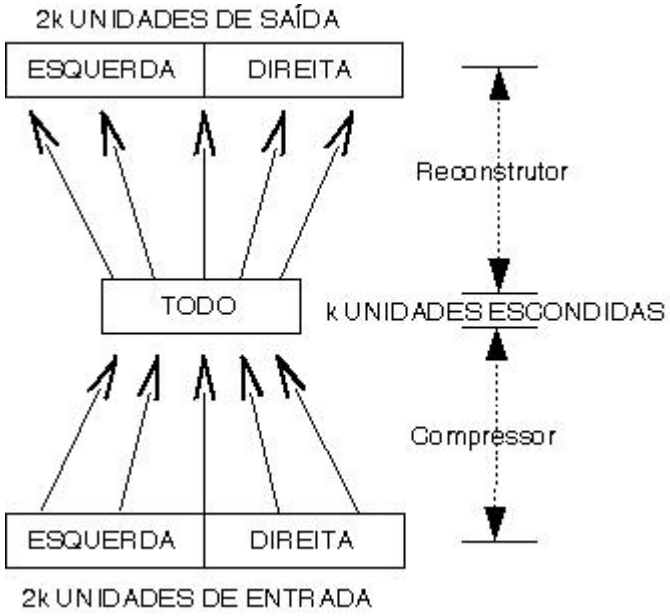

Figura 11 - Rede simples reunindo o compressor e o reconstrutor

Na Figura 11 estão reunidos os conceitos de compressor e reconstrutor, bem como uma singela representação do funcionamento descrito [op. cit. fig. 3]. Por outro lado a Figura 11 também se assemelha muito a uma rede "multi-layer" com três camadas. De facto a implementação avançada [op. cit. p. 8], consiste numa rede semelhante à da Figura 6, uma rede "multi-layer" com "back-propagation". Facilmente se observa que é na camada escondida que se acolhe a representação comprimida [ibid.]. A rede é treinada em modo autoassociativo, ou seja, o padrão de saída desejada é igual ao padrão de entrada [ibid.].

[33] RAAM significa "Recurrent AutoAssociative Memory".

Neste caso, *recursividade* significa que o padrão resultante de cada compressão de ramos de uma árvore binária, R_1, será utilizado como entrada na compressão de nível superior [ibid.]. Isto significa que este padrão será guardado até à criação do seu par, R_2, uma vez que resultam de processamentos separados [op. cit. p.10]. Em qualquer implementação é necessário considerar a existência de um "stack" de memória para guardar padrões a processar.

No sentido de evitar a necessidade do referido armazenamento adicional, a arquitectura pode ser modificada por forma a admitir uma lógica da sequência de acesso do tipo "Last-In- First-Out" [ibid.]. Neste caso, o padrão resultante funciona como "stack" [ibid.], emparelhando com o padrão seguinte no processamento da nova compressão e assim sucessivamente [ibid.]. Esta modalidade recebe o nome de "Sequential RAAM" [ibid.].

2.2.3. Representações com associação a vectores

O problema da uniformização do comprimento das representações de palavras aparece resolvido em [MIIKKULAINEN90] de outra forma. Na arquitectura de rede que apresenta há um dicionário ("Lexicon") em que cada palavra está associada a um vector de números reais entre *0* e *1*. Na representação gráfica, visualmente sugestiva (Figura 12), os números reais são representados por níveis de cinzento.

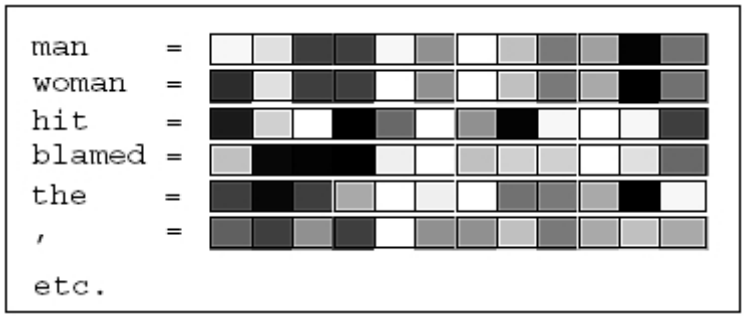

Figura 12 – Léxico [MIIKKULAINEN90: fig. 2]

O "Lexicon" em [MIIKKULAINEN90: 2] acaba por ser, no mínimo, uma tabela com dois atributos em que um deles é obrigatoriamente um vector representando um padrão de entrada ou de saída numa rede neuronal. O outro atributo, no caso específico do "Lexicon" representa cadeias de caracteres com significado no contexto das frases, ou seja, em regra, palavras.

As representações utilizadas e desenvolvidas são padrões. Esse aspecto, testemunhado na Figura 12 no que respeita a representações de palavras, não se fica por aqui. O mesmo autor apresenta uma variedade de trabalhos em que faz associar padrões a frases completas ou fragmentos divididos por sintagmas [MIIKKULAINEN96; MIIKKULAINEN97]. Também

[BRYANT01][34] e [MAYBERRY03][35] apresentam a mesma tendência de utilizar padrões para representar frases e expressões com valor semântico.

2.3. Na senda das representações semânticas e gramaticais

Verifica-se que há muita actividade de investigação em torno das diversos aspectos de análise linguística. As implementações com redes neuronais, apesar dos constrangimentos específicos que adicionam mais interrogações e dúvidas, também sugerem novas visões sobre a abordagem das questões linguísticas [ALLEN99].

Nos artigos [ELMAN90] e [POLLACK90] os aspectos relacionados com o processamento de linguagem natural poderão ser considerados como acessórios. O aspecto principal em qualquer dos trabalhos é a apresentação do modelo respectivo. Em [ELMAN90] centra-se a análise do modelo de SRN exposto nas representações desenvolvidas internamente pela rede, demonstrando que, neste modelo, o tempo ajuda a "contextualizar" as representações que o modelo recebe. Num dos exemplos de aplicação, mostra-se como é possível treinar uma SRN para classificar palavras entre verbos, substantivos e outras subclasses. Nesse caso são utilizadas sequências de palavras, tendo como objectivo conseguir que a rede preveja a classificação da palavra seguinte. Pode-se estabelecer o mesmo tipo de considerações sobre [POLLACK90].

Estas duas arquitecturas aparecem como componentes, total ou parcialmente, na constituição de arquitecturas mais complexas que visam o Processamento de Linguagem Natural. Por exemplo, [MIIKKULAINEN97] baseia a sua arquitectura "Subsymbolic case-role assignment of simple sentences" numa SRN [op. cit. p. 4]. Mais adiante, quando se trata de analisar frases com orações subordinadas, adiciona um "stack" a partir de uma RAAM [op. cit. p.10].

Aferir da "boa formação das expressões" [ALLEN99: 3], "classificar frases em linguagem natural como gramaticais ou não gramaticais" [LAWRENCE00: 1], é uma tendência geral de investigação onde estas - SRN e RAAM - e outras arquitecturas são utilizadas. Trata-se de procurar representações para frases a nível sintáctico e semântico.

No contexto do presente trabalho, o registo de algumas arquitecturas de redes neuronais que se inserem nesta tendência para procurar representações sintácticas e semânticas de frases, poderá parecer desajustado na perspectiva da localização de erros. Contudo, os erros resultantes da aquisição por OCR, como acima se referiu, nem sempre se traduzem simplesmente em erros ortográficos. Podem traduzir-se em trocas, por confusão gráfica, de palavras que, por sua vez, resultam em deficientes construções sintácticas e desajustes no emprego de termos, no contexto semântico da frase.

[34] Com Risto Miikkulainen.
[35] Idem.

2.4. Conclusões do capítulo 2

As redes neuronais, procurando simular o funcionamento cerebral, podem ser empregues como solução para uma variedade muito grande de problemas. Tais redes são constituídas por unidades de processamento simples que modelam o neurónio cerebral, podendo ser dispostas em variadas configurações. Aprendem, ou desenvolvem representações de conhecimento, por ajustes iterativos dos seus pesos sinápticos. A aprendizagem é condicionada por um algoritmo, ou regra de ajuste dos pesos sinápticos, e um paradigma que reflecte a forma como a rede se relaciona com o meio.

Orientadas para o processamento de linguagem natural, têm sido propostas diversas arquitecturas e investigações, utilizando redes neuronais. Destacam-se, desse conjunto, a SRN e a RAAM como elementos básicos de arquitecturas mais complexas. A criação da capacidade de generalizar a análise gramatical e semântica, uma tendência com alguma expressão no domínio das redes neuronais, seria um contributo importante para o encontro de soluções visando a correcção de erros. Persistem, no entanto, dificuldades relacionadas com a representação de palavras e frases.

Capítulo 3. Técnicas de Recuperação de Informação

O conceito genérico de *Information Retrieval* tem evoluído muito nas últimas três décadas. Esta evolução está relacionada com a disponibilidade crescente de documentos, em variados formatos, nos dispositivos de armazenamento dos computadores [WEIDE01]. As reflexões, em torno deste conceito, podem ser muito profundas e alargadas.

> "(...) do ponto de vista do armazenamento de informação, a breve história da recuperação de informação começou com a pesquisa de dados e termina com a pesquisa de informação multimédia" [36] [WEIDE01: 3].

O autor, neste trecho, finaliza uma breve discussão em torno da distinção entre "Data Retrieval" e "Information Retrieval". *Data Retrieval* considera ter a ver com a pesquisa de dados estruturados, "factos gravados em sistemas de base de dados" [WEIDE01: 2]. O que foi gravada foi informação *sobre* os documentos e não a *contida* nos documentos em si mesmo [Ibid.].

Desta forma se procura salientar que Information Retrieval é um conceito sob o qual se abriga o tratamento e pesquisa de informação residente em documentos pouco ou nada estruturados. A propósito, [FRAKES92] afirma no seu prefácio "IR (...) é frequentemente confundida com DBMS - um campo com o qual partilha preocupações e, contudo, diferente" [FRAKES92: vii].

Information Retrieval segundo [WEIDE01: 4-5] está balizada entre dois aspectos. Por um lado estão armazenados *objectos de informação*, pouco ou nada estruturada, e por outro há *necessidades de informação,* materializadas pela *formulação* de *consultas*. Se o armazenamento à partida não exige especiais tratamentos, o mesmo não se passa com a resposta às consultas. Esta última asserção determina a atitude que o sistema de *Information Retrieval* terá face ao armazenamento.

3.1. Sistemas de Information Retrieval

Situado entre o aspecto das necessidades e o dos objectos de informação, está o Sistema de *Information Retrieval*. Relativamente aos objectos de informação, a sua acção consiste em os caracterizar ou, mais simplesmente, indexá-los. Na vertente da resposta às consultas, procurará apresentar os documentos mais relevantes face aos termos da consulta formulada.

[36] Orign. "from the point of view of information storage, the brief history of information retrieval started with data retrieval and ends with multi-media information retrieval" [WEIDE01: 3].

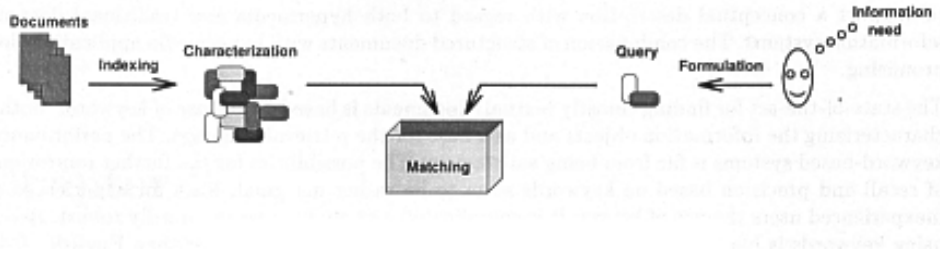

Figura 13 - Paradigma da recuperação de Informação (in [WEIDE01])

De uma forma sistemática, [FRAKES92] sugere que os sistemas de recuperação de informação[37], também designados por "information retrieval" ou simplesmente IR, podem ser analisados segundo um enquadramento que os classifica em seis facetas básicas [FRAKES92: cap. 1]:

- modelo conceptual
- estrutura de ficheiros
- operações de consulta
- operações com termos
- operações com documentos
- hardware

Cada uma destas facetas tendo associados termos que descrevem sucintamente conceitos e vocabulários que podem tomar, pode ser considerada como um ponto de decisão no projecto, exploração e desenvolvimento de sistemas de pesquisa de informação e análise documental. Para ilustrar, transcreve-se a tabela [FRAKES92: tab. 1.1].

Conceptual Model	**File Structure**	**Query Operations**	**Term Operations**	**Document Operations**	**Hardware**
Boolean	Flat File	Feedback	Stem	Parse	vonNeumann
Extended Boolean	Inverted File	Parse	Weight	Display	Parallel
Probabilistic	Signature	Boolean	Thesaurus	Cluster	IR Specific
String Search	Pat Trees	Cluster	Stoplist	Rank	Optical Disk
Vector Space	Graphs		Truncation	Sort	Mag. Disk
	Hashing			Field Mask	
				Assign IDs	

Tabela 2 - Esquema de classificação de "Information Retrieval", in [FRAKES92]

A Tabela 2 procura ser exaustiva na reunião de conceitos e termos que estão envolvidos neste domínio. Por esse motivo dá uma ideia clara sobre o número de estruturas de dados,

[37] Mais precisamente no contexto da organização da obra citada [FRAKES92] . Sendo uma importante e relativamente actual obra de referência, a sua organização reúne as principais tendências do desenvolvimento de sistemas de recuperação de informação.

algoritmos e técnicas abordadas na obra onde se insere [FRAKES92]. Perante um cenário tão vasto, no contexto do presente trabalho apenas são examinados alguns destes aspectos.

Na orientação para a classificação automática de documentos, por exemplo, são de destacar os termos que significam a possibilidade de agregar documentos e conceitos. Assim, expressões que significam “agregação de documentos”, “busca de termos mais frequentes em documentos”, “representação de documentos em espaço vectorial” serão pontos de decisão obrigatórios. Ou seja, e partindo da tabela anterior, identificam-se neste contexto os termos *cluster*, *boolean* e *vector space* como os vocábulos mais significativos.

A perspectiva da detecção de erros ortográficos pressupõe uma reflexão em torno do significado de *Inverted Files*, *Stem*, *Thesaurus*, *Stop list*, entre outros. São expressões associadas ao tratamento dos termos. *Inverted Files* são estruturas de dados que reúnem termos, permitindo um conjunto de operações de associação a documentos.

Os parágrafos anteriores demonstram que não se devem considerar estanques as facetas identificadas. Frakes afirma “o problema com estas taxinomias é que as categorias não são mutuamente exclusivas” [FRAKES92: 3].

3.1.1. Fluxo de Informação

O paradigma de IR, ilustrado na Figura 13, evidencia alguma simetria entre o processo de colheita de características para indexação, e o processo de estabelecimento ou formulação de consultas. No diagrama apresentado na Figura 14 reconhece-se essa simetria com alguma dificuldade. Todavia, examinando os percursos de informação, verifica-se que, tanto a recolha destinada à indexação como a originada pelas consultas, atravessam processos com características idênticas e dispostos de forma simétrica.

Partindo do interface, a consulta (*query*) é remetida a um processo designado por *parse query*. Repetindo o mesmo exercício a partir dos documentos, verifica-se que o texto é lançado num processo designado por *break into words*, logo seguido da remoção das palavras *ruidosas*.

Ambos os percursos são uma análise lexical [FRAKES92: cap7]. Recebem sequências de caracteres e convertem-nas em palavras ou, mais precisamente, em *tokens*, na Figura 14 designados por “Query terms” ou “non-stoplist words”. Estes “tokens” são entregues a processos designados por *stemming*. No caso da consulta, as “stemmed words” são entregues directamente para processamento pela base de dados. Ignorando a presença do processo *term weighting*, o mesmo se passa no caso do processamento dos documentos.

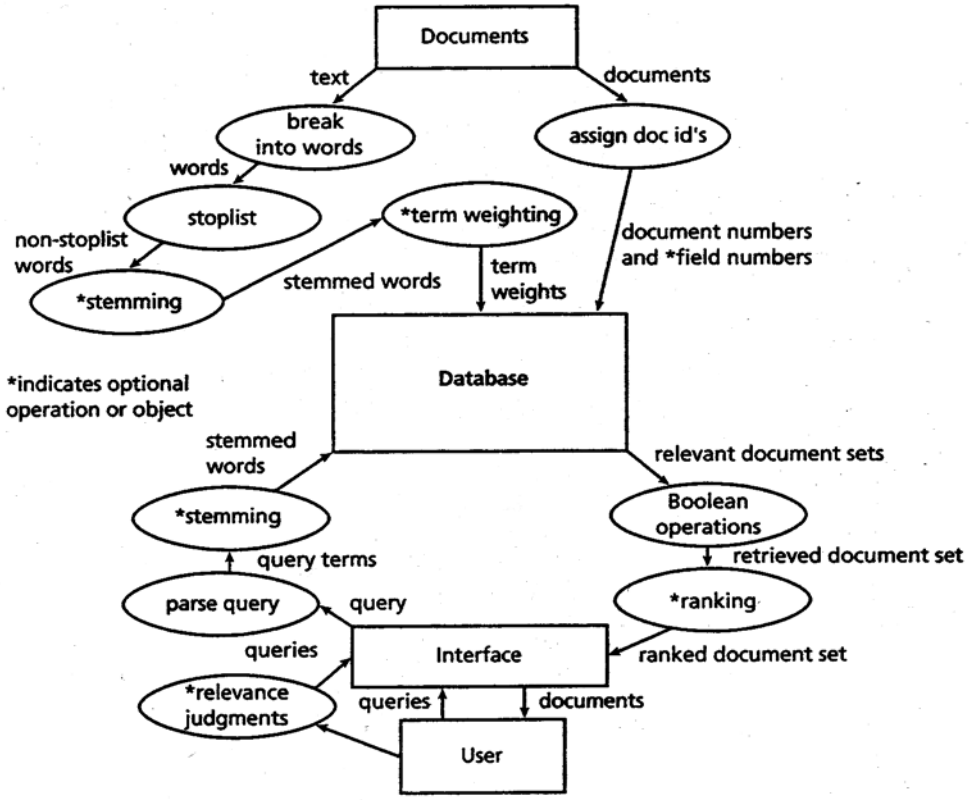

Figura 14 - Diagrama de um sistema booleano de IR (in [FRAKES92])

3.1.2. Tratamento de termos

O tratamento de termos passa por processos que visam, sobretudo, reduzir a dimensão do sistema de indexação, aumentar o desempenho geral do sistema de IR e, em especial, das consultas. Embora habitualmente com designações sugestivas relativamente aos seus objectivos, é sempre conveniente examinar o seu significado e inserção.

A análise lexical consiste "na conversão de uma cadeia de caracteres de entrada numa cadeia palavras ou *tokens*" [FRAKES92: 102]. Pode ser simplesmente a implementação da separação de palavras localizadas entre caracteres delimitadores. Este conceito abarca também alguma actuação sobre as palavras resultantes. Num contexto de análise da cadeia de caracteres lançados na execução de uma consulta, a análise lexical proporciona a distinção entre operadores, indicadores de agrupamento e palavras [FRAKES92: 104]. Este processo também identifica ou omite as palavras que reconhece como pertencentes à *stoplist*.

Stoplist, dicionário negativo ou listas de "noise words" ou "noise lists", constituem conjunto de palavras que serão omitidas tanto na indexação com nas acções de pesquisa de informação. Habitualmente considera-se que são palavras desnecessárias num contexto de diferenciação entre documentos [FRAKES92: 113]. Conjuntos de palavras como as referidas na secção 1.3.2, "O Ruído e o contexto ruidoso", são um exemplo disso mesmo.

A tradução literal de *stem* para português é caule ou tronco[38]. Todavia, é mais rigoroso designar este conceito por *radical.* Segundo [CUNHA02], "Ao (...) MORFEMA LEXICAL dá-se tradicionalmente o nome de RADICAL. É o radical que irmana as palavras da mesma família e lhes transmite uma base comum de significação". A transformação por redução ao *radical* das palavras, acção designada por *stemming*, é uma outra técnica destinada a reduzir a dimensão dos ficheiros de indexação e a aumentar o desempenho dos sistemas de IR.

> "A desvantagem da redução ao radical, no momento da indexação, reside na perda de informação relativa aos termos completos; em alternativa será necessária capacidade de armazenamento adicional, para guardar tanto a forma original como o seu radical" [39] [FRAKES92: 131].

Por sua vez um *thesaurus* é um tipo de dicionário que agrupa palavras com significado semelhante[40]. Sendo um vocabulário "comum, preciso e controlado" (Srinivasan in [FRAKES92: 161]) um *thesaurus* pode ser um esquema de classificação ou um catálogo com funcionalidades mais ou menos ampliadas. A sua utilização em sistemas de IR é resumida como se segue.

> "Ao indexar, é derivada uma representação sucinta do documento, enquanto a consulta é baseada no processo de pesquisa através do qual são identificados os itens relevantes." [41] .

3.2. Os erros ortográficos

Os sistemas de *Information Retrieval*, de um modo geral, ignoram o tratamento de erros ortográficos. A presença dos termos *stoplist* e *stem*[42], na Tabela 2, acentuam esta noção com clareza. Em sistemas de *Information Retrieval*, segundo Srinivasan in [FRAKES92: cap. 9], "o *thesaurus* serve para coordenar os processos básicos de indexação e de pesquisa de documentos" [43]. O problema da correcção de erros não é apresentado como tendo importância neste contexto.

Decorre do conceito de stoplist que erros ortográficos coincidentes com palavras pertencentes ao dicionário negativo, são pura e simplesmente ignorados. No limite, implementando a sequência *stoplist->stemming->thesaurus*, o efeito combinado consistiria em procurar substituir termos pelos sinónimos dos seus radicais, tanto para a indexação como para a elaboração de consultas. Neste quadro, os erros ortográficos são tratados como quaisquer outros

[38] Cf. FERREIRA, P. [e] Júlio Albino (1954) – **Dicionário Inglês-Português**, Editorial Domingos Barreira, Porto.
[39] Orign. "The disadvantage of indexing time stemming is that information about the full terms will be lost, or additional storage will be required to store both the stemmed and unstemmed forms. [FRAKES92:131].
[40] Cf. Advanced Learner, **Cambridge Dictionaries Online** - Cambridge University Press [Em linha, consultado em 2004-02-17] Disponível na WWW:<http://dictionary. cambridge. org/>
[41] Orign."In indexing, a succinct representation of the document is derived, while retrieval refers to the search process by which relevant items are identified*. "* [FRAKES92]
[42] *Stem* significa a redução da palavras ao seu radical [FRAKES92: 131-132].
[43] "the thesaurus serves to coordinate the basic processes of indexing and document retrieval" [FRAKES92: 161].

termos. Podem chegar a influir fortemente em esquemas de ponderação estatística apenas baseados na frequência de termos em documentos [SINGHAL96a].

O impacto desta acção não é, todavia, simplesmente negativo. Em primeiro lugar, por definição, o sistema de IR existe para acelerar a pesquisa de informação. Em segundo lugar, e em geral, tanto o processamento da indexação como o da consulta sofrem o mesmo tratamento, i. e, limpos de *noise words* e reduzidos pela acção de *stemming* e eventual acção do *thesaurus*. Em terceiro lugar, cabe ao utilizador a análise crítica do resultado da sua pesquisa.

Considere-se, por exemplo, um universo de associações de radicais de palavras que ocorrem por erro ortográfico, mas que, do ponto de vista semântico, dificilmente teriam lugar. O sistema de IR com essa informação indexada, responde a qualquer consulta formulada, o que inclui as associações referidas. Caberá ao utilizador identificar aqueles documentos que lhe interessa conferir.

