UNIVERSIDADE DA BEIRA INTERIOR
Departamento de Informática

Agrupamento Automático de Páginas *Web* Utilizando Técnicas de *Web Content Mining*

Ricardo Nuno Taborda Campos

Dissertação apresentada na Universidade da Beira Interior para

obtenção do grau de Mestre em Engenharia Informática

Orientador: Professor Doutor Gaël Dias
Universidade da Beira Interior

Covilhã, 2005

Prefácio

Este documento entregue em Junho de 2005, contém uma dissertação intitulada "Agrupamento Automático de Páginas *Web* Utilizando Técnicas de *Web Content Mining*", um trabalho do aluno Ricardo Campos no âmbito do Mestrado em Engenharia Informática da Universidade da Beira Interior, com orientação do Professor Doutor Gaël Dias do Departamento de Informática da Universidade da Beira Interior.

O autor do trabalho é licenciado em Matemática/Informática pela mesma Universidade.

Agradecimentos

Devo muito do que sou a um conjunto de pessoas que ao longo do meu percurso académico por vezes sem se aperceberem, me apoiaram, motivaram e desafiaram a evoluir.

Gostaria assim, em primeiro lugar de agradecer a todos aqueles que, de uma ou de outra forma, tentaram dificultar o meu trabalho e condicionar o meu progresso. A minha evolução e crescimento fez-se também, da necessidade de ultrapassar obstáculos. O vosso contributo foi um verdadeiro estímulo para sorrir nos momentos menos bons.

Gostaria de agradecer ao Professor Gaël Dias. A sua orientação foi de um profissionalismo e dedicação ímpar, tendo sido também um bom amigo, mantendo-me motivado com o seu permanente acompanhamento durante este longo período de estudo. Foi desde os tempos de licenciatura uma referência para mim. Influenciou e continua a influenciar ainda hoje a forma como desempenho a minha profissão.

Ao Hugo Veiga, ao Guillaume Cleuziou e ao Alexandre Gil por me terem facultado as aplicações que desenvolveram.

Ao Professor João Muranho meu orientador de projecto e ao Professor Abel Gomes, meu orientador de estágio, que em fases diferentes, marcaram o meu percurso académico. A todos os restantes que contribuíram para a minha formação.

Aos responsáveis da Wedo Consulting - Decision: Luís Rodeia, Jorge Rodrigues e José Lourenço. Aos meus colegas de trabalho: Ana Duarte, Manuel Oliveiros, Ricardo Batista e António Matias. A Wedo foi uma escola de saber, companheirismo e profissionalismo, a que tive o privilégio de pertencer.

Aos meus colegas da Área Interdepartamental de Tecnologias de Informação e Comunicação do Instituto Politécnico de Tomar: Célio Marques, Vasco Silva, José Mendes e Paulo Ferreira, por entenderem que a carreira de Docente também se faz de investigação científica.

Gostaria de agradecer a alguns dos meus amigos que conheci na Universidade da

Beira Interior e que, de uma ou outra forma, influenciaram o meu percurso de estudante: à Cláudia Santos, à Fátima Silva, ao João Pedro, ao Miguel Batista, ao Vasco Fernandes, à Sandra Rodrigues, à Sónia Antunes e ao Paulo Martins.

À minha família mais próxima que são os meus amigos: Luís Nina, Ricardo Bichinho, João Seixas, Rui Pedro, Fausto Ramos, Ruben Pedro, Hugo Simões, Gonçalo Fiadeiro, Sérgio Fonseca, Daniel Dias, Hugo Sainhas e Sandra Pinto.

A ti Célia, pelo teu Amor, amizade, palavras de coragem e enorme apoio, pelos teus conselhos, por todo o conforto que comigo partilhaste durante estes últimos anos. A tua alegria, vivacidade e confiança, fez-me acreditar ser possível chegar aqui. Aos seus Pais Maria Lucinda e Bento Nunes.

Aos meus Pais, Arlindo Campos e Suzel Campos e à minha irmã Carla Campos, pelo vosso conforto, pela vossa sabedoria, pela vosso apoio, pelo vosso carinho, por aquilo que representam para mim, pelos valores que me transmitiram e estabilidade familiar que me proporcionaram. Vocês são o meu orgulho e a razão de aqui chegar. Ao meu cunhado José Manuel e à minha avó Berta Taborda.

Ao meu querido avô, Franklim Taborda e à sua memória, eu dedico esta tese. Esteja onde estiver, participará sempre das minhas conquistas. O meu sorriso final vai para ele.

Resumo

Com o massivo aumento da disponibilização de novos conteúdos na Internet, a pesquisa de informação tornou-se cada vez mais importante. Desde o seu início que os sistemas têm vindo a sofrer constantes desenvolvimentos e a ser alvo de investigação por parte de uma vasta comunidade científica.

Derivado do nosso estudo, implementamos um novo sistema de agrupamento e apresentação de resultados que advêm da procura da informaçao em ambiente *web*. Utilizamos técnicas de *Web Content Mining* para representar os textos a partir dos seus termos mais relevantes, que tanto podem ser palavras simples como palavras compostas. No seguimento, aplicamos um algoritmo de *soft clustering* que agrupa em *clusters* documentos relativos ao mesmo conceito, apresentando-os de forma hierárquica, *labelizados* com os seus termos mais relevantes. Este sistema evoluirá para uma nova vertente de *Query Expansion*, denominada de *Classified Query Expansion*. Será Automática (acrescentando termos à *query*) com a implementação progressiva de uma *WebWarehouse* (que guardará os termos relacionados a cada nova pesquisa efectuada) e será Interactiva (sugerindo ao utilizador um conjunto de resultados que se apresentam de forma hierárquica), propondo ao utilizador a expansão da *query* escolhendo um ou mais *clusters*.

A organização dos resultados, que decorre deste novo conceito facilitará a navegação do utilizador pela lista de páginas devolvidas por um motor de busca qualquer. Com uma ferramenta que automatize estes processos, o utilizador não mais necessitará de fazer uma procura exaustiva da página do seu interesse, o que significa um considerável ganho de tempo que o mesmo poderá dedicar a outras tarefas como sejam o estudo em concreto do site do seu interesse.