3.3. Medidas de qualidade das consultas

Ao ser desencadeada uma pesquisa, pretende-se obter o maior número de respostas exactas e que abarquem a totalidade das respostas possíveis. Este objectivo não é fácil de conseguir ao desencadear consultas sobre ficheiros de texto não estruturado. A eficácia de uma pesquisa depende de dois factores principais [SALTON88: 516]. Em primeiro lugar devem ser obtidos os documentos[44] provavelmente mais relevantes para o utilizador. Em segundo lugar devem ser rejeitados todos aqueles documentos que são estranhos [Ibid.].

É necessário estabelecer alguns compromissos, o que significa encontrar medidas de precisão e de revocação[45] para uma dada consulta. Neste sentido [SALTON88] e [FRAKES92], entre outros, expõem os conceitos “recall” e “precision” da seguinte forma.

Partindo do princípio de que, na colecção, há um conjunto de documentos relevantes para o utilizador, nem sempre é abrangida a totalidade desses documentos em consequência de uma consulta. “Recall” é a proporção de documentos relevantes obtidos, medida pela razão entre o número de documentos relevantes obtidos sobre o número de documentos que se revoca[46] como relevantes na colecção.

$$recall = \frac{\text{número de documentos relevantes obtidos}}{\text{número de documentos relevantes da colecção}}$$

Por outro lado, em resultado de uma consulta é apresentado um conjunto de documentos que satisfazem os critérios de pesquisa. Nesse conjunto há documentos que são relevantes, i. e, interessam ao utilizador, ou não. Precisão é a relação entre o número de documentos relevantes e o número de documentos obtidos.

[44] “item” é a expressão usada em [SALTON88] quando se refere a documentos.
[45] Revocação ou recordação, como tradução de *recall*.

$$precision = \frac{\text{número de documentos relevantes}}{\text{número total de documentos obtidos}}$$

Estes factores *recall* e *precision* tomam valores situados entre 0 e 1. Seria desejável que ambos fossem iguais a 1. Todavia, na prática , quanto maior é a precisão, menor é a abrangência ("recall"). O conhecimento deste compromisso torna-se quase intuitivo e é vulgarizado entre quem habitualmente desencadeia pesquisas na Web.

A relevância dos documentos depende dos critérios do utilizador no acto de consulta. Isto identifica uma carga de subjectividade que não pode ser ignorada. O número de documentos relevantes em regra é desconhecido [FRAKES92: 10], só sendo possível criar estimativas partindo de um ambiente de amostras controladas no qual seja possível determinar, com rigor, quais são os documentos relevantes para uma dada pesquisa.

3.4. Representação de documentos no espaço vectorial

A ideia de que os documentos podem ser representados por vectores de termos significativos é relativamente antiga[47]. Num artigo que "resume 20 anos de experiência em ponderação automática de termos no SMART" [48], Salton e Buckley [SALTON88], expõem a ideia de que os documentos podem ser representados por vectores de termos.

$$D = (t_i, t_j, \ldots, t_p)$$

Em que t_k representa o termo associado e contido no documento D.

Os termos são escolhidos por extracção de determinadas palavras dos textos ou por escolha arbitrada por especialistas de classificação. Paralelamente as consultas ou pesquisas podem ser formuladas a partir de vectores de termos de pesquisa

$$Q = (q_a, q_b, \ldots, q_r)$$

ou por expressões de lógica matemática booleana

$$Q = (q_a \text{ and } q_b) \text{ or } (q_c \text{ and } q_d \text{ and } \ldots) \text{ or } \ldots$$

Onde, uma vez mais, q_k representa o termo associado a consulta Q.

Considerando w_{dk} como a ponderação dos termos dos documentos e w_{qk} das consultas, a representação vectorial do documento e das consultas passará a ser

---

[46] Ou recorda.

[47] Salton e Buckley [SALTON88] situam a formulação exposta em finais dos anos 50 ("late 1950s"), referindo Luhn, H. P. , A statistical approach to the mechanized encoding and searching of literary information, **IBM Journal of Research and Development** 1:4; 309-317; October 1957. Donna Harman (HARMAN, Donna, "Ranking Algorithms", in [FRAKES92: cap 14]) vai mais longe oferecendo-nos uma significativa citação do mesmo artigo: "the more two representations agreed in given elements and their distribution, the higher would be the probability of their representing similar information".

$D = (t_0, w_0; t_1, w_1; ...; t_t, w_t)$

e

$Q = (q_0, w_0; q_1, w_1; ...; q_t, w_t)$

O ajustamento (ou semelhança) entre a formulação de uma consulta e um documento pode ser medido por comparação dos vectores respectivos utilizando a fórmula do produto vectorial

$$\text{semelhança}(Q,D) = \sum_{k=1}^{t} w_{qk}.w_{dk}$$

Quando se limitam as ponderações a 0 e 1 esta expressão devolve o número de termos coincidentes entre a consulta Q e o documento D. Tais ponderações são um caso particular e não proporcionam uma discriminação de conteúdos tão boa como a que será obtida variando os factores de ponderação continuamente entre 0 e 1 [SALTON88].

Por outro lado, dado que os documentos têm diferentes dimensões, torna-se necessário normalizar os vectores de representação de documentos e de consultas. Assim, introduzindo um factor de normalização, a representação dos documentos é $\frac{w_{dk}}{\sqrt{\sum_{vector} (w_{di})^2}}$ e das consultas $\frac{w_{qk}}{\sqrt{\sum_{vector} (w_{qi})^2}}$.

Considerando as representações anteriores, a semelhança entre as consultas e os documentos resulta na fórmula do co-seno

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

Esta forma de representação de documentos no espaço vectorial, exposta por Salton e Buckley [SALTON88] e aqui resumidamente reproduzida, é frequentemente considerada na literatura. De facto em [DHILLON00: 2] e [DHILLON99: 1] pode ler-se que o espaço vectorial

[48] (HARMAN, Donna, "Ranking Algorithms", in [FRAKES92: cap 14]): "summarizes 20 years of SMART experiments in automatic term-weighting"

constitui “um ponto de partida para a aplicação de algoritmos de agregação sobre dados de texto não estruturado” [49] . Os mesmos autores e W. Scott Spangler em [DHILLON98], sugerem que “os documentos de texto (...) podem ser tratados como vectores num espaço de características multidimensional“. Em [WANG00: 3-4] e [MENG02: 6], “Cada documento pode ser representado como um vector de termos com ponderações” [50]; mais adiante esta frase é repetida com a palavra “documento” substituída por “descrição”, com um significado muito idêntico ao de “consulta” ou “query”.

3.4.1. Esquemas de ponderação

Acompanhando a exposição sobre a representação de documentos no espaço vectorial, é exposto, em [SALTON88], um esquema que utiliza três componentes para o cálculo da ponderação, *tf*, *idf* e *factor de normalização*. Para o estabelecimento deste esquema foram tidas em consideração as medidas de qualidade *recall* e *precision*. As considerações que fundamentam o estabelecimento de cada um dos factores desse esquema aparecem também, talvez mais clara e resumidamente, em [SINGHAL96a: 4] da seguinte forma.

- *Term frequency* ou *tf*

> “Tipicamente, um termo que ocorre frequentemente num texto é mais importante nesse texto do que um termo pouco frequente. Portanto, o número de ocorrências de um termo, vulgarmente chamado *frequência de termo* ou *tf*, é utilizado no texto como factor de ponderação do termo.”

- *Inverse document frequency* ou *idf*

> “Palavras comuns tendem a ocorrer em numerosos documentos de uma colecção, e são fracos indicadores do conteúdo de um documento. Quanto maior for o número de documentos em que um termo ocorre, menos importante poderá ser. Portanto, a ponderação de um termo deverá ser inversamente proporcional ao número de documentos em que o termo ocorre, em toda a colecção, chamada frequência do termo na colecção (ou frequência de documentos).”

- Factor de normalização

> “Os documentos longos e palavrosos geralmente usam repetidamente os mesmos termos. Em consequência, os factores de frequência de termos podem ser grandes para documentos longos. Os documentos longos também têm numerosos termos diferentes. Isto aumenta o número de coincidências de palavras entre uma consulta e um documento longo, aumentando as possibilidades da sua localização face a documentos mais reduzidos. Para compensar este efeito é utilizada frequentemente a

[49] Orign. “A starting point for applying clustering algorithms to unstructured text data (…)”
[50] Each document d in D can be represented as a vector of terms with weights

normalização da ponderação de termos. A normalização é uma forma de impor alguma penalização aos pesos dos termos de documentos mais longos. Uma técnica de normalização eficaz é a *normalização pelo co-seno*. Cada peso de termo num documento é dividido pela distância Euclidiana do vector ponderado do documento $tf \times idf$, $\sqrt{w_1^{\,2} + w_2^{\,2} + ... + w_n^{\,2}}$, onde w_i é a ponderação $tf \times idf$ do termo de ordem *i* no documento. Isto dá-nos a ponderação final de um termo como:

$$\frac{\textit{ponderação } tf \times idf}{\textit{Comprimento Euclideano do vector do documento}}$$

Os termos ausentes de um documento são considerados como tendo ponderação zero."

É de notar que as considerações, relativas ao estabelecimento do factor *idf,* sugerem uma resposta às observações tecidas no capítulo 1 em torno do que aí se considerou ser o ruído do contexto.

Term Frequency Component	
b	1,0
t	tf
n	$0{,}5 + 0{,}5 \times \frac{\text{tf}}{\max \text{tf}}$
Collection Frequency Component	
x	1,0
f	$\ln \frac{N}{n}$
p	$\ln \frac{N-n}{n}$
Normalization Component	
x	1,0
c	$1 \Big/ \sqrt{\sum_{vector} w_i^{\,2}}$

Tabela 3 - Componentes de ponderação de termos (in [SALTON88])[51]

Descrevendo a ponderação de termos como uma tríade completa-se o esquema [SALTON88]. A tríade é um produto de três factores, designados em [SALTON88] respectivamente por "Term Frequency Component", "Collection Frequency Component" e "Normalization Component". A Tabela 3, explicita a composição de cada um dos factores que podem formar uma tríade. O aspecto que toma a cada uma destas tríades pode ser, por exemplo, *tfc* representando o produto $tf \times \ln \frac{N}{n} \times 1 \Big/ \sqrt{\sum_{vector} \sqrt{w_i^{\,2}}}$ ou, mais compactamente,

[51] Nesta tabela usa-se a notação à portuguesa, e substitui-se log por ln, designado logaritmo neperiano.

$\frac{tf \cdot \ln \frac{N}{n}}{\sqrt{\sum_{vector} \left(tf_i \cdot \ln \frac{N}{n_i} \right)^2}}$. Isto é, substituiu-se cada uma das letras, pela expressão correspondente da Tabela 3 [SALTON88].

A mesma noção é aplicável à ponderação de consultas. Desta feita, associando uma tríade *ddd* à ponderação de termos em documentos e outra *qqq* às ponderações de termos estabelecidos nas consultas, o produto *ddd. qqq* pode caracterizar um ensaio de recuperação de informação [SINGHAL96a: 5].

A composição da Tabela 3 não deve ser tomada como sendo uma referência final. Em [SINGHAL96a][52], ver Figura 15, pode observar-se uma tabela ligeiramente diferente da Tabela 3.

Term Frequency		Inverse Document Frequency		Normalization	
First Letter	$f(tf)$	Second Letter	$f(\frac{1}{df})$	Third Letter	$f(length)$
n (natural)	tf	n (no)	1	n (no)	1
l (logarithmic)	$1+\log(\text{tf})$	t (full)	$log(\frac{N}{df})$	c (cosine)	$\sqrt{{w_1}^2 + {w_2}^2 + \ldots + {w_n}^2}$
a (augmented)	$0.5 + 0.5 \times \frac{tf}{max\ tf}$				

Table 1: Term Weights in the Smart System

Figura 15 - Componentes de ponderação de termos (in [SINGHAL96a])

Também [DHILLON00; DHILLON99] se refere a um conjunto apresentado por Kolda[53], com "5, 5 e 2 esquemas, respectivamente para os componentes termo, global e normalização – um total de $5 \times 5 \times 2 = 50$ escolhas"[54]. Desse conjunto designa como "populares" os esquemas **txn** e **tfn**.

> "Ambos os esquemas enfatizam as palavras com frequências mais elevadas, e utilizam $t_{ji} = f_{ji}$. O esquema **txn** utiliza $g_j = 1$, enquanto que o esquema **tfn** enfatiza as palavras com baixa frequência no conjunto da colecção e utiliza $g_j = \log(n/d_j)$. O vector[55] de cada documento está normalizado, em ambos os esquemas, com norma unitária L^2, isto é, $s_i = \left(\sum_{j=1}^{d} (t_{ji} g_j)^2 \right)^{-1/2}$ " [DHILLON00: 5].

Comparando com a Tabela 3, pode dizer-se que os esquemas atrás expostos são idênticos aos esquemas **txc** e ao **tfc** que resultariam dessa tabela.

[52] De notar que os autores deste paper são Amit Singhal e também Gerard Salton e Cris Buckley.
[53] Kolda, T. G.: 1997, 'Limited-Memory Matrix Methods with Applications'. **Ph.D. thesis**, The Applied Mathematics Program, University of Maryland, College Park, Mayland.
[54] Na Tabela 3 o número de possibilidades é $3 \times 3 \times 2 = 18$ e $3 \times 2 \times 2 = 12$ na Figura 15.
[55] Os autores [DHILLON00] consideram que os documentos podem ser representados por vectores resultantes de vectores de termos que serão, *grosso modo*, as tríades.

3.4.2. Medidas de semelhança

Nas descrições das medidas de ponderação as comparações têm sido efectuadas entre a formulação de consultas e os documentos. Todavia podem ser consideradas para medir a proximidade entre documentos. As medidas de semelhança serão necessárias para proceder à análise de clusters.

Tomando dois vectores num espaço de N dimensões, uma forma de os comparar consiste em medir a sua proximidade angular. Esta formulação tem sido muito utilizada porque é simples a sua interpretação geométrica.

O coeficiente de Jaccard mede o grau de coincidência entre dois conjuntos que se interceptam [GUHA00: 350; WEIDE01: 49]. Considerando os conjuntos A e B a semelhança entre eles é medida dividindo o número dos elementos coincidentes pelo número total de elementos considerando a reunião dos dois conjuntos: $\text{semelhança}(A,B) = \frac{|A \cap B|}{|A \cup B|}$.

O coeficiente Dice "é a relação entre a sobreposição dos conjuntos A e B com uma estimativa da sua dimensão média" [WEIDE01: 50]: $\text{semelhança}(A,B) = \frac{|A \cap B|}{\frac{1}{2}(|A| + |B|)}$

Tanto as consultas como a representação dos documentos têm uma natureza objectiva. Portanto, estas medidas de semelhança também são objectivas. Não dependem dos critérios subjectivos do utilizador, conforme a definição das medidas de qualidade.

3.5. Duas atitudes de classificação

Há duas atitudes possíveis para proceder ao início de uma classificação de documentos. Ou se parte de uma classificação existente e são identificados os documentos que se conformam com essa classificação, ou são analisados os documentos e é construída uma classificação a partir de características comuns entre eles. Estas duas visões serão casos extremos, mas têm a sua correspondência em termos de implementação técnica.

No primeiro caso pode ser construído um catálogo suportado num esquema de classificação cujas classes são caracterizadas por palavras-chave ou *keywords*, i.e., um *thesaurus*. No segundo caso pode ser construída uma classificação recorrendo à realidade do corpus.

Nenhuma destas atitudes está isenta de dificuldades no que se refere à dimensão do que está envolvido. A primeira, porque exige a identificação clara das palavras-chave a considerar. A segunda, porque necessita de um grande esforço na determinação e remoção de características (palavras) desnecessárias.

A análise de clusters está largamente documentada. [WILLET88; FRAKES92; DHILLON99; DHILLON00; WISE00], expõem modos de utilização destas metodologias no contexto de processamento de linguagem natural.

3.6. Agregação ou clustering

No capítulo 1, e citando [WILLETT88], referiu-se que a análise de clusters[56] permite a identificação de classes de forma algo organizada, contribuindo para a formação do esquema de classificação. Assim, essas metodologias reúnem dois aspectos decisivos. Por um lado identificam qualidades agregadas com um significado mais definido, e por outro dispõem tais agregados segundo diferentes níveis ou posições.

Uma das formas mais simples de representação do conceito de agregação consiste na sua forma de representação gráfica, mais conhecida por dendrograma [REIS01: 314 sgg.], ou diagrama hierárquico de cluster [ELMAN90: 20]. Neste diagrama, Figura 16, as palavras foram agregadas de acordo com o seu papel nas frases (v. cap. 2). As diferentes distâncias entre elas evidenciam a maior ou menor semelhança desse papel. Esta figura ilustra também a forma de chegar a uma classificação, partindo da análise de clusters.

[WILLETT88: 579-582] e [REIS01: cap. 12] descrevem diferentes métodos de agregação. Identificam dois importantes agrupamentos de técnicas: métodos não hierárquicos e hierárquicos [WILLETT: 579]. Ambos sublinham que os métodos não hierárquicos são pesados em termos computacionais [ibid.; REIS01: 298].

"Os métodos de análise de clusters mais divulgados e mais utilizados são os hierárquicos aglomerativos" [REIS01: 298][57]. A utilização destes métodos pressupõe a escolha das medidas de semelhança a utilizar. Elisabeth Reis identifica os "coeficientes de correlação" e as "medidas de distância" como as mais utilizadas [op. cit. p. 299].

No caso do processamento de documentos, a opção adoptada em regra é a medida da distância entre documentos, conforme as definidas em 3.4.2 ou matrizes de semelhança (Rasmussen in [FRAKES92: 423] e [WILLET88: 581].

[56] Em inglês "cluster analysis". Entre nós, portugueses, estas metodologias também são conhecidas por Análise de Clusters. [REIS01] intitula dessa forma um capítulo inteiramente dedicado a este assunto.
[57] [WILLETT: 580] expõe algumas considerações colocando as técnicas divisivas em alternativa às aglomerativas. Também conclui que os métodos aglomerativos são os mais populares.

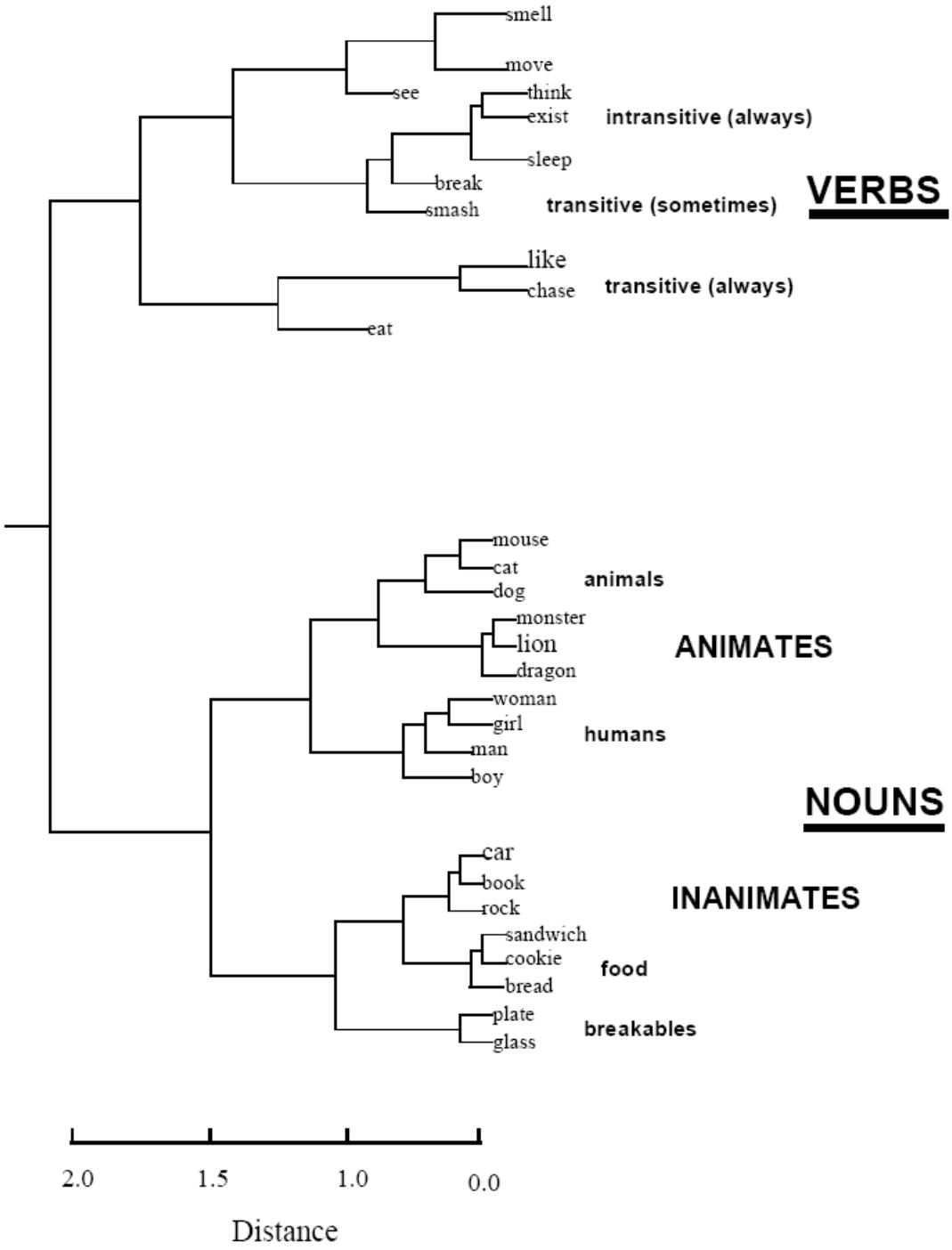

Figura 16 - Exemplo de dendrograma [ELMAN90: fig. 8]

Após esta definição, aplicam-se os critérios de agregação, que "diferem no modo como estimam distâncias entre grupos já formados e outros grupos ou indivíduos por agrupar" [REIS01: 310]. Os mais conhecidos [ibid.]

- *Single linkage* ou critério do vizinho mais próximo;

- *Complete linkage* ou critério do vizinho mais afastado;
- Critério da média dos grupos;
- Critério do centróide;
- Critério de Ward.

Qualquer que seja a metodologia escolhida será sempre necessário construir vectores de frequência de palavras por documento e dimensão dos documentos, i.e., número total de palavras no documento ou a respectiva dimensão em *bytes* [SINGHAL96a].

3.7. Conclusões do capítulo 3

A preocupação dos sistemas de Information Retrieval é facilitar a localização de documentos. Criam-se, com esse objectivo, sistemas de índices em que o desempenho na resposta às consultas é a principal prioridade. O sistema de índices criado pelo sistema de *information retrieval* é uma base de dados. Os documentos e as consultas podem ser representados por vectores de termos com variados esquemas de ponderação.

De um modo geral, documentos com erros ortográficos são tratados, pelo sistema de *information retrieval*, como quaisquer outros, não havendo distinção entre palavras bem ou mal escritas.

A classificação de documentos pode ser efectuada metodicamente recorrendo à análise de clusters.

Capítulo 4. O sistema de serviços de indexação da Microsoft

4.1. Génese e evolução do Index Server

Krishna Nareddy, do Windows NT Query Team da Microsoft Corporation no seu artigo "Introduction to Microsoft Index Server" de 15 de Outubro de 1997, descreve esta evolução. O Index Server começou por ser um Content Indexer (CI) que fazia parte do Object File System (OFS), desenvolvido como parte do sistema operativo Windows NT [NAREDDY97b].

A necessidade de encontrar soluções de pesquisa para a Web que reunissem elevados níveis de fiabilidade, eficiência e robustez, conduziu à construção do Index Server em torno do CI. De Agosto de 1996 até finais 1997 a Microsoft já tinha disponibilizado as versões 1.0 e 1.1 do Index Server destinado a ser utilizado com o Windows NT 4.0 Server e com o Internet Information Server (IIS) 3.0 [ibid.].

Seguiu-se-lhes, em finais de 1997, o Index Server 2.0 distribuído com o Windows NT Option Pack 4.0. Por esta altura, do ponto de vista conceptual, terá atingido alguma estabilidade[58] [NAREDDY98]. Renomeado como Indexing Service 3.0 ou simplesmente Indexing Service veio a ser incorporado no Windows 2000 ([ibid.], [SPENCER00]) e, mais tarde, no Windows XP (MSDN, 2003a). Em ambos os casos, é uma opção na instalação normal do sistema operativo, uma vez que se trata de um componente que necessita de utilização intensa de acesso a disco [TECHNET01].

Está incluído no conjunto de produtos que incorporam conceitos de pesquisa de texto da Microsoft [op. cit.].

4.2. Alguns conceitos e características do Index Server (IS)

4.2.1. Corpus

O "corpus" significa "uma molhada de documentos em diversas formas de ficheiro, estruturados de diversas maneiras e, possivelmente, escritos em diversas linguagens" [59]. Na caracterização de um corpus foram considerados os seguintes aspectos: segurança, formato de ficheiros, linguagens, dimensão, fluxo e localização [NAREDDY97a].

[58] Esta ideia está implícita no terceiro parágrafo do artigo referido. Aí também se diz que haverá diferenças de comportamento da versão 2.0 para a 3.0. Como no mesmo parágrafo se anuncia a actualização do artigo e tal não se encontrou entretanto, presume-se que tais diferenças são as enumeradas em [MSDN03a], que se doravante se distinguirão por referência explícita ao IS 3.0.

[59] Orign. "a bunch of documents in several file formats, structured in several ways, and possibly written in several languages" [NAREDDY97A]. A tradução literal de "bunch" é molho ou feixe [LANGENSCHEIDT60], significando conjunto. O texto citado está escrito num estilo informal, conferindo-lhe alguma proximidade com o leitor.

Por segurança entende-se que, se um documento não puder ser acedido através dos meios convencionais, também o não poderá ser recorrendo ao motor de pesquisa[60] [op. cit. p. 2]. Embora ao processo de indexação do IS seja potencialmente permitido[61] o acesso ao conjunto dos ficheiros residentes no filesystem NT, na execução de consultas, implementa a segurança de acordo com as permissões de acesso do utilizador. Isto significa que os documentos, fora do âmbito autorizado, nem sequer são referidos no resultado de uma consulta efectuada nessas condições [op. cit. p. 5].

Permite a indexação de documentos criados em diversos formatos. Esta característica é possível porque o IS utiliza um interface, designado por IFilter, para extrair as características dos documentos, mediante filtros adequados [NAREDDY97b: 2].

O IS diferencia as linguagens em que cada documento está escrito, incluindo variações que pode conter [ibid.]. Dispõe de módulos que "reconhecem os conceitos no texto" [62]. Talvez infelizmente, no conjunto de línguas consideradas nos módulos disponíveis, a língua portuguesa está omissa. As línguas consideradas à partida são Inglês, Chinês, Francês, Alemão, Coreano, Castelhano (Espanhol), Italiano, Holandês, Sueco e Japonês[63] [NAREDDY98: 8]. Mais recentemente é adicionado o Tailandês [MSDN03c]. É utilizada a informação "locale" para identificar a língua. Através desta identificação são escolhidos os instrumentos léxicos adequados [NAREDDY98: 8]. Por omissão assume-se a língua estabelecida pelo "locale" do servidor. Contudo os documentos podem ultrapassar essa definição se contiverem o metadado "MS.Locale" estabelecido para uma língua diferente [ibid.].

O IS não apresenta grande sensibilidade às variações da dimensão do *corpus*, no que respeita a tempos de indexação e resposta a consultas. Esta característica é importante num contexto de desempenho de algoritmos que exijam pesquisa. Contudo, no dimensionamento, deve considerar-se que o IS necessita de um espaço em disco que ronda os 40% do espaço ocupado pelo corpus [NAREDDY98: 16; FEDOROV98: 5].

O fluxo do *corpus*, isto é, alterações ao seu conteúdo por adição, remoção ou alteração de ficheiros não exigem que o IS pare o seu serviço de indexação [NAREDDY97a: 3]. O IS, por mecanismos que a seguir se descrevem, detecta alterações ao *corpus* e providencia a indexação de documentos alterados. Pode aceitar notificações das alterações produzidas pelo *filesystem* [NAREDDY97b: 3]. Como se trata de um sistema que utiliza intensivamente os recursos de acesso ao disco, naturalmente não disponibiliza imediatamente a informação relativa a novos documentos [NAREDDY97a: 3].

Há dois mecanismos de obtenção de documentos para indexação: o varrimento e as notificações [NAREDDY98: 6]. O mecanismo das notificações do Windows NT, por ser o mais

[60] O aspecto da segurança tem tido falhas, conforme se ilustra no Anexo II.
[61] Depende da configuração de "scopes" para indexação, infra.
[62] Orign. "modules (...) that recognize concepts in text", op. cit., p. 4. Esta asserção deve ser tomada com alguma cautela. Como se verá adiante, neste capítulo, será válida apenas na execução de consultas.

eficiente, é o mais utilizado quando está disponível [ibid.]. Na versão IS 3.0 e em volumes formatados com o *filesystem* NTFS 5.0[64] [MSDN03a] é introduzida a utilização do *Update Sequence Number* (USN) do registo de alterações, designado por *change journal*. Desta forma, a utilização intensiva de disco no varrimento é dispensada.

Por sua vez o varrimento, executado pelo Index Server, tem dois tipos de execução: completo e incremental [ibid.]. A execução do varrimento completo dá-se quando o IS procede à construção do inventário completo de documentos, por inclusão de novos directórios ou por falha grave [ibid.]. O varrimento incremental consiste numa actualização dos índices sempre que o IS reinicia a sua actividade após uma paragem [ibid.]. O IS reconhece a presença de ficheiros, entretanto alterados, e procede à sua reindexação. Também pode ser desencadeado se a frequência das alterações, no *filesystem*, for muito elevada, conduzindo a perdas, por sobrecarga de notificações, do sistema operativo [ibid.].

O IS indexa documentos que residem principalmente no *filesystem*[65] [NAREDDY97b: 3]. Não se consideram, no âmbito do IS, outras possibilidades de obtenção de documentos para indexação localizados fora do controlo do sistema operativo em que o IS está a correr[66]. O corpus está "organizado como um conjunto de escopos" [67] que coincidem com os directórios locais ou remotos.

4.2.2. Documentos

Identificam-se dois pontos de vista para um documento, o do IS e o do autor. O IS considera cada ficheiro como um documento [NAREDDY97b: 3]. É frequente a organização de trabalhos, com preparação de documentos, conduzir o seu autor (ou autores) à criação de vários ficheiros com textos, gráficos, mapas, a título de exemplo. Nestas condições – do ponto de vista do(s) autor (es) - o resultado final é um documento distribuído por vários ficheiros. Se o autor pretender que o IS identifique esse conjunto de ficheiros como um documento, terá que os agregar e organizar apenas num ficheiro [ibid.]. A mesma lógica aplica-se no sentido inverso. Se o conteúdo de um ficheiro significar documentos diversos, para que o IS os diferencie, terá que ser subdividido de acordo com os documentos que contém [ibid.].

Sendo um produto da Microsoft seria de esperar que o IS considerasse, na sua configuração original, um tratamento mais detalhado e orientado para os formatos de ficheiro que o fabricante introduziu. Os ficheiros com extensão ".doc" e ".xls", por exemplo, são considerados

---

[63] Em [NAREDDY97b] estão omissos o Chinês e o Coreano.
[64] Segundo [RUSSINOVICH02], o NTFS 5.0 é a versão NTFS incluída com o Win2K. Em [MICROSOFT00] pode ler-se que o NTFS 5.0 é compatível também com o Windows NT 4.0 a partir do Service Pack 4.
[65] Este aspecto não é muito claro, havendo referências a indexação de múltiplos servidores Web (Cf. ex. [NAREDDY97b]). Refere-se como preferível, nesse caso, a utilização do Microsoft Site Server Search [ibid.]. No contexto do presente trabalho não se consideram como relevantes essas referências.
[66] Cf. Sec. "Gathering Documents", ibid.
[67] orign. "is organized as a set of scopes", ibid.

como apresentando um formato Microsoft Office sendo as suas propriedades consideradas à partida pelo IS [op. cit. p. 2].

Como se referiu na secção anterior, o que regula a colheita de propriedades dos ficheiros, são filtros que implementam o interface IFilter, destinado a permitir a indexação de diferentes formatos de documentos [NAREDDY97b: 3]. A configuração inicial do IS vem acompanhada de filtros para HTML, simples ficheiros de texto e formatos Microsoft Office [ibid.].

4.2.3. Catálogo do Index Server

Toda a informação necessária ao funcionamento do IS relativamente a um corpus é designada como "catalog". Tal informação está reunida no *registry* e por armazenamento em disco [NAREDDY98].

No acto de instalação, o IS cria um ou mais "catalogs", o que depende da versão e da configuração do MS Windows. O IS 2.0 distribuído com o Windows NT Option Pack 4.0, cria à partida um "catalog" intitulado Web. O IS 3.0, distribuído com o W2K e na condição de o IIS estar a executar, cria dois "catalogs", *System* e *Web* [SPENCER00; MSDN03a].

O *catalog* é constituído por quatro elementos fundamentais, "source directories", "property cache", "content index" e "control attributes" [NAREDDY98]. A informação relativa a "source directories" e "control attributes" é guardada no *registry*. Essencialmente consiste em parametrizações, tanto dos "scopes" ou áreas a abranger na indexação, como em parametrizações comuns dos *catalogs* criados [ibid.].

Um conjunto de ficheiros designado por "property cache", guardados em disco destina-se a optimizar a pesquisa de informação relativa a propriedades mais utilizadas como, por exemplo, o "Path" e "Filename". Pela sua dimensão, a "property cache" está dividida em partes paginadas a 64Kb. Os mecanismos de fluxo atrás descritos são governados por este elemento [op. cit. p. 2-3].

Finalmente o "Content Index" contém a informação textual extraída dos documentos. Esta informação está armazenada sob uma forma compilada e distribuída por vários ficheiros, tendo em vista tornar mais eficaz a resposta a consultas [op. cit. p. 3-4]. Trata-se de uma área que deve merecer alguma protecção de acesso visto que contém um resumo completo do *corpus*. A reconstituição da informação seria difícil, trabalhosa, mas possível para quem se dê a esse trabalho [ibid.].

4.2.4. O Processamento da Indexação e das Consultas

De um modo geral, importantes aspectos do processamento da indexação e das consultas já foram descritos ao longo desta secção. A Microsoft, em [MSDN03b], também descreve estes processamentos como integrados na utilização dos recursos de linguagem. [MSDN03b] pode ser

considerada como uma síntese esclarecedora sobre o que acontece com as palavras nas duas circunstâncias.

Na Figura 17 ilustra-se o fluxo de informação no acto de indexação. De facto, é "utilizado um componente IFilter para aceder a um documento no seu formato nativo"[68] [ibid.]. Conforme já referido, este componente é responsável por identificar o "locale" do documento que está a filtrar, e por extrair o texto, propriedades e formatação [ibid.].

O *word breaker*, providencia a individualização das palavras e frases, normalizando também os formatos de datas e horas [ibid.]. As palavras individualizadas atravessam um processo de normalização que consiste na respectiva conversão em maiúsculas [ibid.]. Finalmente são guardadas no *Full Text Index* exceptuando as palavras ruidosas específicas da língua, identificadas com o *locale*.

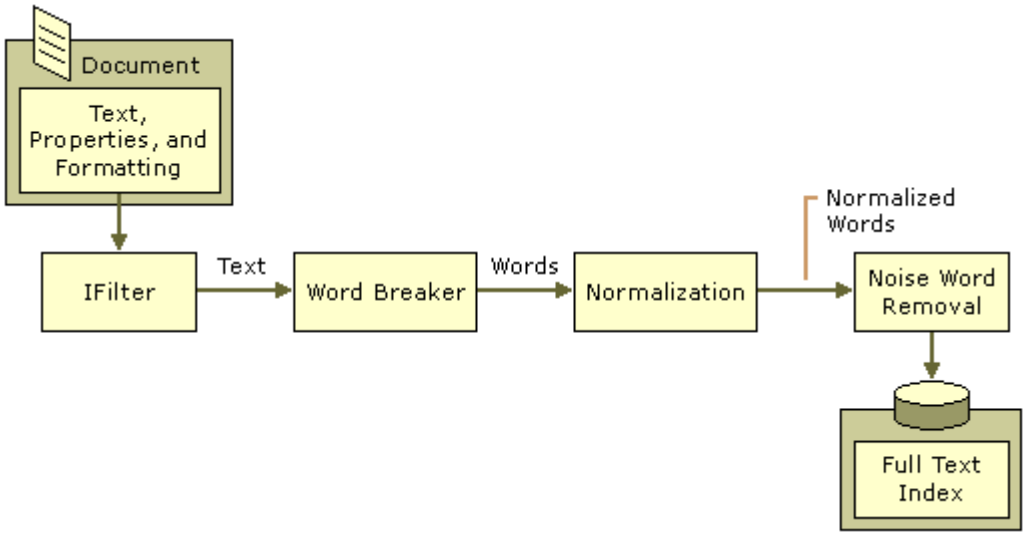

Figura 17 - Recursos da linguagem durante a criação do Índice (in [MSDN03b])

Em [MSDN03b] é também exemplificada toda a sequência que uma frase atravessa, desde o documento até ser guardada pelo IS. Na Tabela 4 repete-se esse exercício com a frase "Figura 1 - Recursos da linguagem durante a criação do **Índice**".

A Figura 18 ilustra o fluxo de informação no lançamento e execução das consultas. Na apresentação de uma consulta ao IS, há lugar à passagem do identificador de *locale*, LCID. Na omissão desse identificador, o IS assume o *locale* estabelecido para o servidor. Do *locale* estabelecido desta forma, depende a escolha dos módulos que se prendem com o tratamento da linguagem a utilizar na consulta, ou mais precisamente, o *word breaker*, *stemmer* e *noise list*. Na

[68] orign. "(…) uses an IFilter component to access a document in its native format"

realidade, conforme se viu, não está considerada a língua portuguesa no conjunto das linguagens para as quais existam módulos preestabelecidos para o IS[69] [MSDN03c].

Processamento	Texto resultante
Texto original	Figura 1 - Recursos da linguagem durante a criação do **Índice**
Filtragem	Figura 1 - Recursos da linguagem durante a criação do Índice
Word breaking	Figura, 1, Recursos, da, linguagem, durante, a, criação, do, Índice
Normalização	FIGURA, 1, RECURSOS, DA, LINGUAGEM, DURANTE, A, CRIAÇÃO, DO, ÍNDICE
Remoção de noise words	FIGURA, RECURSOS, LINGUAGEM, DURANTE, CRIAÇÃO, ÍNDICE
Salvaguarda no IS	FIGURA, RECURSOS, LINGUAGEM, DURANTE, CRIAÇÃO, ÍNDICE

Tabela 4 - Sequência de tratamento de palavras na indexação

Uma consulta, como se verá adiante, pode corresponder a uma expressão SQL como por exemplo,

SELECT filename, path FROM System..SCOPE('DEEP TRAVERSAL OF "C:\My Documents\Estudo\Textos\C1821\"') WHERE CONTAINS(Contents, 'barra').

Neste caso a clausula WHERE tem o predicado CONTAINS. Como a Figura 18 sugere, isto significa que as palavras são entregues directamente para normalização. No exemplo dado seria pesquisada a palavra BARRA, sem qualquer alteração, dado que, em princípio, não pertencerá à *noise list*.

Se o predicado CONTAINS não estivesse presente, a palavra passaria pelo *stemming* antes de ser entregue à normalização. Neste caso a consulta teria em conta todas as palavras cujo radical coincidisse com BARRA.

O conceito de *stemming*, neste contexto, é diferente da simples redução ao radical, conforme definição da secção 3.1.2 acima. Dir-se-á que faz exactamente o contrário da redução. Em [MSDN, 2003b] diz-se "O *stemmer*, para aquela palavra, cria uma lista de formas alternativas ou inflectidas". Tomando como exemplo a palavra BARRA, obter-se-ia BARRAS, BARRINHA, BARRINHAS, e assim por diante desde que o radical fosse o mesmo. Em [MSDN03c] esta acção é exemplificada, talvez ilustrando melhor a ideia, com a palavra *swim*, que se "expande para incluir os termos *swimming*, *swam*, *swum*".

[69] Isto não significa impossibilidade de integração de recursos de linguagem específicos para Português ou outras línguas. Essa possibilidade está descrita em MSDN [Em Linha], Platform SDK, February 2003, passim.

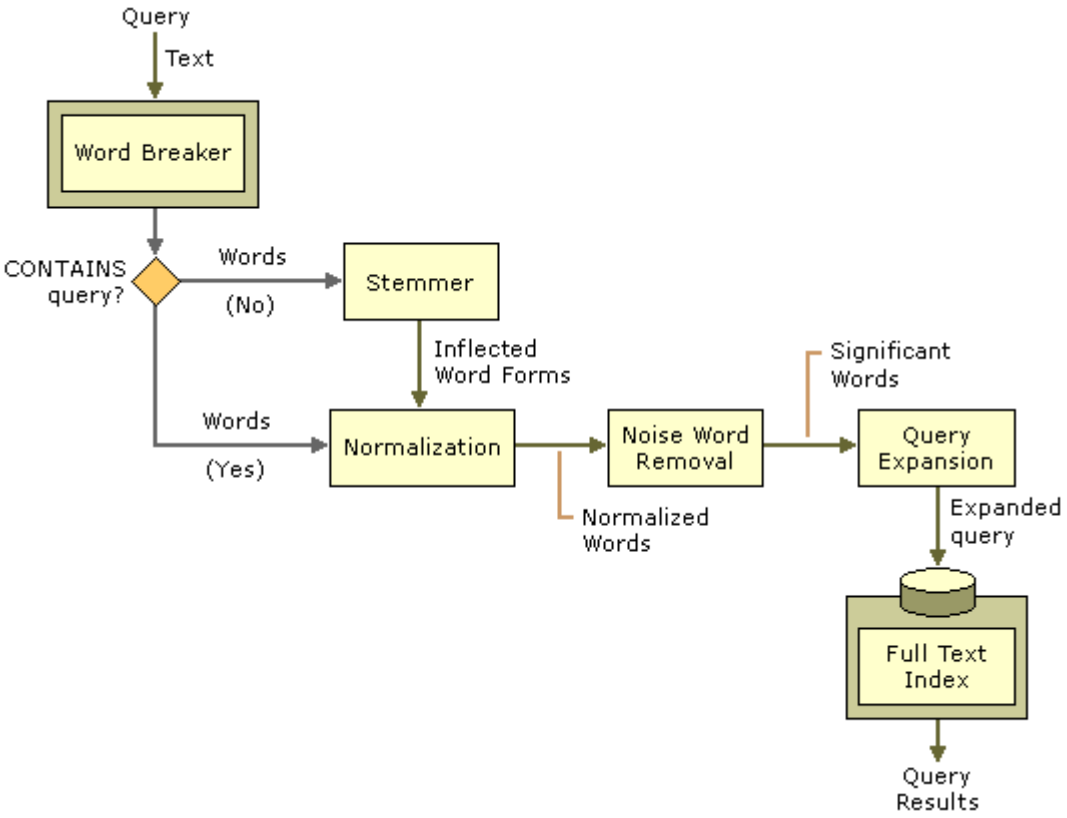

Figura 18 - Recursos de linguagem durante a execução de Consultas (in [MSDN03b])

4.3. Ênfase nas Consultas

O excerto seguinte, extraído do Help na página "Index Service Overview", resume a despreocupação com que o utilizador pode encarar toda a actividade de indexação do IS.

> "O Serviço de Indexação está desenhado para correr continuamente e exige pouca manutenção. Após a sua configuração, todas as operações são automáticas, incluindo a criação e actualização do índice, bem como a recuperação de *crash* na sequência de uma falha de energia."[70]

Considera-se o IS como uma base de dados a partir da qual apenas será possível a leitura, ou seja, lançar consultas. Esta ideia aparece com frequência na documentação que se refere à utilização do IS. Nem sempre, infelizmente, da forma mais rigorosa.

4.3.1. O Indexing Service não é um SGBD

Tomar o IS como uma base de dados a partir da qual é possível fazer consultas, sem esclarecer inequivocamente que alterações directas estão interditas, pode conduzir a

interpretações erradas. Por exemplo, [FEDOROV98: 3], dirigindo-se ao seu leitor, sugere: "Pense no Index Server como uma base de dados, semelhante ao SQL Server ou DBASE, um máquina potente capaz de processar consultas complexas e devolver um conjunto de resultados."[71] A urgência de uma leitura deste texto, e pouco atenta, sugere a imediata e errada interpretação de que o IS é um SGBD. Naturalmente não é possível ignorar o contexto em que os autores escrevem. Em [FEDOROV98] o que se pretende transmitir circunscreve-se à recuperação de dados a partir do IS.

O facto de [FEDOROV98] estar acessível a partir do *site* MSDN, também não é motivo para ajuizar apressadamente sobre a qualidade da informação aí disponibilizada. A Microsoft é uma empresa cuja enorme dimensão, nos tempos que correm, é inegável. Disponibiliza gigantescos volumes de informação cujo tratamento, em termos de precisão de linguagem, será inevitavelmente prejudicado. Tendo isto em consideração, há que conjugar toda a informação, com especial atenção para a que vem assinada pelo fabricante, com o detalhe das características dos respectivos produtos.

Lê-se, em [MSDN03a], "está disponível acesso programático e exclusivamente de leitura aos índices e propriedades do Indexing Service"[72]. Interpretando a conjugação desta frase com tudo quanto se disse sobre o IS, fica claro que não é, rigorosamente, um SGBD semelhante ao DBASE ou ao SQL Server.

4.3.2. Consultas com o formulário do IS

No Anexo IV apresenta-se a forma de criar e aceder a janelas de gestão do *Indexing Service* no Windows 2000. Também se ilustra o aspecto inicial do formulário de consultas. Examinando com detalhe a documentação em linha, ou Help, do IS verifica-se que a sintaxe de consulta está documentada. Mas é importante notar que "Indexing Service Query Language" é um sub-capítulo de "Advanced Topics". Talvez signifique que este assunto se destina a pessoas com competências mais avançadas no domínio da pesquisa de documentos.

O facto de existir um formulário de consultas imediatamente disponível, é um aspecto muito positivo. O formulário permite escolher entre consultas simples, estabelecidas em texto livre, e mais avançadas onde a complexidade dos critérios pode aumentar. Relativamente à lista resultante da consulta, permite ordená-la ascendente ou descendentemente segundo os resultados de *rank*, título, *path*, dimensão e data de modificação. Possui também um *link*, designado por *Unfiltered Do*cs, a partir do qual são apresentados os documentos por indexar.

[70] Orign. "Indexing Service is designed to run continuously and requires little maintenance. After it is set up, all operations are automatic, including index creation, index updating, and crash recovery if there is a power failure."
[71] Orign. "Think of the Index Server as a database, such as SQL Server or DBASE, a powerful engine that can process complex queries and return what amounts to a result set." Parece haver uma expressão idiomática no final, "what amounts to a", cuja tradução por "um conjunto de resultados" poderá não ser a mais precisa. Admite-se que o sentido geral será esse.
[72] Orign. "Programmatic read-only access is available to Indexing Service contente and property indexes(…)"

Tratando-se de um formulário genérico, o resultado obtido está muito orientado para quem apenas pretende efectuar localização de documentos que obedecem a determinados critérios[73].

A rigidez na apresentação dos resultados será um aspecto menos positivo. Poderia ser simpático, para quem estivesse interessado em conhecer imediatamente o valor do *ranking* calculado, poder escolher essa variável em substituição, por exemplo, da data de modificação.