Conteúdo

Capítulo 1

Introdução

1.1 Motivação

A *WWW (World Wide Web*) contém uma quantidade enorme de informação, mas a sua globalização e a facilidade com que hoje se acede à Internet transformou-a numa rede de informação gigantesca. Na actualidade, os motores de busca confrontam-se com o problema de terem que ajudar o utilizador a lidar com mais informação do que aquela que na realidade consegue absorver. Na maioria dos casos, pela falta de organização da informação e não pelo seu excesso, acabamos por ignorar prováveis dados preciosos: lemos apenas umas quantas notícias de um jornal, fazemos uma procura na *web* e limitamo-nos na maioria dos casos aos primeiros 20 resultados (ver Silverstein *et al*, 1998) devolvidos da execução da pesquisa.

A localização e organização de recursos com conteúdo relevante e de qualidade é uma tarefa complicada. O conceito de pesquisa de informação (*Information Retrieval*[1]) surge neste contexto como um processo onde são devolvidos e ordenados por ordem de importância os documentos mais relevantes de acordo com uma pergunta (*query*) especificada pelo utilizador. Depois de completada a pesquisa, o conjunto de documentos é

[1]Que a partir de agora definiremos como IR.

dividido em 2 grupos (ver Figura 1.1):

(1) o conjunto dos documentos devolvidos;

(2) o conjunto dos documentos omitidos pelo sistema.

cada qual dividido em documentos considerados relevantes ou não relevantes de acordo com a *query*.

	Relevantes	**Não Relevantes**	
Devolvidos	a	b	(a +b) todos os documentos devolvidos
Não Devolvidos	c	d	(c +d) todos os documentos deixados de fora
	(a+c) todos os documentos relevantes na colecção	(b +d) todos os documentos não relevantes na colecção	(a +b +c +d) toda a colecção

Figura 1.1: *Divisão dos documentos devolvidos em 2 grupos: relevantes e não relevantes.*

A avaliação deste tipo de sistemas é sempre sujeita a uma certa subjectividade (o que pode ser relevante para um utilizador pode não ser para outro) mas é normalmente feita com recurso a três medidas:

(1) *precision* (precisão): avalia de entre todos os documentos devolvidos os que são relevantes;

(2) *recall* (cobertura): avalia de entre o universo de todos os documentos relevantes aqueles que são devolvidos pelo sistema;

(3) *fallout*: avalia de entre o universo de todos os documentos não relevantes quais os que foram devolvidos pelo sistema.

As seguintes equações ilustram estas medidas:

$$precision = \frac{a \times 100}{(a+b)} \tag{1.1}$$

$$recall = \frac{a \times 100}{(a + c)} \tag{1.2}$$

$$fallout = \frac{b \times 100}{(b + d)} \tag{1.3}$$

A *query* (lista de palavras conjugadas opcionalmente com operadores) e as características definidas pelo autor do documento para o caracterizarem, são normalmente os dois itens que permitem averiguar a sua similaridade e desta forma proceder a uma classificação que defina a sua importância no contexto de todos os documentos devolvidos.

A relevância dos documentos obtidos pode no entanto ser virtual e não satisfazer as necessidades do utilizador. Como descrito em Xu & Croft (1996), o maior problema associado à temática da pesquisa de informação passa por entender que os utilizadores possam usar um conjunto de palavras diferentes para descrever um conceito, comparativamente ao conjunto das palavras usadas pelos autores das páginas para descrever esse mesmo conceito, não havendo desta forma um ponto de intersecção entre os dois. Tal problema conhecido como ambiguidade genuína é um problema central no contexto da pesquisa de informação e agrava-se quando o utilizador não está familiarizado com o que procura.

Assim, o sucesso ou o insucesso do processo de pesquisa de informação, passa pela aproximação entre as palavras definidas na *query* e as palavras definidas pelos autores das páginas. Neste contexto, ajudar o utilizador a definir a pergunta aumenta as hipóteses de obter documentos relevantes. A concretização deste processo passa pelo conceito de *Query Expansion*, onde à pergunta definida pelo utilizador são acrescentadas palavras ou grupos de palavras (*phrases* ou n-gramas[2]) com significado semelhante, ou pelo menos relacionado[3], procedendo-se a um refinamento da mesma, tentando limitar desta forma a área de pesquisa.

[2]Expressão relevante, correspondente a uma sequência contígua, ou não contígua, de unidades lexicográficas (tokens) na língua em que o corpus está expresso.

[3]Por relacionado, entendem-se as relações presentes em ontologias ou thesaurus.

Este conceito assume duas vertentes: a *Automatic Query Expansion* e a *Interactive Query Expansion* (ver Vechtomova *et al*, 2004). Como o nome indica, a primeira vertente é um refinamento automático da *query* por parte do sistema enquanto a segunda depende da interacção com o utilizador. Alguns motores de busca já proporcionam aspectos de *query expansion* aos seus utilizadores mas apresentam no entanto dois problemas fundamentais:

(1) não interpretam o conteúdo dos documentos no seu contexto geral da língua (i.e., tendo em conta as ambiguidades da linguagem segundo o contexto tratado);

(2) e como consequência não apresentam a informação de uma forma estruturada, isto é, classificada.

Em resumo, os sistemas não estão, por um lado, capacitados para entender o que os utilizadores procuram (podem nem procurar nada em específico) e por outro devolvem um conjunto enorme de informação não estruturada.

Podemos ilustrar este problema da seguinte forma: suponha-se o caso de um jovem adepto benfiquista que ao efectuar uma pesquisa relativa ao tema, obtém um conjunto de documentos relativos ao clube, aos jogadores, mas também relativos ao bairro de Benfica em Lisboa e à venda de casas no mesmo. A ambiguidade do termo benfica que tanto pode estar relacionado com o bairro como com o clube, resulta na devolução de dois tipos de documentos que aparecem de forma não estruturada, dificultando a pesquisa de informação ao utilizador.

Esta dissertação tenta ultrapassar as limitações acima referidas. Assim, neste trabalho propomos dar uma resposta a estes problemas, desenvolvendo um sistema que delegue ao motor de busca a fastidiosa tarefa de organizar a informação dispersa por entre várias páginas de resultados, tornando o processo anteriormente realizado pelo utilizadores, num processo automático. Consequentemente, a procura de informação em grandes bases de

dados, em particular a informação existente na *WWW*, ficará grandemente facilitada, uma vez os documentos agrupados em *clusters* (ver Willet, 1990), permitindo aos utilizadores escolherem os conceitos desejados.

Desenvolvemos assim, no âmbito da dissertação, a aplicação **WISE (Web Interactive Search Engine)** disponível em *http://wise.di.ubi.pt*. Com recurso a técnicas de *Web Content Mining* associadas a técnicas de *Clustering*, o sistema WISE apresenta-se como um *Meta Crawler* que apresenta, de forma hierárquica, a informação proveniente de um qualquer motor de busca.