Para aceder a este formulário é necessário escolher o *catalog* sobre o qual irá ser desencadeada a consulta. O *catalog* escolhido incluirá os *scopes*, em regra volumes e directórios, sobre os quais se processa a indexação. Como não é conveniente, por questões de desempenho, fazer proliferar o número de *catalogs* por forma a abrangerem apenas determinados *scopes*, há que elaborar critérios de pesquisa que considerem esse aspecto [NAREDDY98][74]. O ideal seria estabelecer previamente o *scope* sobre o qual deverá incidir a pesquisa. O formulário não facilita essa possibilidade.

4.3.3. Modalidades de programação de consultas

Em alternativa, recorrendo a programação, é disponibilizado um conjunto de mecanismos de consulta que agilizam as possibilidades de utilização do IS. Em [FEDOROV98] estão detalhadamente explicadas algumas implementações, utilizando três técnicas diferentes.

A primeira, designada por "Static Search", utiliza "uma combinação de ficheiros HTM, IDQ e HTX". Esta técnica em [FEDOROV98] é considerada limitada, porque obriga sempre à transferência da totalidade dos dados resultantes da *query*. Mas por outro lado é considerada "eficiente e muito rápida" [ibid.].

A segunda, designada por "Active Server Pages searching", de facto destina-se a ilustrar a utilização dos objectos *Query* e *Utility*. Estes objectos podem ser utilizados noutros ambientes Microsoft, mais vulgarizados, que utilizam interpretadores de scripting, para além das ASPs. Nesse conjunto de ambientes inclui-se toda a família MS Office, designadamente o MS Access e MS Excel[75], bem como os interpretadores de *scripting* nativos do Windows 2000/XP[76]. Em [MSDN02] é detalhadamente explicada a construção de um Query utilizando a linguagem JScript. O primeiro caso do Anexo I é uma adaptação deste para JScript em ASP. Destes objectos, após o estabelecimento das suas propriedades resulta um *recordset* ADO.

O terceiro exemplo, designado por "ActiveX Data Objects (ADO) searching" recorre à possibilidade que o IS oferece de se constituir como um fornecedor de informação do tipo OLE

[73] Poder-se-ia também pesquisar com o utilitário Start/Search, onde o "Indexing Service está exposto" [TECHNET01], mas são sensíveis diferenças de desempenho na pesquisa de documentos quando se estabelecem critérios relativos ao conteúdo. A consulta pelo Indexing Service Query Form tem-se revelado muito mais rápida nessas condições.
[74] Cf. Sec. "One Catalog or Multiple Catalogs?".
[75] Por exemplo no MS Word 2000 a correr sobre Windows 2000 com o IS instalado, isto pode ser verificado. Acedendo a Macros e escolhendo o Visual Basic Editor, podem ser adicionadas as referências a estes objectos. Para os adicionar escolhe-se Tools > References e coloca-se um visto sobre "ixsso Control Library". Ficam disponíveis os objectos, podendo ser visualizados no Object Browser ou utilizados na programação dos módulos.

DB. Isto significa que pode ser utilizada a linguagem SQL para consultar a base de dados do IS, linguagem essa mais vocacionada para desenvolvimento em ambientes típicos de base de dados. Em [TECHNET02] são oferecidos alguns exemplos que ilustram bem este aspecto, utilizando o MS SQL Server 7.0. Após a ligação ao servidor OLE DB este exemplo também resulta na criação de um *recordset* ADO. No Anexo I é ilustrada também uma implementação utilizando esta modalidade.

Dos dois últimos casos resulta um objecto ADO. A diferença entre o segundo e o terceiro exemplos reside na forma como o *recordset* ADO é criado. Adicionalmente a utilização dos objectos *Query* e *Utility* fica condicionada a "determinadas combinações da linguagem de consulta" do IS; em contrapartida a linguagem SQL "incorpora diversas características únicas" do IS [FEDOROV98].

4.3.4. Recuperação de algumas propriedades úteis

Embora haja diferenças importantes entre as diferentes modalidades de recuperação de dados do IS, há aspectos essenciais que acabam por ser comuns. E ainda mais simples se torna essa identificação se o que se pretende é conhecido desde o início.

Considere-se o exercício seguinte[77]: identificar todos os documentos com a palavra "barra" e associar, a cada documento identificado, o número de ocorrências dessa palavra, o *ranking* da consulta e a dimensão do ficheiro. Para efectuar uma tal consulta chega-se à seguinte expressão SQL executada em ASP, como argumento para a criação de um recordset ADO (v. Anexo I):

```
SELECT path, HitCount, Rank, Size FROM System..SCOPE('DEEP TRAVERSAL OF
"C:\My Documents\Estudo\Textos\C1821\"') WHERE CONTAINS (Contents,'barra') .
```

Executando esta query, obtém-se uma lista, da qual se apresentam apenas 5 linhas, de um total de 151 no conjunto de documentos de teste:

```
1 - c:\my documents\estudo\textos\c1821\a1822\m11\d02\a1822m11d02-0972.htm : 1 : 80 : 6077
2 - c:\my documents\estudo\textos\c1821\a1822\m10\d31\a1822m10d31-0947.htm : 2 : 160 : 6304
3 - c:\my documents\estudo\textos\c1821\a1822\m10\d28\a1822m10d28-0894.htm : 2 : 160 : 6001
4 - c:\my documents\estudo\textos\c1821\a1822\m10\d28\a1822m10d28-0896.htm : 2 : 80 : 6524
5 - c:\my documents\estudo\textos\c1821\a1822\m10\d15\a1822m10d15-0792.htm : 1 : 80 : 5996
...
```

A expressão SQL apresentada, contém elementos que são nomeados pela Microsoft como propriedades. Em [MSDN04], relativa à documentação do SQL Server 2000, existe uma tabela que identifica e descreve brevemente o significado de cada uma. De facto podem ser encontradas listas semelhantes para cada uma das metodologias de *query* atrás descritas.

Analisando a query SQL, verifica-se que a sintaxe é muito semelhante à de outras *queries* ou seja,

[76] Pode encontrar-se uma descrição genérica sobre Scripting no Help do Windows 2000, sob o título "Windows Script Host overview".

SELECT <lista de colunas> FROM <origem dos dados> WHERE <condição>.

A *lista de colunas* está preenchida com as designações das propriedades que se pretende extrair. São elas: path, HitCount, Rank e Size. Na lista resultante pode verificar-se , à excepção do número de ordem e hífen iniciais, o conteúdo de cada uma destas propriedades.

Na *origem dos dados*, no habitual lugar de uma tabela, está uma expressão mais complicada: System..SCOPE('DEEP TRAVERSAL OF "C:\My Documents\Estudo\ Textos\C1821\"'). Esta expressão significa que a consulta irá incidir sobre o *catalog* System, abrangendo todos os ficheiros localizados no subdirectório indicado e subordinados.

A *condição* é o critério de escolha estabelecido. Neste caso CONTAINS (Contents,'barra') é satisfeito exclusivamente para os documentos que, na propriedade *Contents*, apresentem a palavra "barra". Para documentos que contenham palavras derivadas, o critério não é satisfeito e não serão exibidos por esse motivo.

Em [MSDN03d] está descrita a sintaxe completa do comando SELECT face ao IS. Também apresenta exemplos e ligações que descrevem com detalhe, nomeadamente, as clausulas FROM e WHERE.

A Tabela 5 é um fragmento da lista de propriedades apresentada em [MSDN04]. Nem todas as propriedades aqui relacionadas têm interesse para a classificação de documentos. Como se ilustrou no capítulo 3, é necessário começar por contabilizar a frequência de palavras por documento, ou seja *tf*. Para efectuar os cálculos de ponderação, determinar o *idf*, pode ser necessário obter o número de documentos em que a palavra ocorre no total de documentos da colecção. Finalmente, poderá ser necessário incluir, nos cálculos, um factor de normalização. Tal factor pode ser calculado a partir dos factores *tf* e *idf* , mas também pode ser determinado a partir da dimensão em *bytes* do ficheiro. Perante estas considerações, as propriedades "HitCount" e "Size" são importantes para a classificação de documentos.

Há, na Tabela 5, outras propriedades. Poder-se-ia dizer que "DocWordCount" seria útil em substituição de "Size". Mas, examinando a tabela, não poderá ser listado. Aliás, considerando a remoção de palavras ruidosas, é discutível a sua utilidade. "Filename" e "Directory" podem ser listadas, mas "Path" substitui-as com vantagem. "Path" pode, por sua vez, ser substituída por "FileIndex", uma forma mais compacta e utilizável como chave de indexação. O "Rank", poderá ter algum interesse, mas apenas como referência em *queries*. Eventualmente poderá ser interessante abordar esta variável na perspectiva de classificação por *keywords*.

Resumindo, e sabendo que "Contents" é a propriedade sobre da qual se referem as palavras, identificam-se como necessárias para referenciar os documentos, as propriedades "Path" e "FileIndex".

[77] v. também Anexo V

Property name	Data type	Description	Use in ORDER BY	Use in select list
Contents	nvarchar or ntext	Main contents of the file.		
Directory	nvarchar	Physical path to the file, not including the file name.	Yes	Yes
DocWordCount	integer	Number of words in the document.	Yes	-
FileIndex	decimal(19,0)	Unique identifier of the file.	Yes	Yes
FileName	nvarchar	Name of the file.	Yes	Yes
HitCount	integer	Number of words matching query.	Yes	Yes
Path	nvarchar	Full physical path to the file, including file name.	Yes	Yes
Rank	integer	Value from 0 to 1000 indicating how closely this row matches the selection criteria.	Yes	Yes
Size	decimal(19,0)	Size of file, in bytes.	Yes	Yes

Tabela 5 - Algumas propriedades úteis

4.4. Conclusões do capítulo 4

O serviço de indexação da Microsoft é um sistema robusto, não requerendo frequentes intervenções de correcção.

Não sendo um SGBD, apresenta uma grande variedade de possibilidades de resposta a consultas. Integra-se, naturalmente, na generalidade das aplicações mais comuns da Microsoft.

Tem características que permitem retirar dados úteis à classificação de documentos.

Capítulo 5. Ajustamento das técnicas à sintaxe do IS

Há diferenças entre as motivações subjacentes ao estudo de técnicas de processamento de linguagem natural por um lado, e do *Index Server* por outro. No primeiro caso, muito embora esse aspecto não tenha sido estudado com detalhe, pode afirmar-se que se inserem num âmbito maioritariamente académico, significando uma ampla colecção de objectivos, visões e contextos. O *Index Server*, pelo contrário, é desenvolvido para resolver uma necessidade concreta e que se coloca num ambiente técnico e cultural muito específicos.

Esta especificidade cultural não pode ser confundida com estreiteza de visão. Pelo contrário, o fabricante integra, no seu sistema de indexação, conceitos e conhecimentos identificados no domínio da *Information Retrieval*.

5.1. O IS comparado com sistema de IR

O *Indexing Service* é um sistema de *Information Retrieval*, como se pode verificar comparando as descrições de um e outro. Essencialmente pode dizer-se que:

- Está muito orientado para resposta a *queries*;
- linguagem de *query* muito flexível;
- integra mecanismos de eliminação de palavras ruidosas;
- não é um SGBD.

Cronologicamente, verifica-se que surge num contexto em que já existe uma sólida e ampla caracterização sobre o significado e a inserção de sistemas de *information retrieval*.

5.2. A correcção de erros

Conforme referido no capítulo anterior, há características no IS que suscitam alguma dúvida quanto à sua utilidade num contexto de detecção e correcção de erros ortográficos. Suscitam dúvida porque, se por um lado não são indexadas as palavras ruidosas, i. e., as que pertencem à “noise list”, todas as outras são guardadas quase intactas, ou seja, apenas normalizadas. As acções de *stemming* são empregues apenas num contexto do processamento da linha de comandos da *query*.

Mas estes aparentes impedimentos não devem ser tomados à letra. Começando pelo facto de a língua portuguesa não ser originalmente considerada, em si mesmo, não constitui uma dificuldade. Além disso, mesmo que o fosse, haverá sempre a possibilidade de restringir a “noise list” por forma a admitir palavras cuja importância em determinados contextos semânticos não possa ser ignorada.

É uma possibilidade a explorar, a localização de palavras que não fazem sentido quando embebidas num contexto, conceptual ou semântico, muito distante do seu significado. O operador de proximidade NEAR poderá ser útil na construção de um instrumento com esse objectivo.

5.3. O IS num contexto de classificação de documentos

A principal dificuldade aparente consiste no facto de não ser possível recorrer ao IS para obter os termos de classificação de documentos, palavras ou frases com origem directa no conteúdo da sua base de dados. A lista de propriedades de documentos que é possível utilizar nas consultas, é muito clara a esse respeito. Na descrição das propriedades “Contents”, “HtmlHref” e “HtmlHeading” é explicitamente dito que não pode ser obtido o respectivo conteúdo. Mas pode ser pesquisado, i.e., é um componente do critério de pesquisa.

Como se viu, no final do capítulo 3, para efectuar Análise de Clusters é necessário constituir conjuntos de palavras por documento ou o conjunto de todas as palavras do *corpus* que interessa estudar do ponto de vista da geração dos termos de classificação. O facto de não ser possível construir tais conjuntos a partir da base de dados do IS, em si mesmo, acaba por não representar uma dificuldade por aí além. Tais palavras podem ser extraídas dos documentos. Todavia não deixa de ser, pelo menos, trágico esse procedimento uma vez que o IS já o terá efectuado.

Em contrapartida, e conforme se examinou no final do capítulo 4, o IS disponibiliza propriedades cujo significado e importância é necessário considerar. Por exemplo: “HitCount” é o número de toques da consulta de um termo no documento; “Size” a dimensão do documento em bytes.

Em regra, o IS está configurado para assumir como *corpus* uma variedade muito grande de directórios, por vezes a totalidade do disco. Portanto, é necessário delimitar o *scope* de recolha dos termos o que é idêntico a dizer que se estabelece um directório como patamar superior de pesquisa. A definição de “Scope” simplifica a pesquisa de documentos a tratar, considerando que estão localizados abaixo desse directório.

5.4. Ficheiros inversos

Na secção 3.1 acima, menciona-se uma estrutura de dados designada por *Inverted Files*.

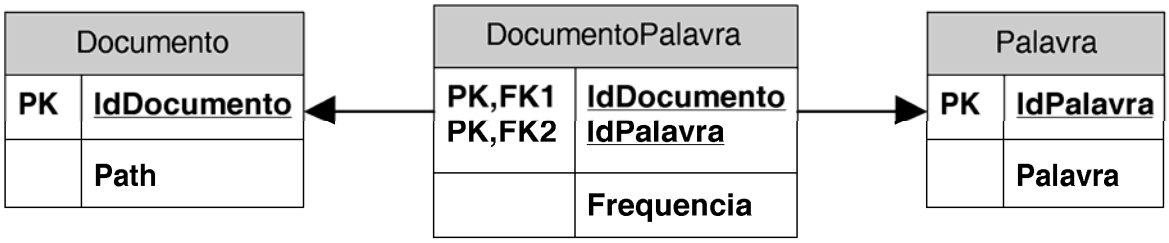

Figura 19 - Diagrama de Ficheiros Inversos

Esta estrutura consiste essencialmente na associação de palavras a documentos [FRAKES92: cap. 3]. Em termos de implementação (Figura 19), identifica-se uma tabela de palavras, uma tabela de documentos e uma tabela que as relaciona. Esta última contém as frequências de palavras em cada documento.

O IS pode ser visto como se utilizasse uma estrutura de dados semelhante. A principal dificuldade, nessa perspectiva, reside no facto de só a disponibilizar de uma forma recorrente. Como esta estrutura de dados é necessária para a classificação de documentos, o seu preenchimento ou a definição do seu acesso será um aspecto importante.

5.5. Duas modalidades de implementação de Ficheiros inversos

As palavras podem ser extraídas e contabilizadas lendo os documentos. Por outro lado, ao efectuar a leitura dos ficheiros, garante-se que estão acessíveis, sendo de esperar a correspondência com o resultado da pesquisa utilizando o IS, isto é, os ficheiros existem e não são apresentadas limitações de segurança ou outras. Ressalvando eventuais atrasos decorrentes do processo de indexação, é de esperar que tais documentos estejam também indexados.

A contabilização de palavras pode ser simplesmente efectuada de forma sequencial, identificando e contabilizando, individualmente, cada palavra em cada documento. É um processo de força bruta, cujo diagrama de fluxo está representado na Figura 20.

O IS dispõe de uma base de dados e fornece a frequência das palavras por documento conforme se ilustrou no capítulo 4. Isto significa que o algoritmo de contabilização de palavras pode ser alterado radicalmente, conforme diagrama de fluxo representado na Figura 21. Para cada palavra identificada e não pertencente ao conjunto de palavras ruidosas, verifica-se se já foi ou não contabilizada. Se não foi, cria-se um *recordset* a partir do IS, contendo as referências dos ficheiros – *Path*, *FileIndex* (cf. Tabela 5, sec. 4.3.4 acima) – e frequências respectivas – *HitCount* (ibid.). Com esse *recordset* completa-se o preenchimento da tabela de documentos e frequências de palavras. Na leitura de cada palavra, no caso de existir o tuplo {palavra, documento}, passa-se directamente à leitura da palavra seguinte, porque se presume que a contabilização respectiva está efectuada.

A implementação de uma solução deste tipo, como se verá no capítulo seguinte, é tão simples como descrito. Os problemas de desempenho que se verificam, na prática dessa implementação, prendem-se com a excessiva actividade imposta à base de dados de destino.

A implementação do paradigma de ficheiros inversos pode ser ainda mais simplificada. Pode considerar-se o IS com sendo uma base de dados que contem as tabelas Documento e PalavraDocumento (Figura 19).

Esta modalidade de aproximação à implementação tem a vantagem de fazer reflectir no IS as alterações produzidas nos termos dos documentos quase em tempo real (cf. Sec. 4.2.1 acima).

Mas, será sempre necessário, para este efeito, estabelecer um critério baseado em palavras obtidas a partir dos documentos, e garantir uma adequada gestão do respectivo refrescamento.

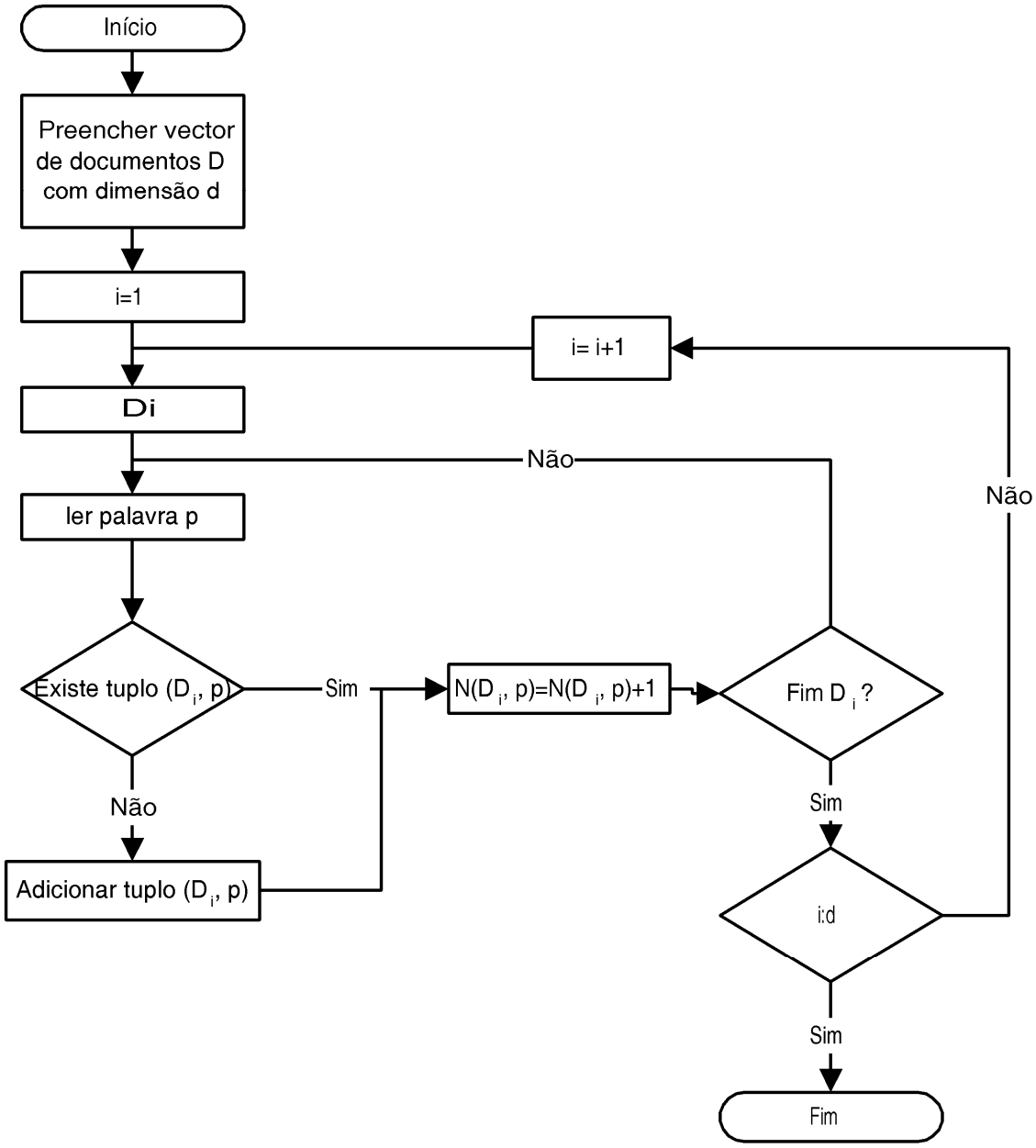

Figura 20 - Diagrama de fluxo do preenchimento de frequências à força bruta

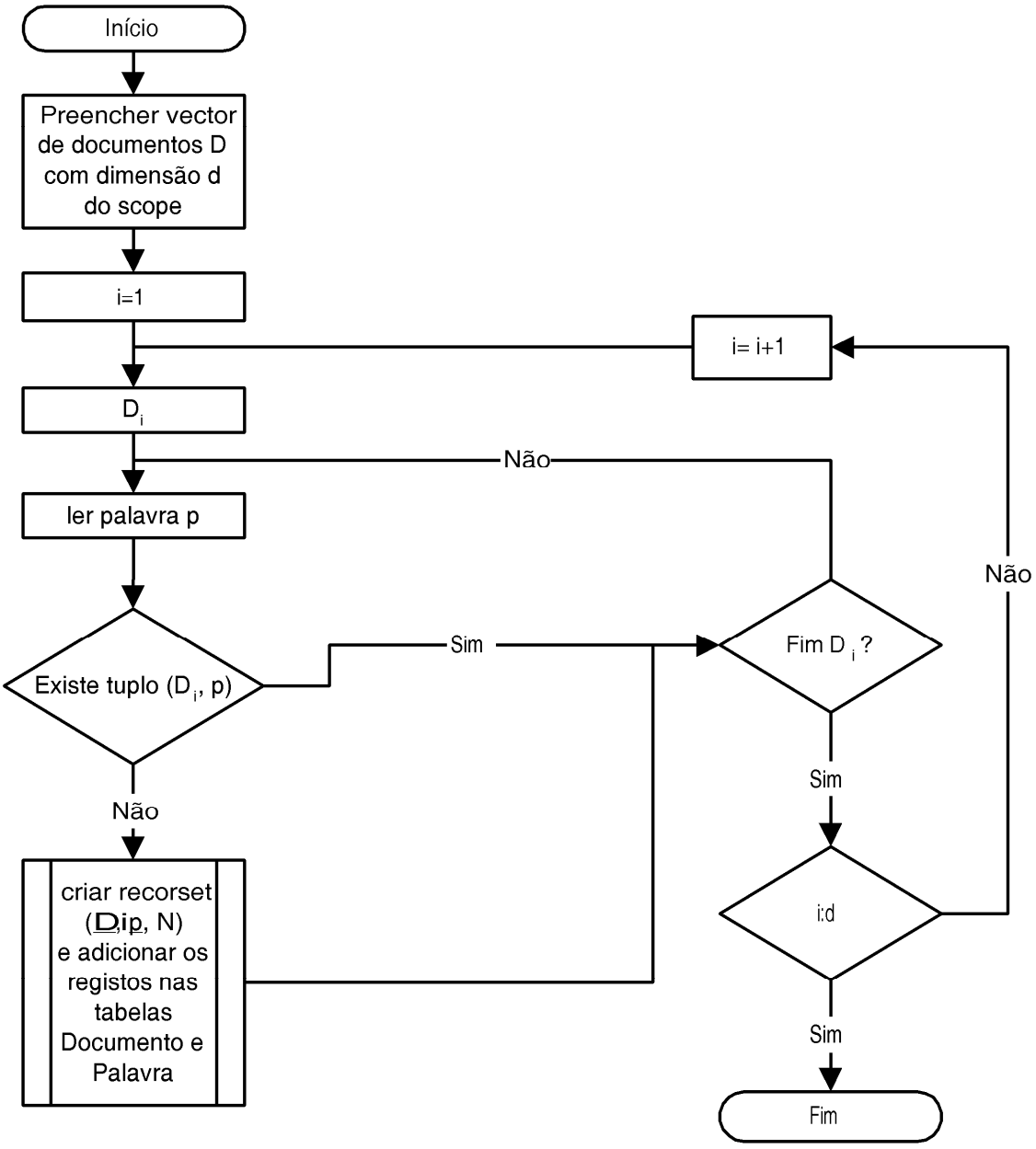

Figura 21 - Diagrama de fluxo do preenchimento de frequências de palavras usando o IS

5.6. Conclusões do capítulo 5

O *Indexing* Service é um sistema de *Information Retrieval*.