Utilizando técnicas de *Web Content Mining* desenvolvidas no âmbito da aplicação WebSpy (ver Veiga *et al*, 2004) e agrupando os documentos com base no algoritmo de *Soft Clustering* Poboc (ver Cleuziou *et al*, 2004), aumentamos a qualidade da visualização dos resultados ao usar uma estrutura hierárquica, colocando os documentos num ou mais *clusters*, mostrando ao utilizador a respectiva descrição de cada grupo (através de um *label*[4] dando-lhe assim a possibilidade de mais facilmente escolher o(s) URL(s) do seu interesse.

1.2 Contribuição

Numa primeira fase a partir da aplicação WebSpy (ver Veiga *et al*, 2004) que implementa uma árvore de decisão, são extraídos automaticamente termos relacionados com a *query* vasculhando as páginas devolvidas pela execução da mesma, o que conseguimos com recurso ao desenvolvimento e implementação de um *spider*[5].

Numa segunda fase são aplicadas técnicas de *soft clustering* hierárquico, através do algoritmo Poboc (ver Cleuziou *et al*, 2004), para agrupar e desambiguar os termos rela-

[4]Palavra simples ou composta caracterizadora/indicadora de cada um dos seus documentos

[5]Componente do motor de busca que percorre os documentos (percorrendo os links), adicionando ao índice os URLs, palavras e texto que encontra.

cionados, previamente extraídos, permitindo assim a organização lógica e a classificação dos documentos relevantes para com a *query*.

Assim, em contraponto com a actual classificação de páginas relevantes que se baseia na comparação entre os termos definidos na *query* pelo utilizador e o conjunto de características do documento, a análise semântica do seu conteúdo aumentará o universo de palavras a comparar, havendo lugar a uma maior aproximação entre as pretensões do utilizador e o que a *Web* lhe tem para oferecer. Acresce a isto, que uma das inovações do projecto reside no facto do sistema vasculhar não só a página devolvida pelo motor de busca, mas, e no caso de se tratar de um endereço absoluto, também as 10 melhores páginas (de acordo com a *query*) do site ao qual a página de resposta pertence, permitindo que a solução seja mais consistente, dado que a procura de termos relacionados passa a ser feita com base num maior conjunto de resultados.

A plataforma desenvolvida é além do mais flexível na sua adaptação ao mundo real, pelo facto de ser independente da língua e do domínio/contexto dos textos. De certa forma, a solução apresentada encontra-se entre os 2 domínios de *Query Expansion*, podendo entender-se como uma evolução da *Interactive e Automatic Query Expansion*: *Interactive* na medida em que o utilizador poderá seleccionar os resultados que mais lhe interessar a partir dos termos relacionados apresentados de forma hierárquica [6] e utilizar estes termos para a extensão da *query*[7], *Automatic* no sentido em que, uma vez integrada toda a informação numa *WebWarehouse*[8] com a finalidade de definir gradualmente um *thesaurus*, a mesma poderá ser utilizada no refinamento automático da *query* com consequências ao nível da performance do sistema. Chamar-lhe-emos de *Classified Query Expansion*.

Neste sentido, assistir-se-á a uma evolução nos motores de busca que deixarão de ser simples ”páginas amarelas” (lista de resultados) para passarem a ser um catálogo de

[6] Funcionalidade já implementada

[7] Funcionalidade a implementar no âmbito de trabalho futuro.

[8] Futuramente implementada à medida da utilização do *software* WISE.

conteúdos (lista de resultados remissivos) como propõem Ferragina *et al* (2005).

1.3 Plano da Tese

Do entendimento destas necessidades surge o nosso projecto estruturado da seguinte forma:

No próximo capítulo é feita uma descrição do trabalho relacionado: contextualizamos no âmbito do trabalho a desenvolver os fundamentos teóricos da dissertação e apresentamos paralelamente um levantamento da investigação realizada até ao momento na área.

No capítulo três explicamos a nossa contribuição na área.

No quarto capítulo, explicamos a arquitectura global do projecto.

No quinto capítulo, definimos as técnicas de *clustering* utilizadas para o agrupamento das páginas *web*, em particular o algoritmo de *soft clustering*, Poboc (ver Cleuziou *et al*, 2004).

Finalmente apresentamos uma conclusão da dissertação e um conjunto de propostas para trabalhos futuros.

Capítulo 2

Trabalho Relacionado

Descrevemos neste capítulo um conjunto de conceitos resultantes de um apurado trabalho de investigação, do que melhor tem sido feito na área e do que ainda há para fazer. A sua leitura permitirá entender o trabalho relacionado, contribuição e familariedade com termos, conceitos e linguagem utilizados na dissertação.

2.1 Motores de Busca

Os Motores de Busca são uma ferramenta essencial para encontrar algo na Internet. São também a face mais visível da investigação efectuada em IR, mas ainda se procura o sistema ideal que garanta resultados mais precisos.

Um sistema de busca é um conjunto organizado de computadores, índices, bases de dados e algoritmos, reunidos com a missão de analisar e indexar as páginas *web*, armazenar os resultados dessa análise e indexação numa base de dados e devolvê-los posteriormente aquando de uma pesquisa que preencha os requisitos indicados pelo utilizador por ocasião de uma consulta. As suas funções são portanto as de *crawling*, *indexing* e *searching*.

Cada motor de busca conta com as suas particularidades resultante de diferentes filosofias e procedimentos no desenvolvimento do software que o suporta. Se usarmos sucessivamente motores de busca diferentes para encontrar informação tendo por base o mesmo termo ou conceito, tanto poderemos obter respostas substancialmente diversas como poderemos reconhecer muitas das já apontadas pelo motor de busca anterior. Surgiram por isso os Meta Motores cuja pesquisa é direccionada para um conjunto alargado de Motores de Pesquisa.

Por via das diferentes particularidades, os motores de busca podem ser enquadrados em dois distintos processos de funcionamento:

(1) Motores de Busca alimentados manualmente;

(2) Motores de Busca alimentados por *crawlers/spiders/robots*.

No 1^o caso, a base de referências do motor de busca é alimentada manualmente (*Crawler* manual), ou seja, por pessoas que pesquisam, catalogam em directórios e verificam a continuidade das páginas, ou em alternativa, por pessoas que analisam os pedidos de submissão levados a efeito pelos interessados em publicar um site. Exemplos destes tipos de motores de busca são o DMOZ[1], LookSmart[2], Zeal[3], etc..., mas o mais popular deles todos, continua a ser o Yahoo[4]. Embora continuando a funcionar nestes moldes, o Yahoo, activo desde o ano de 1994, passou a combinar, desde Fevereiro de 2004, o *crawler* automático com o manual.

Assim, ao invés de serem alimentados manualmente, os motores de busca podem ser alimentados por um robot. Como o próprio nome indica este processo é alimentado por um programa denominado *Spider* ou *Crawler* (ver Page & Brin, 1998) que percorre

[1] www.dmoz.org

[2] www.looksmart.com

[3] www.zeal.com

[4] www.yahoo.com

constantemente a Internet, dissecando-a, saltando de página para página (seguindo os hyperlinks), catalogando e armazenando as páginas que vai encontrando.

Tudo o que o *spider* encontrar vai para a segunda parte do motor de busca, o *index*. Muitas das vezes conhecido como catálogo, o *index* é como um livro gigante que contém uma cópia de todas as páginas *web* que o *spider* encontra. As mesmas poderão ser usadas para construir um índice invertido de palavras-chave para posterior classificação dos documentos em directorias, ou para a construção de um grafo de hyperlinks por forma a desenvolver um *ranking* de links (ver Page *et al*, 1998).

O motor do motor de busca, é a terceira parte do software. Este é o programa que percorre o índice invertido de forma a encontrar os milhões de páginas guardados no *index* para encontrar correspondências com a procura.

Se um documento contiver os termos de pesquisa, existe uma boa probabilidade do documento interessar ao utilizador (ver Costa & Silva, 2001). Assim, na maioria dos casos o processo de definir a relevância de uma página baseia-se na procura de descrições das páginas tanto no código (*URL; Título do site; Meta Keywords; Meta Description; Headers; Links; ALT tags;* etc...), como no próprio conteúdo da página, na maioria dos casos calculando a relevância dos termos através do TF.IDF[5].

As palavras-chave (*Meta Keywords*) em links são também a garantia que a página está ligada a outras páginas (ou sites) sobre o assunto, o que garante para o motor de busca a relevância da página.

Outra das funções do *index*, é determinar essa mesma relevância da página, quando confrontado com milhões de páginas para ordenar, atribuindo-lhes um *rank* de acordo com os seus parâmetros de relevância.

Na perspectiva de Costa & Silva (2001), a definição dos parâmetros de relevância de uma página pode ser feita com recurso a 3 tipos de algoritmos, consoante a informação que analisam:

[5]Term Frequency/Inverse Document Frequency (ver Salton *et al*, 1975)

(1) algoritmos de análise de conteúdo;

(2) algoritmos de análise de estrutura;

(3) algoritmos de análise dos dados de interacção.

Assim, o *ranking* das páginas é uma etapa essencial de cada motor de busca. Altavista[6] e Nothern Light[7] continuam a utilizar as técnicas tradicionais de ranking, outros como o Hotbot[8] combinam estas com um *score* popular que regista os links que os utilizadores mais clicam e o tempo que passam em cada um deles (algoritmos de análise dos dados de interacção: *Web Usage Mining*).

Note-se que já em 1998 é feita referência à necessidade de melhorar o processo de *Information Retrieval* e desenvolver um sistema mais preciso (ver Page & Brin, 1998).

Embora se obtenham, em segundos, referências a milhares de sites ordenados de forma que os mais relevantes apareçam em primeiro lugar, muitas das vezes não encontramos o que procuramos. As páginas não relevantes são uma realidade e temos por isso que vasculhar umas quantas páginas de resposta até encontrarmos numa referência bastante mais avançada aquilo que realmente mais se aproxima do que pretendemos, obrigando o utilizador a perder muito tempo até obter a resposta desejada (ver Baeza-Yates & Ribeiro-Neto, 1999). Mesmo que se gastem apenas 30 segundos a examinar cada um dos 100 resultados devolvidos, o utilizador perderá 50 minutos, se entretanto não desistir!

O motor de busca Google[9] deu um passo nesse sentido inovando na maneira como atribui um *rank* a uma página (algoritmos de análise de estrutura).

Já os algoritmos de análise de conteúdo, continuam a ser utilizados sem ter em atenção qualquer validação semântica. Na perspectiva de Costa & Silva (2001), esta análise continua a sofrer de 2 problemas:

[6] www.altavista.com

[7] www.northernlight.com

[8] www.hotbot.com

[9] www.google.com

(1) ambiguidade: dado um termo de pesquisa este pode ter vários significados;

(2) sinónimos: os documentos podem conter apenas termos sinónimos do termo de pesquisa, não sendo por isso encontrados no processo.

Mais do que desenvolver novos algoritmos de *ranking* a comunidade científica em *Information Retrieval* tem direccionado a sua atenção para os algoritmos já existentes. No âmbito deste trabalho, do estudo das lacunas e das potencialidades dos actuais processos de classificação e organização de resultados, resulta que os actuais algoritmos devem continuar a ser considerados, aos quais se devem acrescentar novas abordagens: o entendimento dos documentos, a ajuda ao utilizador no âmbito da definição da *query* e a organização do conjunto de resultados, que não uma lista ordenada dos mesmos.

Uma leitura atenta destas linhas permite entender que a aplicação de algoritmos de *clustering* aos resultados devolvidos, beneficia não só a organização dos resultados, mas também o *ranking* dos mesmos, uma relação causa-efeito decorrente da proximidade existente entre ambos.

Com esta secção pretendemos conferir ao trabalho um entendimento teórico, saber como os sistemas funcionam, saber quais as suas virtudes e limitações, criar uma base de conhecimento que contextualize a necessidade de desenvolver novos estudos e implementações. Este entendimento prossegue com o estudo do motor de busca Google, na actualidade o sistema mais utilizado.

2.1.1 Google: Metodologia de Extracção de Termos, Indexação e Classificação de Páginas Relevantes

Depois de em 1998 ter sido lançado comercialmente por dois colegas (ver Page & Brin, 1998) da Universidade de Standford nos EUA, o Google rapidamente cresceu em popularidade e lucro: de 10 mil pesquisas em 1998 para 200 milhões em 2004, com mais de 1900 funcionários e recorrendo a 100 mil servidores, o Google é composto por uma

série de *crawlers*, os *GoogleBots*, distribuídos por várias máquinas e um servidor de URL que envia listas de URLs para os *crawlers* procurarem. Como os *crawlers* seguem os links de uma página para outra, o motor de busca consegue encontrar milhões de páginas. Esta técnica conhecida como *deep crawl* é extremamente poderosa mas consome bastante tempo, razão pela qual os *crawlers* robotizam as páginas apenas de mês a mês, sendo que o Google robotiza com maior frequência, as páginas mais frequentemente actualizadas.