O *indexing service* pode ser utilizado para integrar soluções que implementem o paradigma de ficheiros inversos. Na secção 4.3.4 acima, no capítulo seguinte e no Anexo V, exemplifica-se essa possibilidade.

O principal inconveniente do *Indexing Service* consiste no facto de não disponibilizar o acesso à lista de palavras do corpo dos documentos indexados. Isto obriga à recuperaçã0 dos termos a partir dos documentos quando essa lista se torna necessária.

A sua utilização para localização de erros está condicionada a palavras que não pertençam à *stoplist*. Como, por definição, a “noise list” contem termos válidos, os restantes serão indexados, incluindo aqueles que correspondem a erros ortográficos. Desta forma o IS facilita a localização de documentos que contêm erros ortográficos.

Opções e razões

O presente trabalho é o resultado de um conjunto de opções. À partida, a única questão, que ao autor se colocava, consistia em determinar a natureza dos exemplos concretos. A experiência prática é uma das mais eficazes formas de ilustrar os conceitos afirmados. E, de facto, foi assim que se iniciou o presente trabalho. Mais abaixo, neste capítulo, são brevemente relatados alguns ensaios práticos efectuados ainda com essa perspectiva.

O trabalho de investigação começou, portanto, com duas actividades paralelas. Por um lado, como se referiu, uma actividade prática voltada para a determinação de frequências de termos nos documentos. Por outro lado, uma actividade de pesquisa de elementos bibliográficos e de referência relativamente ao "Indexing Service", à classificação de documentos e localização de erros ortográficos. Adicionalmente, e seguindo a sugestão dos professores orientadores, foram pesquisadas referências a redes neuronais e “text mining”.

Desde logo começam a surgir algumas dificuldades. A documentação disponível na Internet sobre o "Indexing Service" é muito reduzida à data de início da investigação, por alturas de Outubro/Novembro de 2002. Por outro lado, aparece dispersa na documentação relativa a outros produtos do mesmo fabricante. Esse panorama melhorou um pouco em inícios de 2003, cujos efeitos só se fizeram sentir passados vários meses[78].

Mas as imprecisões e omissões serão talvez o aspecto mais grave na informação disponibilizada pelo fabricante. Na secção 4.3.1 descreve-se um ilustrativo exemplo de imprecisão de linguagem. Podem referir-se, também, a sucessão de quatro artigos iniciados com [NAREDDY97a], como simpáticos e úteis, mas exigiram um trabalho cuidadoso, especialmente no que se refere ao cruzamento e confirmação da informação a utilizar. É também notória a omissão de informação clara sobre a forma de programar a utilização do IS em ambientes de programação diferentes daqueles que são propriedade do fabricante, ex. Java.

Pode identificar-se, também, uma importante diferença cultural: o que será trivial para os autores dos artigos publicados pelo fabricante, não é trivial para o autor do presente trabalho. Este aspecto é importante, porque o autor do presente trabalho não é um experimentado programador em linguagens utilizadas nas soluções para a Web, designadamente em ASP. Além disso, como no parágrafo seguinte se descreve, apareciam, dispersos pela documentação, conceitos e respectivas definições cujo enquadramento, numa primeira leitura, era completamente desconhecido. O facto de, na literatura do fabricante, não se mencionarem referências que

[78] Note-se que as datas que constam na Bibliografia, a título de data de consulta, se reportam à última visita efectuada ao documento de que, nalguns casos, resultou a respectiva impressão em papel.

confirmem esses conceitos, conduz à ideia de que os autores, dessa literatura, estarão envoltos num ambiente técnico e cultural diferente, onde esses conceitos farão parte do quotidiano. Estas diferenças de domínio científico, técnico e cultural, foram observadas, também, através da leitura das restantes obras de referência.

Assim, à medida que a investigação prosseguia no sentido de encontrar soluções para os problemas colocados, surgiam novos conceitos e definições, cuja inserção, em termos de formação de base do autor, era completamente desconhecida e inesperada. Estabeleceram-se, portanto, novas metas "ocultas", de difícil delimitação, ligadas à arrumação dos conhecimentos emergentes. Neste contexto identificaram-se duas vertentes de conhecimento que, mais tarde, vieram a receber, no contexto do presente trabalho, designações mais próximas do que se identificou como quadro de definição da sua abrangência: "Estatística Multivariada – Multivariate Statistics" e "Recuperação de Informação - Information Retrieval". Adicionalmente, a vertente das "Redes Neuronais" era também completamente desconhecida do autor, mas estava identificada à partida.

Optou-se por abandonar qualquer desenvolvimento prático, exceptuando alguns casos pontuais, simples e exigíveis. Esta opção representou um esforço efectivo para conseguir conter a veia experimentalista do autor. Mas, qualquer incursão na elaboração de aplicações práticas, representaria mais um perigoso factor de dispersão. De facto desenhava-se claramente um discurso teórico.

Alguns resultados experimentais

Aproximações ao tratamento estatístico de palavras

No capítulo 5 referiu-se uma estrutura de dados conhecida como *Inverted File*, para a qual está esboçada, na Figura 19, uma possível representação, neste caso extraída do quadro de relações da base de dados em MS Access. De facto, uma das primeiras actividades inseridas no âmbito do presente trabalho, consistiu em elaborar uma base de dados, em MS Access, com uma estrutura semelhante[79], conforme esboçado na Figura 22.

A única diferença consiste na existência de uma tabela adicional – PalavraOcorrencias –, destinada a receber a frequência total de cada palavra na colecção de documentos. Esta tabela seria perfeitamente dispensável dado que tais totalizações podem ser o resultado de uma consulta de agregação sobre a tabela de frequências de palavra por documento, cuja programação em SQL é relativamente simples e, ainda por cima, no Access, é praticamente oferecida. Todavia, na previsão de que poderia haver problemas de desempenho na execução de uma tal "query" em tempo real, optou-se por criar uma tabela adicional.

[79] Na verdade, foi a partir da base de dados criada que se extraiu o "Diagrama de Ficheiros Inversos" referido, Figura 19.

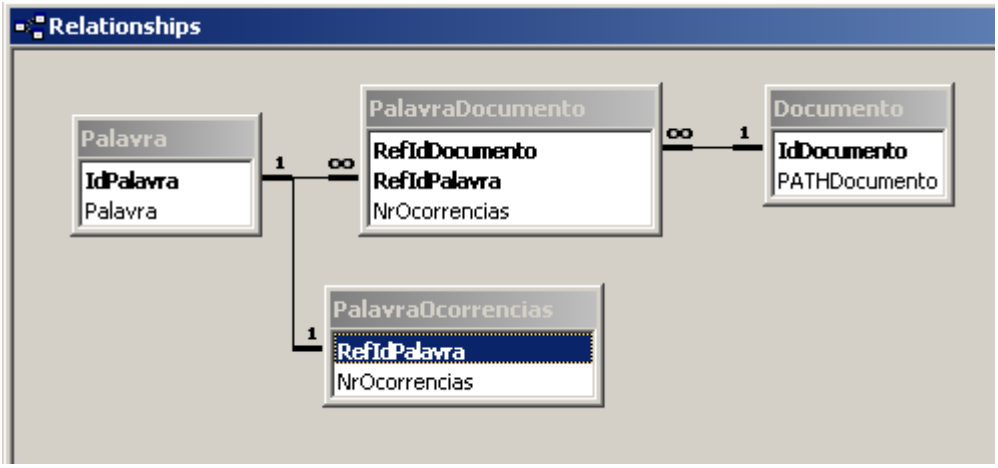

Figura 22 - Diagrama de relações entre tabelas na BD

O preenchimento desta base de dados foi efectuado utilizando um pequeno programa em Java (Anexo VI). A única preocupação na codificação realizada consistiu em garantir alguma fiabilidade no preenchimento. O tempo de execução deste programa é, no entanto, muito elevado: entre 10 a 25 segundos por documento de 7 a 8k. Presume-se, todavia, que é possível melhorar substancialmente o desempenho, considerando outras opções de programação e de salvaguarda dos resultados.

Desta forma foram preenchidas duas bases de dados, a partir de duas colecções de documentos diferentes. A primeira consistiu numa colecção 4172 ficheiros correspondentes às *Actas da Câmara Corporativa* entre 2 de Janeiro de 1954 e 24 de Novembro de 1966. A segunda consistiu numa colecção de 6774 ficheiros pertencentes ao *Diario das Cortes Geraes e Extraordinarias da Nação Portugueza* de 27 de Janeiro de 1821 a 4 de Novembro de 1822.

Apesar de, como acima se referiu, ter sido abandonada qualquer perspectiva de análise objectiva e séria sobre os dados recolhidos desta forma, foram elaborados dois formulários orientados para ilustrar dois aspectos muito concretos: distribuição de frequências de termos na colecção e sua representação gráfica; relação de termos por intervalos de frequência. Este último formulário foi ampliado por forma a apresentar também a relação de documentos e frequências (por documento) do termo seleccionado.

A forma da distribuição de frequências de termos na colecção é algo hiperbólica. A Figura 23 é um gráfico retirado a partir dos termos presentes na colecção de textos do *Diario das Cortes Geraes e Extraordinarias da Nação Portugueza*. Com esta distribuição será relativamente simples intuir o que se considera palavras ruidosas na colecção, e um patamar de termos a utilizar como relevantes num contexto de classificação.

Por outro lado, o formulário correspondente à distribuição de termos apresenta uma lista de palavras por ordem alfabética. Na sequência de actuação do botão direito do rato sobre um termo escolhido, é apresentada a lista de ficheiros onde esse termo está presente.

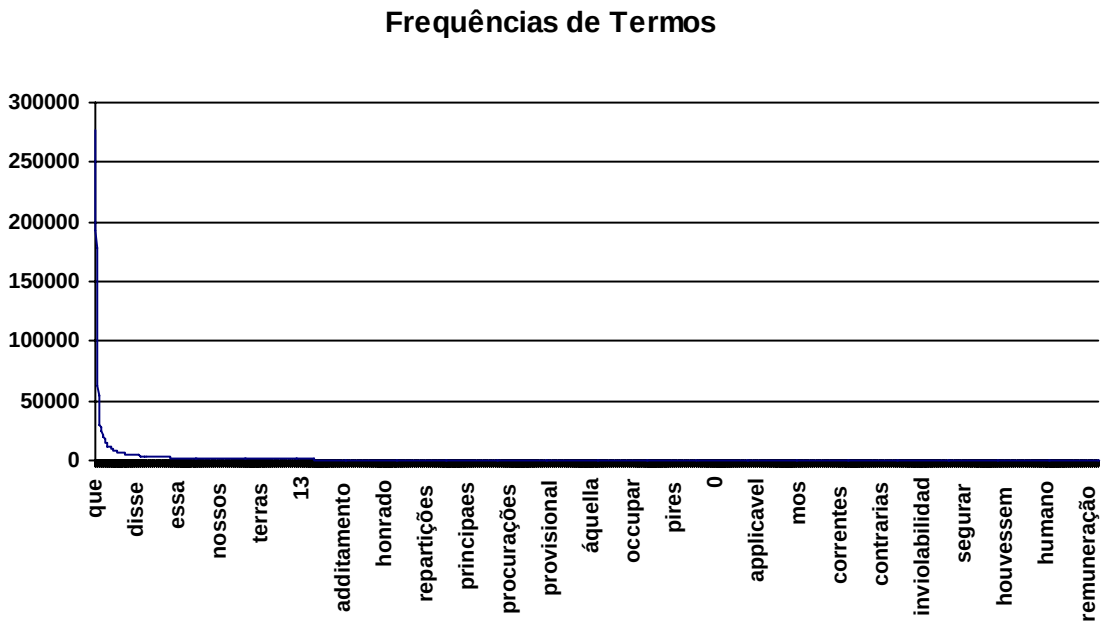

Figura 23 - Gráfico de distribuição de frequências de termos na colecção

No exemplo ilustrado na Figura 24 há alguns aspectos a considerar. A lista de termos apresentada por ordem alfabética é o resultado da imposição de um intervalo de frequências correspondente ao máximo possível no caso ilustrado. Portanto, não será a condição de utilização mais frequente. Por razões de distribuição de espaço no formulário não se apresenta a barra de controlo do formulário embebido com a lista de palavras. O número de palavras listadas pode ser visualizado brevemente colocando o ponteiro do rato sobre o cursor da barra elevadora respectiva, conforme se ilustra.

No formulário da Figura 24, o intervalo de frequências estabelecido admite valores iguais como limites máximo e mínimo. Desta forma é possível examinar, por exemplo, termos que aparecem apenas uma ou duas vezes na colecção. De facto, um exame aos termos listados nestas condições permite inferir que os termos com baixa frequência na colecção serão pouco úteis num contexto de classificação. Por exemplo, um termo com frequência unitária na colecção significa a sua presença apenas num ficheiro. Isto quer dizer que, ao estabelecer o conjunto de termos relevantes num contexto de classificação, como acima se referiu, é possível e necessário estabelecer não só um limite máximo como um limite mínimo de frequências, baseados em considerações que resultarão da análise efectuada nestas condições.

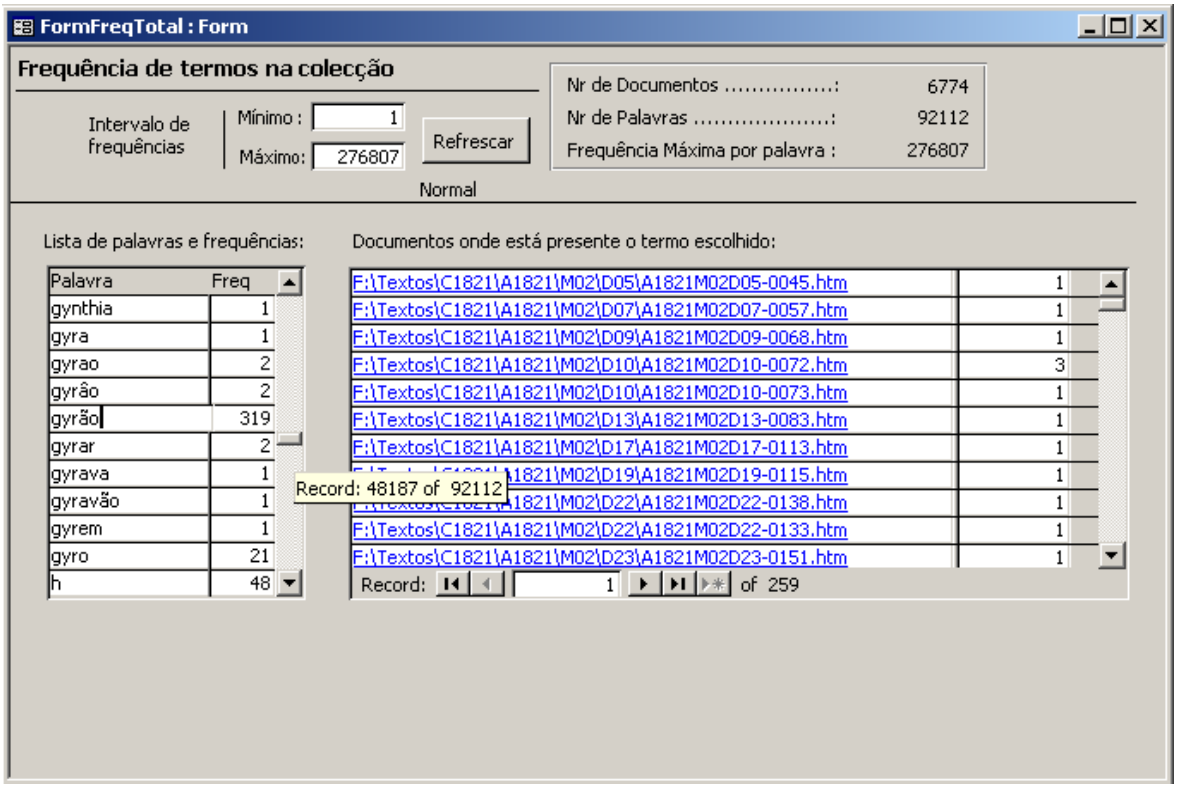

Figura 24 - Formulário de frequência de termos na colecção

Por outro lado, numa perspectiva de localização de erros, é de esperar que a lista de termos com frequência unitária ou da mesma ordem de grandeza, apresente uma maior proporção de erros ortográficos. Observando os termos extraídos da referida amostra de textos do "Diario das Cortes Geraes e Extraordinarias da Nação Portugueza", verifica-se, com facilidade, a presença de erros ortográficos, ao listar termos com frequência unitária. Todavia, também se testemunha a variedade ortográfica que vigorava em princípios do século XIX.

Experimentação de preenchimento utilizando o IS

Uma questão que naturalmente se coloca no âmbito do presente trabalho, é a da possibilidade de utilização prática dos objectos de consulta disponibilizados pelo IS. Esta questão foi abordada na secção 4.3.4 acima e retomada na 5.5 acima, aparecendo ilustrada, no Anexo V, com um pequeno exemplo em "Visual Basic for Applications" (VBA) sob MS Access.

No contexto do presente trabalho interessava demonstrar, objectivamente e apenas, uma forma de integrar o IS para preenchimento de ficheiros inversos no Access, conforme discutido na secção 5.5 acima. Considerou-se que o processamento seria efectuado utilizando apenas os recursos do MS Access e do IS. Este é de facto um ambiente de uma estação de trabalho perfeitamente habitual, com um ambiente tipicamente Microsoft, Windows (2000 Professional) e MS Office 2000. Não foi ensaiado com versões mais actuais do MS Office.

O problema do preenchimento da base de dados foi colocado de forma diferente da ilustrada na Figura 21 - Diagrama de fluxo do preenchimento de frequências de palavras usando o IS, da secção 5.5 acima. Optou-se por dividir a tarefa em três fases individualizadas: preenchimento da Tabela Documento; preenchimento da Tabela Palavra; preenchimento da Tabela de associação PalavraDocumento. A tabela PalavraOcorrencias é ignorada neste caso, para simplificar ainda mais.

Esta divisão em três fases tem a vantagem de evidenciar as actividades que utilizarão o IS. Apenas o preenchimento da tabela Palavra não o fará. Trata-se de uma questão externa à utilização do IS. Em alternativa, o preenchimento da tabela Palavra, pode ser efectuado por importação de outra tabela existente com as palavras da colecção. Nos ensaios efectuados foi esta a opção tomada pelo autor.

O preenchimento da tabela Documento é um procedimento simples, cuja execução é rápida mas parece apresentar alguma hesitação inicial, correspondente à criação dos objectos de acesso ao IS. Por sua vez, o preenchimento da tabela de associações é muito mais pesado e demorado. Há que considerar um conjunto de palavras que estão presentes quase na totalidade dos documentos. À medida que se progride na tabela de palavras, o número de documentos onde a palavra está presente, tendencialmente, vai diminuindo. Isto significa que o ritmo de inserção de associações vai acelerando. Observa-se, também, que são ignorados os processamentos relativos a termos que pertencem à “noise list” [cf. p. 36].

Este ensaio de preenchimento da tabela de associação, como se referiu, é pesado e demorado. Esta lentidão está associada às soluções utilizadas para preenchimento da base de dados em Access. Coincide, em grande medida, com a lentidão verificada na solução de preenchimento da base de dados com programa em Java, referida na secção anterior (Anexo VI).

Conclusões

A classificação de documentos e a correcção de erros são dois problemas radicalmente diferentes no que se refere ao detalhe com que os documentos são percepcionados. Portanto, perante os documentos, identificam-se duas atitudes diferentes a adoptar nas acções de classificação ou na correcção de erros. Esta diferença de atitude determina as características dos agentes que executarão quer uma quer outra actividade.

As redes neuronais, surgindo com uma vocação para solucionar problemas complexos, não serão instrumentos a excluir num contexto de busca de soluções que visem a detecção de erros. Identificaram-se alguns trabalhos de investigação com redes neuronais, que procuram produzir representações sintácticas e semânticas de pequenas frases. Considerando a situação em que o erro ortográfico está camuflado em palavra pertencente a um glossário, será legítimo procurar conferir a validade sintáctica e semântica da palavra no contexto da frase. Poderá ser este um dos quadros possíveis de emprego de soluções com redes neuronais. O problema da

classificação de documentos também poderá ser um campo de aplicação de soluções com redes neuronais, muito embora essa perspectiva não tenha sido explorada no contexto do presente trabalho.

A preocupação dos sistemas de "Information Retrieval" é facilitar a localização de documentos. Criam-se, com esse objectivo, sistemas de índices em que o desempenho na resposta às consultas é a principal prioridade. O sistema de índices, criado pelo sistema de "information retrieval", é uma base de dados. Mas o sistema de "information retrieval" não é um sistema de gestão de base de dados. Os documentos e as consultas podem ser representados por vectores de termos com variados esquemas de ponderação. Documentos com erros ortográficos são tratados como quaisquer outros, não havendo distinção entre palavras bem ou mal escritas. A classificação de documentos pode ser efectuada metodicamente recorrendo à análise de clusters.

O serviço de indexação da Microsoft é um sistema robusto, não requerendo frequentes intervenções de correcção. Não sendo um SGBD, apresenta uma grande variedade de possibilidades de resposta a consultas. Tem características que permitem retirar dados necessários para a classificação de documentos.

A sua utilização para localização de erros está condicionada a palavras que não pertençam à *stoplist* [cf. p. 36]. Como, por definição, a "noise list" [ibid.] contém palavras válidas, as restantes serão indexadas, incluindo aquelas que correspondem a erros ortográficos. Desta forma o IS facilita a localização de documentos que contêm erros ortográficos.

Trabalho para o futuro

O presente trabalho conclui-se com a certeza de que fica muito por fazer em torno das temáticas suscitadas. Ao longo do discurso foram dadas respostas mas, também, restaram questões. Omitiram-se e foram retirados assuntos, aspectos e reflexões, por insuficiência de informação, por se considerarem irrelevantes ou, simplesmente, por redundância. Foi necessário "conter a veia experimentalista do autor" [80].