As páginas encontradas pelos *crawlers* são depois compactadas e guardadas num repositório, ficando associado a cada página um *ID* designado por *DocID*, o tamanho da mesma e o URL. A função de *indexing* é feita pelo *indexer*, que descompacta os documentos percorre-os e converte-os num conjunto de palavras (*WordID*) juntamente com a sua posição no texto, documento a que pertence, etc... O *index* é ordenado alfabeticamente por termo, com cada entrada do *index*, guardando, numa estrutura designada por *Inverted Index*, a lista de documentos nos quais o termo aparece. Mas o *indexer* tem outra função que é a de percorrer a página à procura de links para outras páginas, com o *URLResolver* a converter os URL relativos para absolutos.

Para o utilizador poder usar o sistema, existe uma interface, um motor que procura as correspondências entre as *queries* e os documentos relevantes, e um conjunto de páginas geradas dinâmicamente com os resultados finais. Estes 3 componentes são conhecidos no seu conjunto pelo processador de *queries* (ver Figura 2.1).

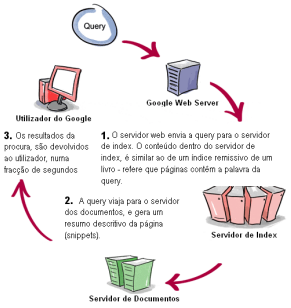

Figura 2.1: Processamento de uma *query* pelo Google.

A determinação da importância de uma página passa por técnicas complexas de correspondência textual, tendo em atenção o *WordID*, *DocID*, a posição da palavra no documento e outras considerações, algumas já descritas na secção acima comuns a muitos outros motores de busca, outras particulares ao Google (em particular a estrutura hipertexto).

Quando por exemplo o utilizador procura por múltiplas palavras-chave, o Google procura encontrar todas estas palavras no documento, isto é, se a procura for:

”Benfica Eusébio Glorioso”,

todas as páginas nas quais estas 3 palavras apareçam recebem uma pontuação **x**. O Google de seguida mede a distância entre as palavras e atribui às páginas uma pontuação **y**. Por exemplo, o texto

O **Benfica** *com* **Eusébio** *era um clube do mais* **Glorioso**

receberá uma pontuação superior a uma página que tenha o seguinte texto:

O **Benfica** *é um clube de nível mundial muito por culpa dos tempos em que o* **Eusébio** *jogava, esse ponta de lança* **Glorioso**.

Depois, o Google mede o número de vezes que uma palavra aparece na página e atribui-lhe uma pontuação **z**. Uma página que tenha a palavra Benfica 4 vezes, Eusébio 3 vezes e Glorioso 2 vezes, tem uma melhor pontuação que uma página que tenha cada uma dessas palavras apenas uma vez.

Estas 3 variáveis, **x(Phrases)**, **y(Adjacência)** e **z(Pesos)** em conjunção com outras 100 permitem definir a relevância das páginas.

A grande inovação deste sistema diz respeito ao incremento da qualidade das páginas devolvidas ao utilizador. Cada página é multiplicada pela pontuação do cálculo obtido da utilização do algoritmo *PageRank* (ver Page *et al*, 1998), obtendo-se desta forma uma classificação final mais conseguida, de acordo com os intentos do utilizador.

O algoritmo calcula a qualidade (ou relevância) de uma página, tomando a vasta estrutura de links da *Web* como um indicador do seu valor, através do número de links que apontam para ela (uma contagem de quantas páginas possuem uma ligação para essa página), num processo em tudo semelhante à indicação do prestígio de um *paper*, que é tanto maior quanto mais citações tiver. Esta análise matemática complexa interpreta, um link da página A para a página B, como um voto da página A para a página B.

Suponha-se um pequeno universo de 4 páginas: **A**, **B**, **C** e **D**. Assuma-se que a página **A** tem 3 páginas (i.e citações) que apontam para ela (**B**, **C** e **D**). Suponha-se também que a página **B** tem um link para a página **C** e que a página **D** tem um link para as outras três páginas (ver Figura 2.2):

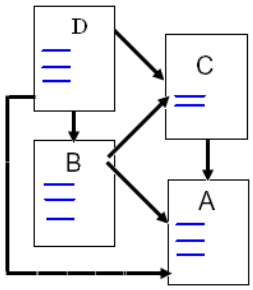

Figura 2.2: Estrutura de links.

Assim, o **PR** *(PageRank)* da página **A** seria a soma dos **PR** das páginas **B**, **C** e **D**.

$$PR(A) = PR(B) + PR(C) + PR(D)$$

Como uma entidade não pode votar duas vezes, considera-se que a página **B** atribui meio voto a cada um, na mesma lógica apenas um terço do voto de **D** é contado pelo *PageRank* de **A**:

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$$

Por outras palavras, o **PR** deve ser dividido pelo número de links (L(B), L(C) e L(D)) que saiem de uma página:

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

Acresce a este cálculo da quantidade de votos, que a importância da página donde é proveniente o link é também ela tida em conta no cálculo do *PageRank*. O algoritmo trata ambos os casos através da propagação recursiva de pesos pela estrutura de links *web* (ver Figura 2.3).

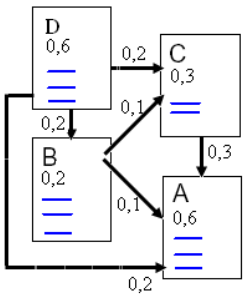

Figura 2.3: Propagação recursiva de pesos pela estrutura de links.

Baseado na descrição acima efectuada é possível observar o seguinte: uma página terá um *rank* elevado se a soma dos *ranks* dos *backlinks*[10] for elevada, ou se por outro lado a página tem poucos *backlinks* mas esses, com elevado *rank*.

O problema do *PageRank* é que apenas usa a estrutura da *Web* para estimar a qualidade de uma página, o que é manifestamente insuficiente!

[10]Links provenientes de outras páginas.

2.2 Query Expansion

No contexto da pesquisa de informação, um dos maiores problemas com que os motores de busca lidam, diz respeito à ambiguidade dos termos de pesquisa: se por um lado o utilizador nem sabe por vezes o que procura, levando-o a inserir termos ambíguos, por outro lado termos como *Porto* são tão gerais que podem aparecer em vários contextos: a cidade, o clube, o porto de pesca, etc..