A Língua Portuguesa - como qualquer outra forma de expressão o será - é um terreno riquíssimo, um património. A sua exploração não deve ser deixada ao arbítrio de outros, que não sejam os seus falantes. A criação, manutenção e divulgação de centros ou bibliotecas de recursos de Língua Portuguesa na Internet, é um trilho. Esse caminho tende a alargar-se para auto-estrada, com áreas de serviço e tudo. Seria interessante observar, em planos curriculares e em opções de engenharia, a integração, intencional e aberta, da língua portuguesa como terreno de exploração. Neste contexto, o problema da localização de erros em textos, resultantes da aquisição por OCR ou não, seria uma questão que poderia ser colocada como desafio. Seria também um interessante pretexto para reunir, em convívio cooperante, diversas áreas do saber.

A questão do tratamento de erros não foi abordada do ponto de vista estatístico. Não se desenhou, por exemplo, um perfil de probabilidade de ocorrência de erros em função da frequência de termos presentes na colecção. Não se mediu a importância que a proximidade gráfica produz em trocas de caracteres. Estas e outras avaliações de natureza estatística, por si só, poderão justificar a criação de instrumentos com alguma utilidade.

A primeira questão, a do perfil de probabilidade, em certa medida é uma questão resolvida do ponto de vista da modelação, considerando o formulário representado na Figura 24 - Formulário de frequência de termos na colecção – e a base de dados que a suporta. Faltar-lhe-á uma ligação a um glossário, um ou outro atributo de confirmação e conferência, enfim, aquilo que se julgar mais conveniente e facilite a utilização prática, ou seja, uma especificação precisa.

A segunda questão, a da proximidade gráfica, já não terá solução tão simples. Admitindo a preexistência de um glossário de termos precisos e controlados, uma dificuldade considerando a ortografia do séc. XIX, pode ser efectuada uma contabilização prévia da estatística dos termos do glossário utilizando o "Indexing Service". Essa contabilização guarda-se para referência. Posteriormente, lançam-se consultas de conferência por termo modificado, substituindo individualmente cada caractere por um ponto de interrogação, "?" (cf. "Regular expression operators" no Help do IS). Se não houver termos errados nos documentos, presume-se que a cardinalidade do conjunto de documentos obtido, será menor ou igual à cardinalidade da reunião dos conjuntos de documentos de referência dos termos ambíguos. Por documento, o "HitCount" (número de coincidências) será igual, assim como a contabilização total de termos. Neste contexto, termos ambíguos são aqueles que se tornam idênticos na falta de um caractere, ex. "BARRA" e "BARBA", sem o penúltimo caractere, não se distinguem.

Ainda relacionada com a localização de erros, a questão mais complexa da análise da qualidade da construção sintáctica pode ser abordada também numa perspectiva heurística, dedicada e especializada. Todavia talvez se torne um caminho desgastante e pouco versátil.

Foi por esta via, da pesquisa de instrumentos versáteis que visem a análise da conformidade sintáctica, que se julgou de interesse estudar e referir investigações que empregam redes neuronais. Apesar de, no contexto do presente trabalho, o que se referiu sobre redes neuronais não representar soluções imediatas e concretas, também não se fica convencido que tais redes não possam integrar, total ou parcialmente, instrumentos de localização de erros e de classificação de documentos. É um assunto a aprofundar.

A classificação de documentos recorrendo a análise de clusters foi uma das experimentações abandonadas pelas razões descritas no início deste capítulo. A escolha dos termos, apesar de um conjunto de atalhos e aproximações, é, em si mesma, uma tarefa demorada e exige atenção. Conforme a panóplia de opções descritas no capítulo 3, este assunto pode dar origem a interessantes e úteis aplicações práticas.

---

[80] Cf. p. 68.

Finalmente restam alguns caminhos não trilhados ao longo da linha do discurso. Coisas pequenas, mas importantes. Na sec. 4.3.3 acima referiram-se as modalidades de consulta “Static Search” e recurso à utilização dos objectos *Query* e *Utility*. Tanto uma como outra, serão soluções de consulta a considerar. Não foram tão detalhadamente utilizadas porque, em termos de apresentação gráfica e talvez porque o autor esteja mais familiarizado com a linguagem SQL, a modalidade ADO pareceu fornecer uma representação mais adequada para a exposição. Considerando a simplicidade da consulta exposta na sec. 4.3.4, que suporta a utilização do termo como chave para criação de um *recordset* com o *Path* e a frequência do termo nos documentos, este objectivo pode ser conseguido com as outras modalidades.

Um outro aspecto prende-se com a obtenção da propriedade *FileIndex* em lugar da propriedade *Path*. Trata-se de um número que pode funcionar como chave na tabela de Documento, evitando compromissos e demoras no preenchimento ou utilização dessa tabela e da tabela de associação.

Bibliografia

[ALLEN99]

ALLEN, Joseph; SEIDENBERG, Mark S. (1999) - The Emergence of Grammaticality in Connectionist Networks [Em Linha] (In B. Macwhinney (ed.), **Emergentist approaches to language: Proc. of the 28th Carnegie Symposium on cognition,** Lawrence Erlbaum Associates) [Consult. em 2 Set. 2003] Disponível na WWW: <http://citeseer.nj.nec.com/allen99emergence.html. >

[AR]

Assembleia da República [Em linha, consultado em 5 Jun. 2003]. Disponível na WWW: <www.parlamento.pt. >

[ARDEBATESINTRO]

Assembleia da República - **Introdução e Período histórico tratado** [Em linha, consultado em 6 Fev. 2004]. Disponível na WWW: <http://debates.parlamento.pt/1_pag/introducao.asp. >

[ARDEBATESMC]

Assembleia da República - **Cortes Geraes e Extraordinárias da Nação Portugueza (1821) - Apresentação do catálogo** [Em linha, consultado em 6 Fev. 2004]. Disponível na WWW: <http://debates.parlamento.pt/mc/c1821/. >

[BERG92]

BERG, George (1992) - A Connectionist Parser with Recursive Sentence Structure and Lexical Disambiguation [Em linha]. (Também in: **Proceedings of AAAI-92, American Association for Artificial Intelligence**, pp 32-37). [Consult. em 3 Nov. 2003] Disponível na WWW:<http://citeseer.nj.nec.com/berg92connectionist.html. >

[BRYANT01]

BRYANT, Bobby D.; MIIKKULAINEN, Risto (2001)- **From Word Stream to Gestalt: A Direct Semantic Parse for Complex Sentences** [Em Linha; Consult. em 4 Nov. 2003], Disponível na WWW: <http://nn.cs.utexas.edu/downloads/papers/bryant.utcstr98.pdf. >

[CUNHA02]

CUNHA, Celso; CINTRA, Lindley (2002) - **Breve Gramática do Português Contemporâneo**, 15ª Edição, Lisboa: Edições João Sá da Costa. ISBN: 972-9230-05-6 (Edição original 1985)

[DHILLON00]

DHILLON, Inderjit S.; MODHA, Dharmendra S. (2000) - Concept Decompositions for Large Sparse Text Data Using Clustering [Em linha] **Machine Learning**, 42:1, pages 143-175, January, 2001.[Consult. em 18 Ago. 2003] Disponível na WWW: <http://www.cs.utexas.edu/users/inderjit/public_papers/concept_mlj.pdf. >

[DHILLON98]

DHILLON, Inderjit S.; MODHA, Dharmendra S.; SPANGLER, W. Scott (1998) - Visualizing Class Structure of Multidimensional Data. [Em linha] (In **Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics**, Interface Foundation of North America, vol. 30, pages 488-493, Minneapolis, May, 1998.) [Consult. em 4 Jul. 2003] Disponível na WWW: <http://www.cs.berkeley.edu/~inderjit/public_papers/interface98-color.ps.gz. >

[DHILLON99]

DHILLON, Inderjit S.; MODHA, Dharmendra S. (1999) - **Concept Decompositions for Large Sparse Text Data Using Clustering** [Em linha], Technical Report Research Report RJ 10147 (95022), IBM Almaden Research Center, July 8, 1999. [Consultado em 13 Nov. 2002] Disponível na WWW: <http://citeseer.nj.nec.com/dhillon99concept.html. >

[ECO98]

ECO, Umberto (1998) – **Como se faz uma tese em ciências Humanas**. 7ª Edição, Lisboa: Editorial Presença. ISBN: 972-23-1351-7

[ELMAN90]

ELMAN, Jeffrey L. (1990) - Finding Structure in Time [Em Linha]. (In: **Cognitive Science**, 14, 179-211) [Consult. em 6 Nov. 2003] Diponível na WWW: <http://crl.ucsd.edu/~elman/. >

[ESTRELA93]

ESTRELA, Edite (1993) – **A Questão Ortográfica – Reforma e Acordos da Língua Portuguesa**, Lisboa: Editorial Notícias. ISBN: 972-46-0611-2

[FAYAD96]

FAYAD, Usama M., et al. eds.(1996) - , **Advances in knowledge discovery and data mining**, The MIT Press, 1996

[FEDOROV98]

FEDOROV, Alex et al (1998) - “Chapter 19: Integrating Microsoft Index Server”[Em Linha] MSDN Home> MSDN Library> Active Server Pages (General) > Professional Active Server Pages 2.0 (Tb. in **Professional Active Server Pages 2.0**,., Dvlp Edt: Elston, Antea, Edt: Beacock, Jeremy, et al., Canada, Birmingham: Wrox Press Ltd., ISBN 1-861001-27-4) [Ult. Consult. em 14 Mar.2004] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/dnproasp2/html/integratingmicrosoftindexserver.asp?frame=true>

[FRAKES92]

FRAKES, William Bruce; BAEZA-YATES, Ricardo, org. (1992)-**Information Retrieval – Data Structures & Algorithms**, Upper Saddle River, New Jersey: Prentice Hall PTR, pp.viii-504

[FRÖHLICH97]

FRÖHLICH, Jochen (1997) – **Neural Networks with Java**, Neural Net Components in an Object Oriented Class Structure [Em linha, consultado em 21 Out. 2002], Disponível na WWW: <http://rfhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-index.html. >

[GUHA00]

GUHA, Sudipto; RASTOGI, Rajeev; SHIM, Kyuseok (2000) –Rock: A Robust Clustering Algorithm For Categorical Attributes [Em linha] (in **Information Systems**, Grã Bretanha: Pergamon Press, Vol. 25, No. 5, pp. 345-366), [Consultado em 31 Out. 2002] Disponível na WWW : <http://citeseer.ist.psu.edu/guha00rock.html. >

[HAYKIN94]

HAYKIN, Simon (1994) – **Neural Networks - A Comprehensive Foundation**, Nova Iorque: MacMillan Publishing Company Inc. ISBN: 0-02-352761-7

[HERTZ91]

HERTZ, John; KROGH, Anders; PALMER, Richard G. (1991) – **Introduction to the theory of neural computation**, Reading, Massachusetts: Addison-Wesley, pp. xxii-327.

[HOUAISS02]

Dicionário Houaiss de Língua Portuguesa Tomo I, Lisboa: Círculo de Leitores, 2002, ISBN: 972-42-2810-X (Obra completa com ISBN: 972-42-2809-6)

[HOUAISS03a]

Dicionário Houaiss de Língua Portuguesa Tomo III, Lisboa: Círculo de Leitores, 2003, ISBN: 972-42-2911-4 (Obra completa com ISBN: 972-42-2809-6)

[HOUAISS03b]

Dicionário Houaiss de Língua Portuguesa Tomo VI, Lisboa: Círculo de Leitores, 2003, ISBN: 972-42-3022-8 (Obra completa com ISBN: 972-42-2809-6)

[JAIN00]

JAIN, Anil K.; DUIN, Robert P. W.; MAO, Jianchang - "Statistical Pattern Recognition: A Review," [Em linha].(Também In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 22, no. 1, Jan. 2000, pp. 4-37), [Consultado em 3 Nov. 2003] Disponível na WWW: <http://www.ph.tn.tudelft.nl/People/bob/papers/pami_00_review.pdf.gz. >

[LANGENSCHEIDT60]

LANGENSCHEIDT (1960) - **Dicionário Universal Langenscheidt Inglês-Português Português-Inglês**, Berlim: Langenscheidt KG

[LAWRENCE00]

LAWRENCE, Steve; GILES, C. Lee; FONG, Sandiway (2000) - Natural language grammatical inference with recurrent neural networks [Em Linha] (in **IEEE Transactions on Knowledge and Data Engineering**, vol. 12, no. 1, pp. 126-140, 2000) [Consult. em 11 Fev. 2003] Disponível na WWW: <http://citeseer.nj.nec.com/lawrence98natural.html. >

[LEE98]

LEE, David (1998) – **Using HTML Meta Properties with Microsoft Index Server**, Microsoft, 1998, in MSDN Library, Technical Articles – Web Development – Microsoft Internet Information Server, [Consultado em 20 Nov. 2002] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/dnindex/html/msdn_ismeta.asp. >

[LIMA02a]

LIMA, Luciano R. S.; LAENDER, Alberto H. F.; JÚNIOR, Hermes R. de F. (2002a) - **MedCode: Uma Ferramenta Web para Classificação e Visualização de Documentos em Bases de Dados Médicas**, <http://www.avesta.com.br/anais/dados/trabalhos/181.pdf. >, 5.9.2002, Belo Horizonte –MG, última visita em 21.5.2003

[LIMA02b]

LIMA, Luciano R. S.; LAENDER, Alberto H. F.; JÚNIOR, Hermes R. de F. (2002b) - **MedCode: Uma Ferramenta para Classificação e Visualização de Documentos em Bases de Dados Médicas** [Em linha], Disponível na WWW: <http://www.avesta.com.br/so/so22_medcode.pdf. >, 22.9.2002, [Consult. em 21.5.2003]

[MAYBERRY03]

MAYBERRY (III), Marshall R.; MIIKKULAINEN, Risto (2003)- Incremental Nonmonotonic Parsing through Semantic Self-Organization [Em Linha] (to appear in **Proceedings of the 25th Annual Conference of the Cognitive Science Society (COGSCI-03),** Boston, Massachusetts) [Consult. em 4 Jul 2003] Disponível na WWW: <http://www.cs.utexas.edu/users/nn/downloads/papers/mayberry.cogsci03.pdf. >

[MAYBERRY94]

MAYBERRY (III), Marshall R.; MIIKKULAINEN, Risto (1994) - Lexical Disambiguation Based on Distributed Representations of Context Frequency [Em linha](in **Proceedings of the 16th Annual International Joint Conference on Artificial Intelligence (IJCAI-99, Stockholm, Sweden)**, 820-825. San Francisco, CA: Kaufmann, 1999.) [Consult. em 3 Nov 2003] Disponivel na WWW: <http://www.cs.utexas.edu/users/nn/downloads/papers/mayberry.disambiguation.pdf. >

[MENG02]

MENG [et al.] (2002) - Concept Hierarchy Based Text Database Categorization [Em linha]. (Também In: **International Journal on Knowledge and Information Systems** , Vol. 4, Vol. 2, pp.132-150, March 2002) [Consultado em: 1.10.2003] Disponível na WWW: <http://opal.cs.binghamton.edu/~meng/pub.d/kais01h.ps. >

[MENG99]

MENG, Wenxian; LIU, King-Lup; YU, Clement; WU, Wensheng; RISHE, Naphtali: (1999) - Estimating the Usefulness of Search Engines [Em Linha] (In **Proc. of the 15th International Conference on Data Engineering (ICDE'99)** , Sydney, Australia, March 1999, pp.146-153.) [Consultado em 1.Out 2003] Disponível na WWW: <http://opal.cs.binghamton.edu/~meng/metasearch.html. >

[MICROSOFT00]

MICROSOFT Corporation – Sample Chapter from MCSE Academic Learning Series -- Microsoft® Windows® 2000 Server [Em linha] (tb. Chapter 2: Installing and Configuring Microsoft Windows 2000 Server, in **ALS Microsoft® Windows® 2000 Server**, 2000, ISBN: 0-7356-0988-8). [Consultado em 19 Mar 2004]. Disponível na WWW: <http://www.microsoft.com/mspress/books/sampchap/4244.asp. >

[MIIKKULAINEN90]

MIIKKULAINEN, Risto (1990) - A PDP Architecture for Processing Sentences with Relative Clauses [Em linha] (In **Proceedings of the 13th International Conference on Computational Linguistics**, Helsinki. Disponível na WWW: <http://citeseer.nj.nec.com/miikkulainen90pdp.html>) [Consultado em 14 Mai. 2003] Disponível na WWW: <http://nn.cs.utexas.edu/downloads/papers/miikkulainen.relative-clauses.pdf. >

[MIIKKULAINEN96]

MIIKKULAINEN, Risto (1996) - Subsymbolic Case-Role Analysis of Sentences with Embedded Clauses [Em Linha] (in **Cognitive Science**, 20:47-73, 1996)[Consult. em 5 Nov. 2003] Disponível na WWW: <http://www.cs.utexas.edu/users/nn/downloads/papers/miikkulainen.subsymbolic-caseroles.pdf>

[MIIKKULAINEN97]

MIIKKULAINEN, Risto (1997) - Natural Language Processing with Subsymbolic Neural Networks[Em linha]. (In A. Browne (editor), **Neural Network Perspectives on Cognition and Adaptive Robotics**. Institute of Physics Publishing, 1997), [Consult. em 14 Mai. 2003] Disponível na WWW: <http://nn.cs.utexas.edu/downloads/papers/miikkulainen.perspectives.pdf. >

[MSDN02]

MSDN (2002) – Creating a Query in JScript (Platform SDK: Indexing Service) [Em linha], **Platform SDK Release : August 2002**, Microsoft, in MSDN Library > SDK Documents > Using Indexing Services with File Systems > Programming with Scripts > Using JScript with Indexing Service, [Consultado em 7 Nov. 2002] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/indexsrv/html/ixufilsc_7wj8.asp?frame=true. >

[MSDN03a]

MSDN (2003a) – What's New in Indexing Service 3.0? (Indexing Service SDK) [Em linha], **Platform SDK Release : February 2003**, Microsoft, in MSDN Library, Windows Development – Windows Base Services – Indexing Service- About Indexing Service, [Consultado em 13 Mar. 2004] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/indexsrv/html/ixintro_24og.asp. >

[MSDN03b]

MSDN (2003b) - Language Resources [Em linha], **Platform SDK Release : February 2003**, Microsoft, MSDN Home > MSDN Library > Windows Development > Windows Base Services > Indexing Service > About Indexing Service > Architecture of Indexing Service > Components of Indexing Service, [Consultado em 16 Mar. 2004] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/indexsrv/html/ixarch_6ulw.asp?frame=true. >

[MSDN03c]

MSDN (2003c) – About Language Resources [Em linha], **Platform SDK Release : February 2003**, Microsoft, MSDN Home > MSDN Library > Windows Development > Windows Base Services > Indexing Service > Extending Language Resources for Indexing Service, [Consultado em 16 Mar. 2004] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/indexsrv/html/wbrscenario_3u2c.asp?frame=true. >

[MSDN03d]

MSDN (2003c) – SELECT Statement [Em linha], **Platform SDK Release : February 2003**, Microsoft, MSDN Home > MSDN Library > Windows Development > Windows Base Services > Indexing Service > Indexing Service Reference > Query-Language Syntax > SQL Queries and SQL Extensions, [Consultado em 26 Mar. 2004] Disponível na WWW: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/indexsrv/html/ixrefqls_38fo.asp. >

[MSDN04]

MSDN (2004) – Using File Properties for File Content Searches [Em linha], **Accessing and Changing Relational Data (SQL Server 2000)**, MSDN Home > MSDN Library > Enterprise Development > Windows Server System > Microsoft SQL Server > Microsoft SQL Server 2000 > Full-text Search > Full-text Querying of File Data > Using Virtual Tables for File Content Queries, [Consultado em 23 Mar. 2004] Disponível na WWW: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_8_qd_15_5r8z.asp. >

[NAREDDY97a]

NAREDDY, Krishna (1997a) - **Anatomy of a Search Solution** [Em linha], Microsoft, 1997, in MSDN Library, Technical Articles – Web Development – Microsoft Internet Information Server, [Consultado em 20 Nov. 2002] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/dnindex/html/msdn_ss-intro.asp. >

[NAREDDY97b]

NAREDDY, Krishna (1997b) - **Introduction to Microsoft Index Server** [Em linha], Microsoft, 1997 in MSDN Library, Technical Articles – Web Development – Microsoft Internet Information Server, [Consultado em 20 Nov. 2002] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/dnindex/html/msdn_ss-intro.asp. >

[NAREDDY98]

NAREDDY, Krishna (1998) - **Indexing with Microsoft Index Server**, Microsoft, 1998, in MSDN Library, Technical Articles – Web Development – Microsoft Internet Information Server, [Consultado em 20 Nov. 2002] Disponível na WWW: <http://msdn.microsoft.com/library/en-us/dnindex/html/msdn_is-index.asp. >

[PEREIRA03]

PEREIRA, Alexandre; POUPA, Carlos (2003) – **Como escrever uma tese, monografia ou livro científico usando o Word**. 1ª Edição, Lisboa: Edições Sílabo. ISBN: 972-618-290-5

[POLLACK90]

POLLACK, Jordan B. (1990) - Recursive distributed representations [Em linha].(in: **Artificial Intelligence**, 46, 77-105). [Consult. em 8 Nov. 2003] Disponível na WWW: <http://citeseer.nj.nec.com/pollack90recursive.html. >

[RAJMAN97]

RAJMAN, Martin; BESANÇON, Romaric (1997) – Text Mining: Natural Language techniques and Text Mining applications [Em linha]. (Também In: **Proceedings of the seventh IFIP 2.6 Working Conference on Database Semantics (DS-7)**, Chapman & Hall, IFIP Proceedings serie, Leysin, Switzerland, 1997, Out. 7-10.) [Consult. em 15 Nov. 2002] Disponível na WWW: <http://citeseer.nj.nec.com/rajman97text.html. >

[RAJMAN98]

RAJMAN, Martin; BESANÇON, Romaric (1998) – Text Mining -Knowledge extraction from unstructured textual data [Em linha]. In: proc. **6th Conference of International Federation of Classification Societies (IFCS-98)**, Rome, 1998. [Consult. em 31 Out. 2002] Disponível na WWW: <http://citeseer.nj.nec.com/besanon98text.html. >

[RAJMAN99]

RAJMAN, Martin; BESANÇON, Romaric (1999) – **Stochastic Distributional Models for Textual Information Retrieval** [Em linha], 1999. [Consult. em 14 Nov. 2002] Disponível na WWW: <http://liawww.epfl.ch/~lnmain/publications/RajmanBesancon99.ps.gz. >

[REIS01]

REIS, Elisabeth (2001) –**Estatística Multivariada Aplicada**, 2ª edição, Lisboa: Edições Sílabo. ISBN: 972-618-247-6

[RUSSINOVICH02]

RUSSINOVICH, Mark (2002) – Inside Win2K NTFS, Part 1 [Em linha]. (Também In **Windows 2000 Magazine (2000)**, Penton Media Inc.) [Consultado em: 19 Mar. 2004] Disponível na WWW: <http://msd.microsoft.com/library/en-us/dnw2kmag00/html/ntfspart1,asp?frame=true. >

[SALTON88]

SALTON, Gerard; BUCKLEY, Christopher (1988) - Term-weighting approaches in automatic text retrieval, **Information Processing & Management**, Pergamon Press, Great Britain, Vol. 24, No 5, pp. 513-523, 1988

[SDUA]

Serviços de Documentação da UA – **O Sistema de Classificação da Biblioteca, a CDU** [Em linha, consultado em 4 Fev. 2004]. Disponível na WWW: <http://www.doc.ua.pt/opac/cdu.html. >

[SINGHAL96a]

SINGHAL, Amit; SALTON, Gerard; BUCKLEY, Chris (1996a) - Length Normalization in Degraded Text Collections. [Em Linha] in proc. **Fifth Annual Symposium on Document Analysis and Information Retrieval**, 149-162, 1996 [Consultado em 16 Out. 2003] Disponível na WWW: <http://singhal.info/ocr-norm.ps. >

[SINGHAL96b]

SINGHAL, Amit; SALTON, Gerard; BUCKLEY, Chris; MITRA, Mandar (1996b) - Pivoted Document Length Normalization. [Em Linha]. **ACM SIGIR'96**, 21-29, 1996.[Consultado em 16 Out. 2003] Disponível na WWW: <http://singhal.info/pivoted-dln.ps. >

[SPENCER00]

SPENCER, Ken (2000)- Indexing Services at Your FingerTips, in **Windows 2000 Magazine**, July, 2000, Penton Media Inc. (antes por Duke Communications International Inc.)

[TECHNET01]

MICROSOFT TECHNET (2001) – Microsoft Full-Text Search Technologies [Em Linha], **Microsoft Technet**, TechNet Home > Products & Technologies > Servers > SharePoint Products & Technologies > SharePoint Portal Server > Evaluate, [Consult. Em 13 Mar. 2004] Disponível na WWW: <http://www.microsoft.com/technet/prodtechnol/sppt/sharepoint/evaluate/featfunc/mssearch.mspx >

[TECHNET02]

MICROSOFT TECHNET (2002) – Textual Searches on File Using MS SQL Server 7.0 [Em Linha], **Microsoft Technet**, TechNet Home > Products & Technologies > SQL Server > Maintain > Feature Usability, [Consult. Em 5 Nov. 2002] Disponível na WWW: <http://www.microsoft.com/technet/prodtechnol/sql/maintain/featusability/textsrch.asp?fram... >

[TUMMARATTANANONT02]

TUMMARATTANANONT, Pornchai (2002?) – **Improvement of Thai OCR Error Correction using Overlapping Constraints to Correction Suggestion** [Em linha, consultado em 10 Set. 2003] Disponível na WWW: <http://arnthai.links.nectec.or.th/papers/OCR_correction_suggestion.pdf>

[WANG02]

WANG, Wenxian; MENG, Weiyi; YU, Clement (2002) - Concept Hierarchy Based Text Database Categorization in a Metasearch Engine Environment [Em linha]. (Também In **Proc. First International Conference on Web Information Systems Engineering (WISE'2000)**, Hong Kong, Junho 2000, pp. 283-290) [Consultado em: 1.10.2003] Disponível na WWW: <http://opal.cs.binghamton.edu/~meng/metasearch.html>

[WEIDE01]

Van der WEIDE (2001) – **Information discovery** [Em linha], [Consultado em 21 Out. 2003] Disponível na WWW: < http://www.cs.kun.nl/is/edu/ir1/ir1.pdf. >, pp. i-ii, 1-145

[WILLET88]

WILLET, Peter (1988) – Recent Trends in hierarchic document clustering: a critical review, **Information Processing & Management**, Pergamon Press, Great Britain, Vol. 24, No 5, pp. 577-597

Notação e Abreviaturas utilizadas

A bibliografia está organizadas por autor-data, ex. [ECO98]. O formato da referência bibliográfica segue a norma portuguesa NP 405 conforme descrito e exemplificado em [PEREIRA03].

Nas citações e referências, a designação da página é estabelecida da seguinte forma: [ECO98: 56], significando a página 56 da obra referenciada, neste caso: ECO, Umberto (1998) – **Como se faz uma tese em ciências Humanas**. 7ª Edição, Lisboa: Editorial Presença. ISBN: 972-23-1351-7.

Na designação de outros elementos, estes são designados com a abreviatura correspondente à sua natureza, precedendo a sua designação, ex. [PEREIRA03: cap. 6], significando o capítulo 6 da referência [PEREIRA03].

As abreviaturas utilizadas, em regra, estão descritas em [ECO98], Quadro 21, páginas 212 e 213.

Exceptua-se a abreviatura Orign., significando “originariamente ou originalmente”, extraída da lista de abreviaturas do Dicionário Houaiss de Língua Portuguesa [HOUAISS02].

Lista das abreviaturas utilizadas:

cf.	Confrontar, ver também
ex.	Por exemplo
i.e.	Isto é, quer dizer
Ibid.	Ibidem, em regra na mesma obra e mesma página
n.	nota
Op. Cit.	Obra citada, seguido de número de página
p.	página
passim	Aqui e ali
Sic	Assim (escrito assim, usado como medida de prudência)
Tr. tr.	tradução
V. v.	ver

Nas citações, a omissão de partes do trecho é representada por (...).

Anexo I – Exemplos de utilização do IS com ASP

Exemplos simples de consultas utilizando ASP [FEDOROV98]. Qualquer um dos casos envia ao utilizador uma listagem do ficheiros indexados, ou seja revela um conteúdo do Index Service.

É de salientar que a ampliação do número de colunas a extrair do IS, em qualquer dos casos, se traduziu numa acentuada perda de desempenho. Como inicialmente o conjunto de colunas a extrair incluía dados como *size*, *write* e *rank*, a resposta era muito demorada. Daí o "Server.ScriptTimeout = 960" parâmetro no primeiro exemplo apresentado, que se manteve apenas para recordar esta dificuldade.

O segundo exemplo, comparativamente, é mais simples do que o primeiro. Oferece também uma visão mais próxima da noção de consulta a uma base de dados.

Sem questionar a utilidade de tais listagens, deve dizer-se que o objectivo é constituir a lista do conjunto de documentos indexados, a partir dos quais é possível extrair palavras e respectivas estatísticas. Estes dados servirão o propósito de constituir vectores de classificação de documentos, ou padrões de pesquisa.

Listagem de ficheiros utilizando os objectos IXSSO

Este código ASP foi criado a partir do tutorial "**Creating a Query in JScript**" [MSDN02]. Neste caso recorre-se à utilização dos objectos ixsso.query e ixsso.utility. O directório estabelecido como patamar superior de pesquisa ("scope") está neste caso estabelecido com o método objU.AddScopeToQuery.

```
<% @language = "JScript" %>
<% Server.ScriptTimeout = 960 %>


<DOCTYPE HTML PUBLIC - "//W3C/DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Query in JScript</title>
</head>
<body>

<%
// Declare Variables
var strGroupBy;         // Name of GroupBy column.
var intJ;         // Index variables.
var objQ;               // Query object.
var strRecord;          // Output record of query results.
var objRS;        // Recordset object.
var intRS_Count;  // Number of current record of Recordset.
var objU;               // Utility object.

// Create a Query Object
```

```
objQ = new ActiveXObject("IXSSO.Query");

//Set Properties of the Query Object
objQ.Columns = "filename";
objQ.Query   = "#filename=*.htm";
objQ.GroupBy = "";
objQ.Catalog = "System";
objQ.OptimizeFor      = "recall";
objQ.AllowEnumeration = true;
objQ.MaxRecords       = 20000;

//Create a Utility Object
objU = new ActiveXObject("IXSSO.Util");

//Add the Physical Path and all Subdirectories
objU.AddScopeToQuery(objQ, 'C:\\My Documents\\Estudo\\Textos\\C1821\\', "deep");

//Output Query Properties
Response.Write("<BR/> Columns = " + objQ.Columns);
Response.Write("<BR/> Query = " + objQ.Query);
Response.Write("<BR/> GroupBy = " + objQ.GroupBy);
Response.Write("<BR/> Catalog = " + objQ.Catalog);
Response.Write("<BR/> CiScope = " + objQ.CiScope);
Response.Write("<BR/> CiFlags = " + objQ.CiFlags);
Response.Write("<BR/> OptimizeFor = " + objQ.OptimizeFor);
Response.Write("<BR/> AllowEnumeration = " + objQ.AllowEnumeration);
Response.Write("<BR/> MaxRecords = " + objQ.MaxRecords);

//Create a Recordset Object
objRS = objQ.CreateRecordset("nonsequential");
//Read through the Recordset Object, Extracting and Outputting Values
intRS_Count = 0;
while (!objRS.EOF) {
    intRS_Count = intRS_Count + 1;
    strRecord = ( intRS_Count + ".      ").slice(0,4);
   strRecord =strRecord + "  " + objRS("filename")

    Response.Write("<BR/>"+strRecord);
    objRS.MoveNext;
};
//Close the Recordset Object
objRS.Close;
objRS = null;


%>
</body>
</html>
```

Listagem de ficheiros utilizando ADO

Um resultado semelhante é conseguido com o script seguinte. O directório patamar superior de pesquisa é estabelecido com a declaração FROM System..SCOPE(...).

```
<% @LANGUAGE=VBScript %>
<html>
<head><title>Teste SQL</title></head>
<body>
<%
SQL="SELECT filename, path FROM System..SCOPE('DEEP TRAVERSAL OF ""C:\My
Documents\Estudo\Textos\C1821\""')"
```

```
SQL=SQL & " WHERE "
SQL= SQL & "CONTAINS(FileName, '*.htm')"
intC=0
strC=""
Set objConnection=Server.CreateObject("ADODB.Connection")
objConnection.ConnectionString = "provider=msidxs;"
objConnection.Open
%>
<%
Set objRS=objConnection.Execute(SQL)
%>
<% =SQL %>
<% If Not objRS.EOF Then%>
<%
Do While Not objRS.EOF
   intC=intC+1
   strC=objRS("Path")
%>
<br><% =intC %> - <% =strC %>
<% objRS.MoveNext
%>
<% Loop %>
<% End If%>
<% Set objRS = Nothing %>
<% objConnection.Close %>
<% Set objConnection = Nothing %>

</body>
</html>
```

Anexo II – Falhas de Segurança com o Index Server

Nas páginas seguintes ilustram-se três exemplos que testemunham falhas de segurança ao nível do Index Server, conforme referido na nota nº 60 da página 50.

Pesquisando no site http://www.microsoft.com, é possível encontrar uma variedade de artigos que referem correcções a problemas de segurança com o IS. Basta pesquisar aí, por exemplo, utilizando as palavras “index server security fix”. Os artigos, localizados na “Microsoft Knowledge Base” com números 232449 e 164059, são transcritos parcialmente abaixo.

Problemas com a mesma caracterização são apresentados no artigo, de Andrew Smith (Maio 1997) , “MS IndexServer exposes passwords”. O autor denuncia a forma como seria possível ler código de ficheiros ASP através de pesquisas usando o Index Server.

Artigo 232449

Sample ASP Code May be Used to View Unsecured Server Files

Aplica-se a

This article was previously published under Q232449

SYMPTOMS

When you install the following Active Server Page (ASP) sample files on a computer running Internet Information Server (IIS) 4.0, a Web visitor may be able to use these files to gain access to and read any known file on the same logical disk as the installed ASP code, which is not protected by setting the system Access Control Lists (ACLs) for these files:

- *IIS_DIRECTORY*\Iissamples\Exair\Howitworks\Code.asp
- *IIS_DIRECTORY*\Iissamples\Exair\Howitworks\Codebrws.asp
- *IIS_DIRECTORY*\Iissamples\Sdk\Asp\Docs\Codebrws.asp
- *Program_Files**Common_Files*\System\Msadc\Samples\Selector\Showcode.asp

Please note, however, that the Web visitor cannot change, delete, or add any files

Artigo 164059

IIS Execution File Text Can Be Viewed in Client

Aplica-se a

This article was previously published under Q164059

SYMPTOMS

NOTE: On Sunday, February 23, 1997, Microsoft was alerted to a posting regarding an Internet Information Server (IIS) security exposure. This bug permits the publication of IIS executable files via a complicated string of commands sent from a web browser to an IIS server.

Because of this security exposure, Internet users can view Active Server Pages (ASP), Internet Server API applications (ISAPI), Internet Database Connector (IDC) applications, or Common Gateway Interface (CGI) applications within an IIS publication installation.
(…)

May 19, 1997 | Editors | Discoveries | InstantDoc #9239

MS Index Server exposes passwords

Microsoft Index Server Exposes IDs and Passwords

Reported May 15,1997 by Andrew Smith

Systems Affected

Windows NT with IIS and Index Server (e.g. any NT system using IIS with webhits.exe in the default location or locatable/executable path)

The Problem

MS Index Server (formerly code named Tripoli) is Microsoft's search engine for Internet Information Server. It recently shipped with Service Pack 2 for Windows NT and is installed on most Microsoft NT Internet information web servers. Index Server is a very useful search engine for the Internet Information Server. One component contained in Index Server is called the Hit Counter. Hit counter enables users to view their searched documents with the words of their queries highlighted..

The Hit Counter (webhits.exe) allows the web server to read files that should not normally be able to be read. This is similar to a bug found recently that allows users to read Active Server Script files by placing a period at the end of the URL. In many cases an Active Server script contains a username and password to a network resource, usually a SQL server. This password and username can be used to gain access to the SQL system and possibly to the web server itself.

If the system administrator has left the default sample files on the Internet Information server, a hacker would have the opportunity of narrowing down their search for a username and password. A simple query of a popular search engine shows about four hundred websites that have barely modified versions of the sample files still installed and available. This file is called queryhit.htm. Many webmasters have neglected to modify the search fields to only search certain directories and avoid the script directories.

Magazine
- Exchange & Outlook Admin
- Mobile & Wireless Solutions
- Security Admin
- Storage Admin
- SQL Server Magazine
- Windows Scripting Solutions
- Windows Web Solutions

NETWORK RESOURCES

- WinInfo News
- SuperSite for Windows
- Forums
- Windows NT/2000 FAQ
- IIS FAQ
- IT Buyer's Network
- Windows IT Library
- Subscribe/Renew
- Events
- Mobile Edition
- User Groups
- Web Seminars
- White Paper Central

Once one of these sites is located a search performed can easily narrow down the files a hacker would need to find a username and password. Using the sample search page it is easy to specify only files that have the word password in them and are script files (.asp or .idc files. cold fusion scripts, even .pl files are good).

The URL the hacker would try is http://servername/samples/search/query.htm then the hacker would search with something like "filename".asp"

When the results are returned not only can one link to the files but also can look at the "hits" by clicking the view hits link that uses the webhits program. This program bypasses the security set by IIS on script files and allows the source to be displayed.

Even if the original samples are not installed or have been removed a hole is still available to read the script source. If the server has Service Pack 2 fully installed (including Index Server) they will also have webhits.exe located in the path

http://servername/scripts/samples/search/webhits.exe

This URL can preface another URL on that server and display the contents of the script.

Stopping the Attack

To protect your server from this problem remove the webhits.exe file from the server, or at least from it's default directory. I also recommend that you customize your server search pages and scripts (.idq files) to make sure they only search what you want - such as plain .HTM or .HTML files. Index Server is a wonderful product but be sure you have configured it properly.

Microsoft's Response:

Andrew Smith has made Microsoft aware of the problem, but they have yet to release a formal fix as of May 19, 1997.

If you want to learn more about new NT security concerns, subscribe to NTSC.

Scott
Original page located here
Post on The NT Shop May 19, 1997

Topics

- Security

Related Articles

- Find Related Articles
- Related Products

Article Information

- Email this Article
- Printer-Friendly

Reader Comments

- Post a Comment

Penton

Anexo III - Ilustração de erros sintácticos e semânticos

Este anexo contém o código integral HTML e o fac-símile da página 396 relativo à acta do dia 10 de Junho de 1822. Trata-se do exemplo referido no capítulo 2.

Início do código HTML ..\C1821\A1822\M06\D10A1822M06D10-0396.htm

```
<HTML>
<HEAD>
<meta name="LEGIS" content="01">
<meta name="SERIE" content="01">
<meta name="SESSION" content="02">
<meta name="NUMBER" content="030">
<meta name="LIMITS" content="0392-0412">
<meta name="PAGE" content="0396">
<meta name="DATE" content="1822-06-10">
<meta name="ESTADO" content="OK">
</HEAD>
<BODY>
<p>[ 396 ]<br>
 <br>
encarregasse a arrecadação da fazenda. He conveniente que não consideremos a
demarcação das com areas do reino com a desigualdade em terreno, e em povoação
como ellas presentemente estão. Eu não me opporei a que o contador da fazenda na
comarca, presida a uma junta convocada na cabeça da comarca só para o fim de
receber os recursos das partes aggravadas da derrama da contribuição directa;
isto he de absoluta necessidade, e estas juntas até já são da creação do subsidio
militar da decima. Ninguem duvida que a arrecadação da fazenda por um methodo
differente do actual vai ser mais dispendiosa, porque até ao presente os
differentes magistrados estavão encarregados dessa arrecadação, e a fim de serem
despachados, e adiantados na sua carreira, erão obrigados a apresentar as suas
contas correntes, muitos padecérão por causa da sua negligencia, desleixo, e
prevaricação; e até um, o ministro do bairro de Andaluz, teve a infelicidade de
padecer a pena ultima no tempo do ministerio do Marquez de Pombal. Até ao
presente os provedores, na qualidade de contadores, são encarregados da cobrança
de differentes imposições, e rendimentos de commendas, dizimarias  pertencentes
ao thesouro, ou como prestimonios, ou fazendo parte de outros ramos de fazenda;
como provedores elles arrecadão as terras reaes. Os corregidores na qualidade de
superintendentes geraes da decima das comarcas,<br>
são as autoridades encarregadas dessa arrecadação, e de outras imposições, além
de serem juntamente com os juizes de fóra, os que presidem aos lançamentos<br>
nas differentes superintendencias particulares. Alguns juizes de fóra estão
encarregados de administrações de rendimentos pertencentes as casas das Senhoras
Rainhas, do Bragança, e do Infantado. O trabalho dessas arrecadações dividia-se
pelas autoridades civis, a fim de se poupar a fazenda ordenados de exactores da
mesma. Estabelecendo-se em cada comarca do reino uma contadoria  da fazenda, na
forma que indica o Sr. Ferreira Borges, vai-se estabelecer um centro de
arrecadação de rendas da comarca, e todas aquellas vantagens, que se dirivão da
presteza, simplicidade, e clareza de contas com o erario. Não ha duvida que o
estabelecimento dos contadores trará alguma despeza mais, e vamos nessa parte
verificar o dito vulgar, que o bonito sáe caro; porém vai-se introduzir na
arrecadação um systema, ordem, e actividade, até ao presente tempo desconhecidos.
Proponho, a fim de evitarmos o não concluir tão cedo cousa alguma de uma
discussão tão vaga, que o projecto torne à redacção, e que a illustre Commissão
de o seu parecer novamente, informada já pelo que tem ouvido<br>
a muitos illustres Membros do Congresso, que já falarão; e que a mesma Commissão
sejão remettidas, tanto a indicação do Sr. Ferreira Borges, como a<br>
minha, e que apezar do reconhecermos a importancia deste objecto, tambem
reconhecemos que tarde poderemos tirar uma conclusão de uma discussão tão vaga
como a presente.<br>
O Sr. Ferreira Borges: - Quasi que era desnecessario falar sobre este objecto, ao
menos pela minha parte, depois do terem falado os ultimo. Preopinantes; entre
```

tanto como o outro dia mal pude pronunciar a minha opinião, direi agora a razão me que me fundo. Eu considero na administração uma delegação do poder executivo, por isso mesmo que quando do tratamos do artigo 30 em que vinha designado o que era poder administrativo se uniu ao poder executivo. Por tanto temos que administração he uma parte do poder executivo; ora isto está sanccionado. Esta administração tem duas partes, uma que respeita ao que he fazenda, e outra que comprehende tudo o mais que não he fazenda: ou não podia confundir uma cousa com a outra; e em um e outro caso, não posso admittir corpos collectivos. Feita esta divisão apresentei, na parte que pertence a fazenda, aquelle projecto. Nos temos actualmente 44 comarcas; porém no futuro devemos ter menos, e naturalmente não passarão de 22; e em cada um destes circulos administrativos haverá um homem a quem eu chamei contador (ou chame-se com outro qualquer nome), o qual poderá ter muito bem a seu alcance estado das rendas publicas, a sua arrecadação, e fiscalização; mediante um por cento que se lhe dê. Parecia-me pois que havendo este homem que administrasse era muito mais facil a escrituração no thesouro, e que desta viria o resultado geral que he a melhor administração. Os corpos collectivos nunca podem promover a administração, como um homem só. Bonnin que fez um tratado de codigo administração, mostra a utilidade das administrações encarregadas a um homem só, que seja responsavel , porque he sempre mais facil responder um homem só do que um corpo collectivo. Eis aqui o meu modo de pensar a este respeito.
Interrompeu o Sr. Presidente o debate, a fim de participar as Cortes que D. Rodrigo Antonio de Mello que acabava de ser governador da ilha da Madeira, vinha felicitar o Congresso, atributar-lhe o mais fiel respeito, e obediencia: a felicitação foi ouvida com agrado, e o Sr. Secretario Freire foi significar isto mesmo áquelle official, na forma do costume.
Continuando a discussão interrompida, apresentou o Sr. Soares Franco uma indicação que remetteu para a meza, e depois de fazer algumas reflexões sobre o objecto em discussão, concluiu dizendo que se conformava com a opinião do Sr. Sarmento.
O Sr. Peixoto: - Sobre a reprovação das juntas administrativas, esta o Congresso quasi concorde: resta sómente concordar na pessoa, ou corpo, que ha de pôr-se a frente da repartição, e fiscalização das rendas publicas: sobre este ponto digo; que convêm , que nas provincial se faça uma repartição systematica de comarcas, e que em cada uma dellas haja um funccionario com o titulo de contador, o qual tenha sempre aberta uma conta corrento com o thesouro, e outra com cada uma das recebedorias suas subalternas, a fim de promover as arrecadações parciaes, e facilitar a expedição das operações do ministerio da fazenda publica. Nada mais se precisa, visto que uma tal autoridade no actual estado das nossas rendas publicas, em nada mais pode entender. No thesouro hão do constar em cada anno todas as verbas de rendimentos publicos, ou sejam constantes, como os cabeções das sizas, ou variaveis, por arrendamentos, e ad-</p>
</BODY>
</HTML>

Fim do código HTML ..\C1821\A1822\M06\D10A1822M06D10-0396.htm

Fac-símile da página 396 do dia 10 de Junho de 1822

encarregasse a arrecadação da fazenda. He conveniente que não consideremos a demarcação das comarcas do reino com a desigualdade em terreno, e em povoação como ellas presentemente estão. Eu não me oporei a que o contador da fazenda na comarca, presida a uma junta convocada na cabeça da comarca só para o fim de receber os recursos das partes aggravadas da derrama da contribuição directa; isto he de absoluta necessidade, e estas juntas até já são da creação do subsidio militar da decima. Ninguem duvida que a arrecadação da fazenda por um methodo differente do actual vai ser mais dispendiosa, porque até ao presente os differentes magistrados estavão encarregados dessa arrecadação, e a fim de serem despachados, e adiantados na sua carreira, erão obrigados a apresentar as suas contas correntes, muitos padecêrão por causa da sua negligencia, desleixo, e prevaricação; e até um, o ministro do bairro de Andaluz, teve a infelicidade de padecer a pena ultima no tempo do ministerio do Marquez de Pombal. Até ao presente os provedores, na qualidade de contadores, são encarregados da cobrança de differentes imposições, e rendimentos de commendas, dizimarias pertencentes ao thesouro, ou como prestimonios, ou fazendo parte de outros ramos de fazenda; como provedores elles arrecadão as terças reaes. Os corregedores na qualidade de superintendentes geraes da decima das comarcas, são as autoridades encarregadas dessa arrecadação, e de outras imposições, além de serem juntamente com os juizes de fóra, os que presidem aos lançamentos nas differentes superintendencias particulares. Alguns juizes de fóra estão encarregados de administrações de rendimentos pertencentes ás casas das Senhoras Rainhas, do Bragança, e do Infantado. O trabalho dessas arrecadações dividia-se pelas autoridades civis, a fim de se poupar á fazenda ordenados de exactores da mesma. Estabelecendo-se em cada comarca do reino uma contadoria da fazenda, na fórma que indica o Sr. *Ferreira Borges*, vai-se estabelecer um centro de arrecadação de rendas da comarca, e todas aquellas vantagens, que se dirivão da presteza, simplicidade, e clareza de contas com o erario. Não ha duvida que o estabelecimento dos contadores trará alguma despeza mais, e vamos nessa parte verificar o dito vulgar, que o bonito sáe caro; porém vai-se introduzir na arrecadação um systema, ordem, e actividade, até ao presente tempo desconhecidos. Proponho, a fim de evitarmos o não concluir tão cedo cousa alguma de uma discussão tão vaga, que o projecto torne á redacção, e que a illustre Commissão dê o seu parecer novamente, informada já pelo que tem ouvido a muitos illustres Membros do Congresso, que já falárão; e que á mesma Commissão sejão remettidas, tanto a indicação do Sr. *Ferreira Borges*, como a minha, e que apezar de reconhecermos a importancia deste objecto, tambem reconhecemos que tarde poderemos tirar uma conclusão de uma discussão tão vaga como a presente.