O campo de pesquisa é de tal forma enorme que se pode dar o caso do motor de busca não retornar qualquer documento no contexto do interesse do utilizador.

Uma das formas que os motores de busca usam para lidar com este tipo de problemas é a contextualização do que o utilizador pretende, adicionando termos ou *phrases* que representem o contexto.

A *Query Expansion* também conhecida por *Query Refinement*, é um mecanismo incremental que recomenda ao utilizador uma nova *query* mais próxima das necessidades do utilizador. O utilizador obtém o conjunto de resultados e um conjunto possível de novas *queries* que quando seleccionadas provocam a obtenção de novos resultados. Na perspectiva de Liu *et al* (2003), este método apresenta no entanto o problema de retornar *web pages* mais de acordo com o contexto. Tal depende, na nossa perspectiva, se o contexto adicionado à *query* é abrangente ou específico, o que depende da qualidade dos resultados inicialmente obtidos, os quais com a aplicação de técnicas de *Web Content Mining* acreditamos poder vir a melhorar.

Outras propostas (ver Chekuri *et al*, 1997; Zeng *et al*, 2004) vão no sentido de utilizar as taxonomias existentes nos directórios *Web*, como os *Yahoo Directories*[11], de uma forma isolada ou em conjunto com *clustering*. Neste sentido seriam adicionadas categorias à palavra definida na *query*, abordagem que na nossa perspectiva compreende 2 problemas:

(1) demasiado abrangente, uma vez que uma categoria é demasiado extensa para permitir contextualização;

[11] http://dir.yahoo.com

(2) dependência das categorias previamente definidas.

Alguns motores de busca oferecem de uma ou outra forma *Query Expansion*. Geralmente comparam a *query* com um dicionário, *thesaurus* ou mesmo uma Ontologia, que mantêm, sugerindo posteriormente o(s) respectivo(s) sinónimo(s) e outros sugerem *queries* relacionadas com a *query* original através do processo de *blind relevance feedback* (ver Baeza-Yates *et al*, 1999).

Cada um destes diferentes motores de busca, com recurso a diferentes técnicas, assume uma abordagem de *Automatic Query Expansion* sugerindo um conjunto de termos possíveis para adicionar à *query*.

Ferragina *et al* (2005), assume uma abordagem interactiva, (*Interactive Query Expansion*), onde o utilizador desempenha o papel principal, seleccionando a partir do conjunto de resultados (termos relacionados apresentados de forma hierárquica), os resultados que mais lhe interessar, refinando a *query* com base nos mesmos. Assume também uma abordagem automática, propondo a lista dos documentos mais relevantes que contêm os *labels* dos *clusters* por simples *pattern matching*.

A nossa proposta assume-se também como automática e interactiva. Como referimos na secção trabalhos futuros, será automática à medida da utilização da aplicação, que registará numa *WebWarehouse* o conjunto de termos relevantes extraídos automaticamente a partir de técnicas de *Web Content Mining*. A proposta interactiva, decorre de uma nova apresentação de resultados que implementamos no nosso trabalho: uma estrutura hierárquica de termos semânticamente relevantes com a *query*. Desta apresentação de resultados, o utilizador poderá seleccionar os termos ou *clusters* que mais lhe interessar para a reformulação da *query*.

Designamos o conjunto das 2 abordagens por *Classified Query Expansion*.

2.3 Web Content Mining para Representação de Documentos

"Web Mining is the intersection between Data Mining and the World Wide Web, using Data Mining techniques to automatically discover and extract information from web documents."

in, O.R. Zaiane (1998).

2.3.1 Web Mining

A utilização cada vez mais intensa da *WWW* e a disponibilização diária de diversos tipos de conteúdos, tornou-a um local privilegiado de partilha de conhecimento, um repositório de informação que a comunidade académica e científica rapidamente se prestou a estudar. A junção destas duas áreas, *Data Mining* e Internet, deu origem ao conceito de **Web Mining** (ver Figura 2.4). Utilizando técnicas de *Data Mining* amplamente usadas em mercados financeiramente poderosos, como os bancos e telecoms, criou-se um ambiente paralelo ao da usual utilização de *Data Mining* ao importar as suas técnicas para o domínio da Internet.

Figura 2.4: *Data Mining* é a identificação de padrões na informação.

A ideia é usar este conhecimento de forma inteligente. Uma definição simplista (ver Kosala & Blockeel, 2000), por um lado descreve a necessidade de criar aplicações de

procura automática e pesquisa de informação, interpretando o conteúdo dos milhares de recursos disponíveis on-line, i.e, **Web Content Mining**, e por outro lado descreve a possibilidade de analisar os padrões de utilização da *Web* por parte dos utilizadores, i.e, **Web Usage Mining**, pondo-os ao serviço dos profissionais que gerem os vários centros de negócio emergidos da Internet (ver Figura 2.5).

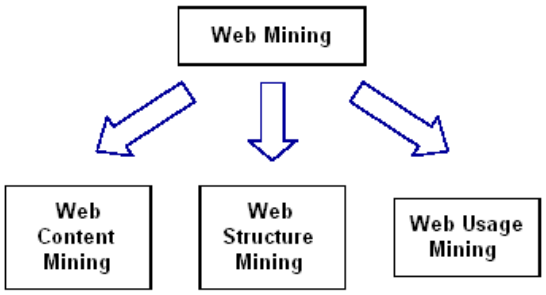

Figura 2.5: Taxonomia de *Web Mining*.

O *Web Usage Mining* (ver Kosala & Blockeel, 2000; Zaiane, 1998) é o processo de extrair padrões interessantes relativos ao comportamento dos utilizadores durante a navegação na *Web*, analisando os *Web Logs*. Este tipo de processo permite a análise de tráfego, a personalização e a criação de *Web* sites adaptados à realidade de cada utilizador. Convém no entanto referir que a realidade do utilizador muda constantemente, derivado da sua volatilidade. Se num âmbito de um trabalho hoje consultamos determinados artigos, de tal não se deve inferir que sejamos leitores interessados de artigos comuns na área. O interesse numa competição desportiva pode também ser apenas pontual e não deve servir para fazer regra. Perante estas dificuldades, o *Web Usage Mining* na vertante da personalização de sites não é uma área atractiva. A melhor personalização é feita pelo próprio utilizador.

Já o *Web Structure Mining* (ver Kosala & Blockeel, 2000; Zaiane, 1998) que se encontra a meio caminho entre o *Web Content Mining* e o *Web Usage Mining*, revela-se mais interessante. Infere conhecimento de como a *WWW* se organiza, explorando a sua

estrutura através dos hyperlinks. Através destas ligações entre documentos de hypertexto, a *WWW* pode revelar mais informação que apenas a contida nos documentos. Métodos como o *PageRank* (ver Page *et al*, 1998), fazem uso deste tipo de processos.

Finalmente o *Web Content Mining* é o processo que extrai conhecimento da *Web*, analisando o conteúdo dos seus documentos. A este propósito veja-se a secção seguinte.

2.3.2 Representação dos Documentos

A maioria dos motores de busca apresentam frequentemente os mesmos problemas (ver Kosala & Blockeel, 2000): baixa precisão e baixa cobertura. O primeiro deriva da irrelevância de muitos dos resultados devolvidos o que torna difícil encontrar informação relevante. O segundo manifesta a incapacidade de indexar toda a informação disponível na *Web*, permitindo que documentos relevantes fiquem fora da pesquisa.

Por outro lado, e apesar de disponibilizarem determinado conforto aos seus utilizadores, a maioria dos motores de busca não apresentam a informação estruturada ou categorizada e nem sequer interpretam os documentos (ver Cooley *et al*, 1997), limitando-se a devolver os que contêm as correspondentes palavras definidas na *query*.

Enquanto o problema inicialmente descrito (baixa precisão e baixa cobertura) é um problema de pesquisa (*retrieval process*), o problema de querer criar conhecimento a partir da informação disponível na *web*, interpretando os documentos, é um problema de informação (*mining the web process*), e pouco tem sido feito nesta área.

Os motores de busca continuam por isso a devolver a lista de resultados classificada de acordo com o tópico/palavra de pesquisa e não com a descrição da página, entendimento ou definição. Continua por medir o valor informativo das páginas *web* e continuamos a obter resultados não informativos nos primeiros lugares de uma pesquisa.

O *Web Content Mining* surge neste contexto como um processo que extrai conhecimento da *Web*, analisando o conteúdo dos seus documentos (ver Kosala & Blockeel, 2000; Zaiane, 1998).

Para a representação dos mesmos conhecem-se 3 abordagens conceptuais:

(1) representação de cada documento, por partilha de n-gramas[12] entre os documentos;

(2) representação de cada documento, com cada palavra da colecção sendo um atributo e a sua relevância registada através do TF.IDF. Esta técnica é conhecida por *Vector Space Model* (ver Salton *et al*, 1975);

(3) representação de cada documento com base num conjunto de termos relevantes capazes de o caracterizar, calculados com o recurso a propriedades definidas no âmbito do *Web Content Mining* (i.e., a relevância de um termo é definida por mais características que apenas o TF.IDF).

Veja-se a descrição de cada um destes itens nas secções seguintes.

Partilha de n-gramas

Trabalhos recentemente publicados não utilizam qualquer técnica de *Web Content Mining* no processo de caracterização dos documentos da colecção, nem utilizam a medida TF.IDF: os termos que caracterizam a colecção são todos aqueles termos partilhados por mais do que um documento.

No caso de Zamir *et al* (1998), é utilizada uma estrutura designada por STC (*Suffix Tree Clustering*) para determinar os termos partilhados por mais do que um documento. Não é utilizado por estes investigadores qualquer outro tipo de cálculo para determinar *phrases* de modo que os termos partilhados por documentos distintos assumem-se eles próprios como *phrases*/n-gramas. Os investigadores acreditam que a partilha de n-gramas entre documentos, é um método suficientemente informativo de sumarizar os seus conteúdos, mas aconselham aprofundar esta questão para trabalhos futuros.

Zhang *et al* (2001), utilizam uma estrutura de dados conhecida por *suffix array* e criticam Zamir *et al* (1998), por estes utilizarem um *suffix tree*, na medida em que esta

[12]Conjunto de n palavras contíguas ou não.

última estrutura é dependente da língua, pois para dialectos com mais caracteres que o inglês (caso do Chinês com cerca de 6000 caracteres) o sistema torna-se inviável em termos de tempo de execução.

Assim, depois de obtidos e registados num *suffix array* os termos e suas respectivas frequências, o algoritmo que Zhang *et al* (2001) designaram por SHOC (*Semantic, Hierarchical, Online Clustering of Web Search Results*), extrai *phrases*/n-gramas através da aplicação de técnicas de *Web Content Mining* sobre o conjunto de termos que aparecem mais frequentemente no texto, utilizando 3 medidas (*completeness; stability (mutual information) e significance*).

Jiang *et al* (2002), na procura de termos que caracterizem os documentos, fazem uso do método n-grama caracter a caracter, procurando depois aqueles que aparecem mais frequentemente.

Zeng *et al* (2004), utilizam a mesma estrutura que Zamir *et al* (1998), mas apenas consideram um termo como sendo um n-grama (com n < 4), se o mesmo ocorrer mais do que 3 vezes em documentos distintos, ao qual acresce o cálculo de um conjunto de propriedades, a saber: a ponderação semântica (IDF[13]) da *phrase*, o seu tamanho, a similaridade entre *clusters* para medir entre os documentos que contêm a *phrase* o quanto compactos eles são, a *cluster entropy* para representar o quanto distinta uma *phrase* é, e a independência da *phrase*. Dadas as propriedades, um modelo de regressão linear é aplicado com base num conjunto de treino, advindo deste cálculo a identificação dos n-gramas relevantes.

Vector Space Model

Martins *et al* (2003) e Fung *et al* (2003), representam os seus documentos como vectores e utilizam apenas o cálculo do TF.IDF para determinar o conjunto de termos rele-

[13]Inverted Document Frequency (ver Sparck-Jones, 1972)

vantes.

Ferragina *et al* (2005), também utilizam o cálculo do TF.IDF para determinar quais os termos de maior interesse dos documentos, mas complementam o trabalho com o conhecimento de duas *Knowledge Bases*, uma abordagem mista portanto. As bases de conhecimento são construídas *offline* e são utilizadas em conjunto com o TF.IDF no processo de atribuição de um *rank* aos termos. Uma, guarda os *anchor texts*[14] extraídos a partir de mais de 200 milhões de *Web Pages* e outra indexa o motor de busca DMOZ que classifica 3 500 000 sites em mais de 460 000 categorias. As duas bases de conhecimento são utilizadas por forma a enriquecer o reduzido texto vindo dos *snippets*[15], sobre os quais se baseia o processo de avaliação do grau de importância de um termo num documento. Os autores do trabalho assumem que considerar apenas os *snippets*, onde só parte do texto é utilizado. É uma solução potencialmente insegura em termos de qualidade e não querendo considerar todo o texto, tentam enriquecê-la acrescentando conhecimento prévio.