O Sr. *Ferreira Borges*: — Quasi que era desnecessario falar sobre este objecto, ao menos pela minha parte, depois de terem falado os ultimos Preopinantes; entre tanto como o outro dia mal pude pronunciar a minha opinião, direi agora a razão em que me fundo. Eu considero na administração uma delegação do poder executivo, por isso mesmo que quando do tratamos do artigo 30 em que vinha designado o que era poder administrativo se uniu ao poder executivo. Por tanto temos que administração he uma parte do poder executivo; ora isto está sanccionado. Esta administração tem duas partes, uma que respeita ao que he fazenda, e outra que comprehende tudo o mais que não he fazenda: eu não podia confundir uma cousa com a outra; e em um e outro caso, não posso admittir corpos collectivos. Feita esta divisão apresentei, na parte que pertence á fazenda, aquelle projecto. Nós temos actualmente 44 comarcas; porém no futuro devemos ter menos, e naturalmente não passarão de 22; e em cada um destes circulos administrativos haverá um homem a quem eu chamei contador (ou chame-se com outro qualquer nome), o qual poderá ter muito bem a seu alcance o estado das rendas publicas, a sua arrecadação, e fiscalização; mediante um por cento que se lhe dê. Parecia-me pois que havendo este homem que administrasse era muito mais facil a escrituração no thesouro, e que desta viria o resultado geral que he a melhor administração. Os corpos collectivos nunca podem promover a administração, como um homem só. *Bounin* que fez um tratado de codigo administrativo, mostra a utilidade das administrações encarregadas a um homem só, que seja responsavel, porque he sempre mais facil responder um homem só do que um corpo collectivo. Eisaqui o meu modo de pensar a este respeito.

Interrompeu o Sr. *Presidente* o debate, a fim de participar ás Cortes, que D. *Rodrigo Antonio de Mello*, que acabava de ser governador da ilha da Madeira, vinha felicitar o Congresso, e tributar-lhe o mais fiel respeito, e obediencia: a felicitação foi ouvida com agrado, e o Sr. Secretario *Freire* foi significar isto mesmo áquelle official, na forma do costume.

Continuando a discussão interrompida, apresentou o Sr. *Soares Franco* uma indicação que remetteu para a meza, e depois de fazer algumas reflexões sobre o objecto em discussão, concluiu dizendo que se conformava com a opinião do Sr. *Sarmento*.

O Sr. *Peixoto*: — Sobre a reprovação das juntas administrativas, está o Congresso quasi concorde: resta sómente concordar na pessoa, ou corpo, que ha de pôr-se á frente da repartição, e fiscalização das rendas publicas: sobre este ponto digo; que convêm, que nas provincias se faça uma repartição systematica de comarcas, e que em cada uma dellas haja um funccionario com o titulo de contador, o qual tenha sempre aberta uma conta corrente com o thesouro, e outra com cada uma das recebedorias suas subalternas, a fim de promover as arrecadações parciaes, e facilitar a expedição das operações do ministerio da fazenda publica. Nada mais se precisa, visto que uma tal autoridade no actual estado das nossas rendas publicas, em nada mais póde entender. No thesouro hão de constar em cada anno todas as verbas de rendimentos publicos, ou sejão constantes, como os cabeções das sizas, ou variaveis por arrendamentos, e ad-

Anexo IV – Acesso administrativo do Indexing Service

No Windows 2000/XP pode aceder-se a um formulário de consulta do Index Server a partir de uma consola de gestão do computador. Como condição prévia é necessário que o perfil do utilizador detenha privilégios adequados de acesso à administração do sistema, vulgo Administrator.

Essa possibilidade está acessível com a seguinte sequência, Start > Settings > Control Panel > Administrative Tools > Computer Management. Em alternativa, e talvez mais directamente, fazendo duplo clique sobre o ícone My Computer e continuando a sequência a partir do ícone Control Panel.

Chegando aqui visualiza-se uma janela aproximadamente semelhante à exposta na figura seguinte. Expandindo o ramo Services and Applications, fica visível o Indexing Service.

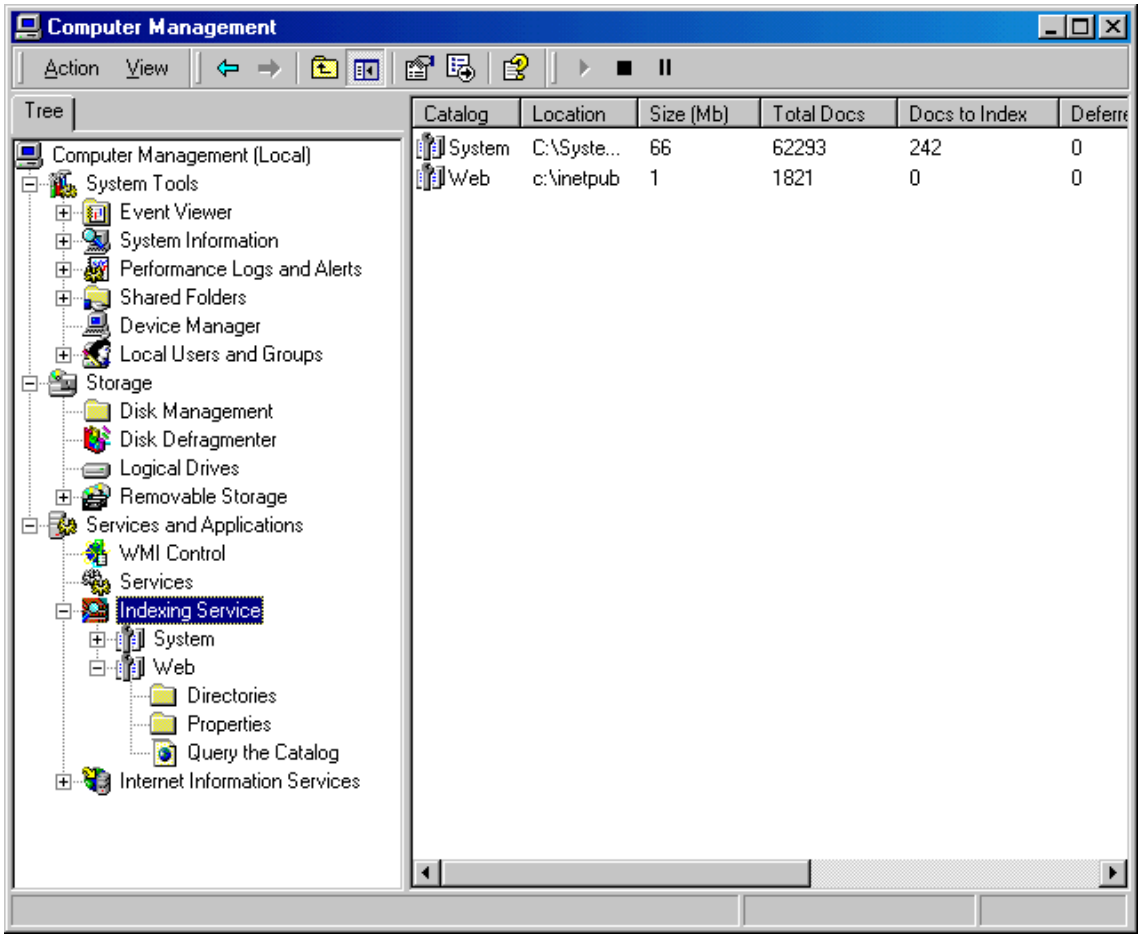

Figura 25 - Janela “Computer Management” do MS W2K

No caso de pretender criar um acesso específico para o Indexing Service, pode-se criar uma consola de gestão, ou Microsoft Management Console ou MMC, nome com que a Microsoft designa um sistema de janelas destinado a receber instrumentos de gestão.

Para criar uma MMC executa-se o comando mmc a partir do Command Prompt ou preenchendo o campo apresentado após a sequência Start > Run. Nestas circunstâncias é apresentada uma janela intitulada, por exemplo, Console1. Pode adicionar-se o Indexing Service, efectuando a sequência Console > Add/Remove Snap In > Add e seleccionando o Indexing Service da lista apresentada. Finaliza-se confirmando com Add, fecha-se a janela da lista dos serviços (Close) e a janela Add/Remove Snap in com OK.

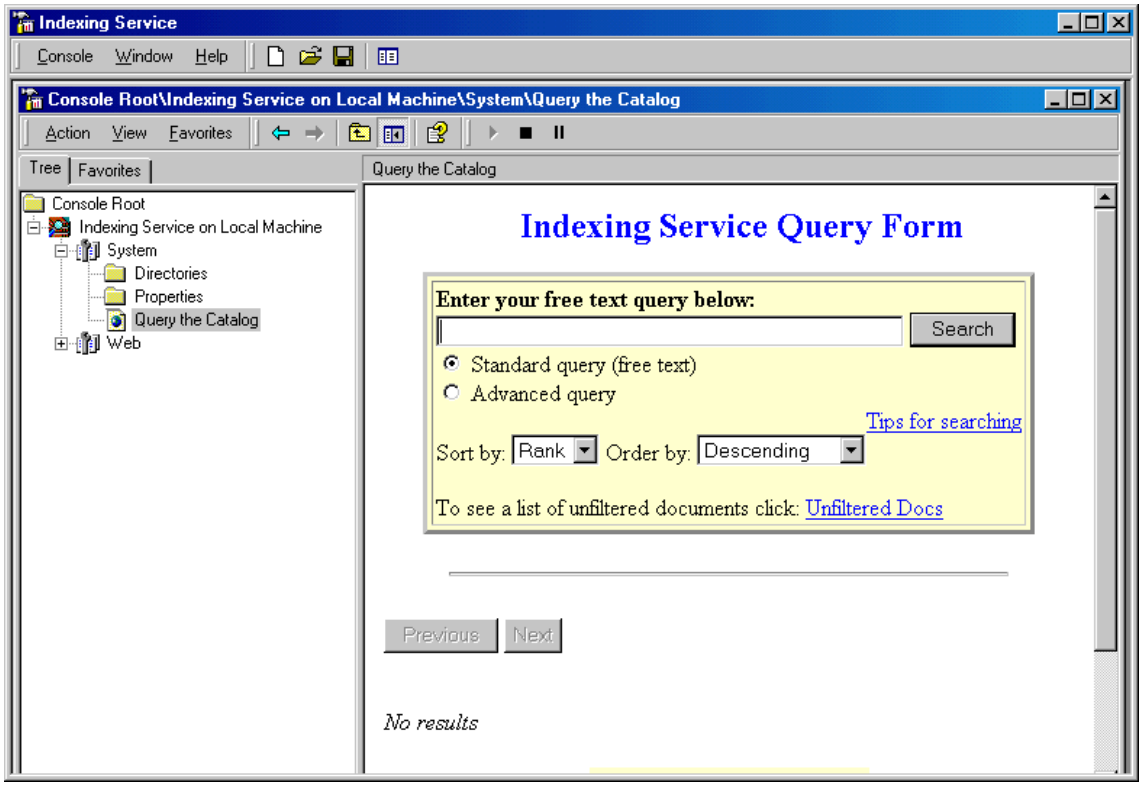

Figura 26 - MMC Indexing Service MS W2K com Query Form visível

Verificar-se-á que a árvore apresentada no painel esquerdo será semelhante ao ilustrado na Figura 26.

Expandindo, por exemplo, o ramo System (clicando o sinal + apresentado à esquerda), e escolhendo a folha Query the Catalog, fica visível o Indexing Service Query Form.

Ao terminar, é pedido para salvar a janela de administração criada, em regra numa pasta que respeita às tarefas de administração acessíveis a partir do menu Start. Aqui a escolha pertence ao utilizador. Pode escolher outro destino e atribuir um nome diferente.

Anexo V – Exemplo em MS Access

Em MS Access pode ser ilustrada a obtenção de listas com a frequência de palavras por documento utilizando código VBA semelhante ao seguinte. Isto mostra a possibilidade de constituir de tabelas de associação de documentos a palavras e as respectivas frequências.

```
Option Compare Database
Dim objCon As Connection, conP As Connection
Dim objRS As Recordset, rsP As Recordset, rsD As Recordset, rsPD As Recordset
Dim intC As Integer, strC As String

Sub Inicializar_IS()
Set objCon = New Connection
Set objCon = CreateObject("ADODB.Connection")
objCon.ConnectionString = "provider=msidxs"
objCon.Open

End Sub

Sub preenche_palavra(palavra As String)
'um erro frequente consiste na inexistência de uma palavra
'por pertencer à lista de palavras ruidosas
'não faz sentido adicioná-la
On Error GoTo termina
'Uma alteração possível consistirá em registar estes erros numa lista
'específica
intC = 0
intH = 0
varSQL = "SELECT Path, HitCount FROM System..SCOPE('DEEP TRAVERSAL OF ""C:\My
Documents\Estudo\Textos\C1821\""')"
varSQL = varSQL & " WHERE "
varSQL = varSQL & "CONTAINS(Contents, '" & palavra & "')"
varSQL = varSQL & " ORDER BY  Path "

Inicializar_IS

Set objRS = objCon.Execute(varSQL)


If Not objRS.EOF Then
   Do While Not objRS.EOF
    intH = objRS("HitCount")
    intC = intC + 1
    strC = objRS("Path")
Debug.Print intC & " - " & strC & " : " & intH
objRS.MoveNext
   Loop
End If
termina:
DoCmd.SetWarnings True
Set objRS = Nothing
objCon.Close
Set objCon = Nothing

End Sub
```

Anexo VI – Preechimento com programa em Java

A classe Preenche tem como principal objectivo controlar o preenchimento das tabelas de palavras e respectivas frequências de ocorrência em documentos localizados sob um directório indicado como argumento inicial.

A primeira tarefa que executa consiste em constituir uma lista de caminhos completos (“full path”) de ficheiros localizados no directório indicado como argumento e respectivos subdirectórios. Essa tarefa é desempenhada por um vector criado pelo objecto DirSub. A enumeração dos elementos desse vector envolverá o restante controlo de execução das actividades, com excepção da seguinte.

De seguida cria um objecto LigaDB, de ligação à base de dados, que virá a ser necessária para o preenchimento das tabelas. Esse objecto surgirá como argumento no tratamento de Documentos, Palavras, ocorrências de palavras por documento. É um pré-requisito nesta solução, a configuração de um DSN do ODBC, nomeado estDB, apontando a Base de Dados.

Envolvido, como se disse, pela enumeração dos elementos do vector de DirSub, cada documento é criado com dois objectivos distintos: adicionar o caminho completo à tabela Documento e extrair as palavras respectivas, preenchendo as tabelas Palavra, PalavraOcorrencia e PalavraDocumento. O primeiro objectivo é conseguido pelo método adicionarDocumento() da classe Documento. O segundo consiste em duas fases.

A primeira fase trata de obter a string contendo o texto limpo de tags HTML.

A segunda fase isola os elementos de texto considerando como delimitadores um conjunto de caracteres que foram identificados, por tentativas sucessivas, como naturais delimitadores de palavras. O hífen aparece incluído nesse conjunto porque, embora haja palavras compostas com hífen na língua portuguesa, o objectivo desta base de dados comporta esse critério. Por outro lado o hífen surgia, com frequência, adjacente a palavras que, por essa via, eram diferenciadas de outras deformando grosseira e desnecessariamente a estatística.

Esta separação de palavras é executada por um objecto StringTokenizer. A enumeração dos tokens assim criados controla a inserção das palavras encontradas, bem como as respectivas frequências (total e por documento).

```
import java.io.*;
import java.util.*;

public class Preenche {
  public static void main(String[] args) throws Exception {
        String doc;
        System.out.println("Constituindo a lista da documentos");
```

```
        DirSub dir=new DirSub(args[0]);
        Enumeration e =  dir.v.elements();
        int t=dir.v.size();
        int d=0;
        System.out.println("Preenchendo a tabela de documentos");
        LigaDB db= new LigaDB();
        while (e.hasMoreElements()){
            d++;
            doc=(String) e.nextElement();
            System.out.println("Documento nº "+d+" de "+t+" "+doc);

// Adiciona path do documento à tabela Documento
            Documento documento= new Documento(doc, db);
            documento.adicionarDocumento();
// Leitura de texto
            String l=documento.texto();
            StringTokenizer st= new StringTokenizer(l,
                            " \n\"'.:?+,;-!_={}[]`´()«»");
               while (st.hasMoreTokens()) {
// Adiciona palavra
                  String p=st.nextToken().toLowerCase().trim();
                  if (p.length()>0) {
                    Palavra palavra=  new Palavra(p,db);
                    palavra.adicionarPalavra();
                    PalavraDocumento pd
                      = new PalavraDocumento(palavra,documento,db);
                    pd.incrementarOcorrencia();
                  }
              }
            }

    System.exit(1);

       }

}
-------------
import java.io.*;
import java.util.*;

public class DirSub {

  public DirSub(String a) {
    this.a=a;
    this.list=list;
    this.v= new Vector();
    try{
      File path= new File(a);
      if (path.isDirectory()){
        list=path.list();
        Vector n = listarFicheiros();
        Enumeration en= n.elements();
        while(en.hasMoreElements()){
          v.addElement(en.nextElement());
        }
        }
      else {
        v.addElement(this.a);
        return;
```

```
      }

    } catch (Exception e) {
      e.printStackTrace();
    }
  }
  private Vector listarFicheiros(){
  Vector vr= new Vector();
  for(int i=0; i<list.length;i++) {
      DirSub b=new DirSub(a+list[i].concat("\\"));
       if (b.list==null){
          vr.addElement(a+list[i]);
        }
        else {
        Enumeration er= b.v.elements();
        while (er.hasMoreElements()){
          vr.addElement(er.nextElement());
        }
        }
    }
  return vr;
  }
  public Vector v;
  private String list[];
  private String a;
}
-------------
import java.io.*;
import java.net.*;
import javax.swing.text.*;
import javax.swing.text.html.*;

public class Documento  {

  public Documento(String doc, LigaDB db)  {
  this.kit = new HTMLEditorKit();
  this.documento = kit.createDefaultDocument();
  // The Document class does not yet handle charset's properly.
  documento.putProperty("IgnoreCharsetDirective", Boolean.TRUE);

  this.doc=doc;
  this.db=db;
  }
  public String texto()  {
            try {
            // Create a reader on the HTML content.
            Reader rd = getReader(doc);
            // Parse the HTML.
            kit.read(rd, documento, 0);
            int i=documento.getStartPosition().getOffset();
            int f=documento.getEndPosition().getOffset();
            texto = documento.getText(i,f);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return texto;
  }
// Adicionar documento à tabela
  public void adicionarDocumento() throws Exception {
```

```
  String query="SELECT * FROM Documento WHERE PATHDocumento='" +doc+"'";
  if(!db.existeRegisto(query)) {
          String insereNovoDoc="INSERT INTO Documento "+
           "( PATHDocumento ) "+
           "VALUES ('"+doc+"')";
           db.escreveRegisto(insereNovoDoc);
    }
  }
    private Reader getReader(String uri) throws IOException {
        if (uri.startsWith("http:")) {
            // Retrieve from Internet.
          URLConnection conn = new URL(uri).openConnection();
            return new InputStreamReader(conn.getInputStream());
        } else {
            // Retrieve from file.
          return new FileReader(uri);
        }
    }

  public  String doc;
  private LigaDB db;
  private EditorKit kit;
  private Document documento;
  private  String texto;
}

-------------

import java.sql.*;

public class LigaDB  {
  public LigaDB() throws Exception {
      String dbUrl="jdbc:odbc:Estdb";
      String user="";
      String password="";
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      Connection c=DriverManager.getConnection(dbUrl,user,password);
      this.c=c;
      if (!c.isClosed()){
         c.close();
                c=DriverManager.getConnection(dbUrl,user,password);
      }
      stmt=c.createStatement();
      this.stmt=stmt;
  }
  public boolean existeRegisto(String query) throws Exception {
      this.rs = stmt.executeQuery(query);
      if (!rs.next()) return false;
      else return true;
  }
  public int getID(String query) throws Exception {
      this.rs = stmt.executeQuery(query);
      if (rs.next()) return rs.getInt(1);
      return 0;
  }
  public void escreveRegisto(String query) throws Exception {
      stmt.executeUpdate(query);
  }
  private Connection c;
```

```
  private Statement stmt;
  private ResultSet rs;
}

-------------
public class Palavra {

  public Palavra(String palavra, LigaDB db) {
    this.palavra=palavra;
    this.db=db;
  }
  public void adicionarPalavra() throws Exception {
      String query="SELECT * FROM PalavraQuery WHERE Palavra='"
+palavra+"'";
      if (!db.existeRegisto(query)) {
           String insereNovaPalavra="INSERT INTO PalavraQuery "+
           "( Palavra, NrOcorrencias ) "+
           "VALUES ('"+palavra+"', 1)";
           db.escreveRegisto(insereNovaPalavra);
           }
      else {
           String incrementarOcorrencia ="UPDATE PalavraQuery "+
           "SET PalavraQuery.NrOcorrencias = "+
           "[PalavraQuery].[NrOcorrencias]+1 "+
           "WHERE ((('"+palavra+"')=[PalavraQuery].[Palavra]))";
           db.escreveRegisto(incrementarOcorrencia);
          }
  }
  public String palavra;
  private LigaDB db;
}
-------------
public class PalavraDocumento {

  public PalavraDocumento(Palavra p,Documento d, LigaDB db) {
    this.palavra=palavra;
    this.db=db;
    this.p=p;
    this.d=d;
  }
  public void incrementarOcorrencia() throws Exception{
      String query="SELECT * FROM Documento WHERE PATHDocumento='"+d.doc
+"'";
      int idDoc=db.getID(query);
      query="SELECT * FROM Palavra WHERE Palavra='" +p.palavra+ "'";
      int idPalavra=db.getID(query);
      if (idPalavra==0 | idDoc==0) return;
      query="SELECT * FROM PalavraDocumento WHERE RefIdDocumento="
            +idDoc+ " AND RefIdPalavra="+idPalavra+"";
      if (db.existeRegisto(query)){
           String incrementarOcorrencia ="UPDATE PalavraDocumento "+
           "SET PalavraDocumento.NrOcorrencias = "+
           "[PalavraDocumento].[NrOcorrencias]+1 "+
           "WHERE ((("+idPalavra+")=[PalavraDocumento].[RefIdPalavra]) "+
           "AND (("+idDoc+")=[PalavraDocumento].[RefIdDocumento]))";
           db.escreveRegisto(incrementarOcorrencia);
           }
      else {
           String insereNovaPalavra="INSERT INTO PalavraDocumento "+
```

```
            "( RefIdDocumento, RefIdPalavra, NrOcorrencias ) "+
            "VALUES (" + idDoc + ", " + idPalavra + ", 1)";
            db.escreveRegisto(insereNovaPalavra);
            }
  }
  private String palavra;
  private LigaDB db;
  private Documento d;
  private Palavra p;
}
```