Depois de definido um primeiro *rank*, os termos com uma pontuação reduzida são rejeitados, utilizando um *threshold*. Os que ficam, juntam-se, formando *gapped sentences* de forma a criar *phrases*.

Este algoritmo poderá ser considerado como uma abordagem superior de entre todos aqueles que apenas consideram o cálculo do TF.IDF introduzindo a identificação de *phrases* entre documentos, para determinar quais os termos que os caracterizam. Isso é conseguido com recurso às bases de conhecimento, o que em contrapartida não deixa de ser um factor negativo, na medida em que o sistema não é auto-suficiente.

Técnicas de Web Content Mining

Outros trabalhos como Liu *et al* (2003), propõem numa primeira fase a identificação de tópicos relevantes e numa segunda fase a definição/descrição das páginas *Web* segundo

[14] Texto visível no *hyperlink*.

[15] Pequeno resumo do texto disponibilizado ao utilizador aquando da devolução de resultados.

estes tópicos.

Para aferirem acerca de conhecimento específico da *web* utilizam um conjunto de técnicas de *web mining*. Consideram palavras dentro de *tags HTML* relevantes (<*h1*> <*titles*>, etc...) e determinam o número de ocorrências de cada palavra. Sobre a totalidade do conjunto de palavras utilizadas são aplicadas regras de associação (ver Agrawal *et al*, 1996) com base num *threshold* definido pelo utilizador. Ao conjunto de termos relevantes encontrados pelas regras de associação, são aplicados métodos linguísticos (que tornam a aplicação dependente da língua inglesa) que possibilitem encontrar de entre a totalidade das páginas, definições para os termos relevantes previamente determinados. O *rank* das páginas *web*, é calculado com base no maior conjunto de definições que cada uma das páginas possui.

Comentários

É de salientar que em todos os trabalhos estudados, utilizam-se listas de *stopwords*[16], filtrando desta forma o conjunto de resultados, e algoritmos de *stemming*[17], aumentando o conjunto de termos a comparar.

Por outro lado, a maioria dos algoritmos tratam os documentos como um conjunto de palavras, ignorando informação importante próxima das mesmas. Zamir *et al* (1998), refere que a questão de considerar *phrases* ainda não foi praticamente aplicada na representação de documentos. Algoritmos que adoptem esse conceito, podem por isso ganhar vantagem utilizando essa informação adicional, aumentando o conhecimento sobre os documentos;

São conhecidas 3 abordagens principais para detectar *phrases*:

[16]Preposições, artigos e outras palavras que aparecem nos documentos e acrescentam pouco significado ou relevância, por aparecerem demasiadas vezes.

[17]Uma função de alguns motores de busca, que permite ao utilizador introduzir a palavra "dançando"e obter resultados da palavra "dança".

(1) *Syntactic phrases*: utilização de técnicas baseadas em métodos linguísticos. Uso de *parsing* sintático para encontrar palavras que cumpram determinada relação sintática (<JJ-NN NN> adjectivo substantivo substantivo; <NN PP NN> substantivo preposição substantivo). Neste caso é obrigatória a existência de conhecimento que registem os padrões linguísticos (ver Daille, 1996);

(2) *Statistical phrases*: uso de abordagens estatísticas tais como *contiguous non-stopped words* (ver Salton *et al* 1975), pares de palavras contínuas que co-ocorrem frequentemente (ver Fagan, 1987; Salem, 1987) ou n-gramas baseado em medidas de associação (ver Church & Hanks, 1990; Dunning, 1993; Smadja, 1993; Shimohata, 1997; Dias *et al*, 1999; Silva & Lopes, 1999; Tomokiyo & Hurst, 2003). Estas abordagens são completamente independentes de qualquer conhecimento prévio sobre a língua e sobre a estrutura do *corpus* a analisar, o que vai no sentido da nossa investigação;

(3) metodologias de *Machine Learning* que têm sido utilizadas ultimamente tentando usufruir de várias medidas caracterizadoras de *phrases* (ver Yang 2003; Diaz-Galiano *et al*, 2004; Ogata *et al*, 2004; Dias & Nunes, 2004).

2.4 Clustering de Páginas Web para Organização não Linear de Resultados

O objectivo do *clustering* é formar grupos diferentes uns dos outros, mas não forçosamente disjuntos, contendo membros muito semelhantes entre eles. Ao contrário do processo de classificação que segmenta informação associando-lhe grupos já definidos, o *clustering* é uma forma de segmentar informação em grupos não previamente definidos.

Na perspectiva de Sanderson *et al* (1999), a descrição de *clusters* enquadra-se dentro de 2 categorias:

(1) *monothetic clusters*: um único *label*. Esta abordagem garante que o(s) docu-

mento(s) existente(s) no *cluster* está(ão) directamente relacionado(s) com ele;

(2) *polythetic cluster*: neste caso a garantia anterior não pode ser dada, uma vez que existe mais do que um *label* a definir o *cluster*.

O *clustering* de documentos que Ferragina *et al* (2005) considera o *PageRank* do futuro, é apenas parte de um problema mais geral da análise de *clusters* e refere-se ao agrupamento de uma colecção de documentos baseada na sua similaridade.

2.4.1 Clustering de Documentos

A similaridade é avaliada ao nível dos tópicos relacionados que os documentos partilham entre si. Esses tópicos são calculados ao nível do conteúdo e são os termos que caracterizam os documentos. Com base nesses tópicos comuns aos documentos e nas medidas particulares a cada algoritmo, agrupam-se os mesmos num *cluster*. Quando os algoritmos disponibilizam uma estrutura hierárquica, estes *clusters* são as folhas da árvore e o nível superior da mesma será estabelecido pela similiraridade existente entre as folhas, condicionada uma vez mais pelo algoritmo a utilizar.

Não obstante a escolha da medida de similaridade entre os documentos (para avaliar a distância entre eles), o processo de *clustering* faz-se com base em 2 abordagens:

(1) algoritmos tradicionais (não hierárquicos/*partitioning* e hierárquicos);

(2) algoritmos não-tradicionais.

Com recurso a técnicas tradicionais (ver Zain *et al*, 1999), tanto os algoritmos não hierárquicos, como os algoritmos hierárquicos, que compreendem 2 abordagens (*Agglomerative* e *Divise*, ver figura 2.6) baseiam-se numa medida de similaridade entre documentos, normalmente uma distância reduzida a um único número.

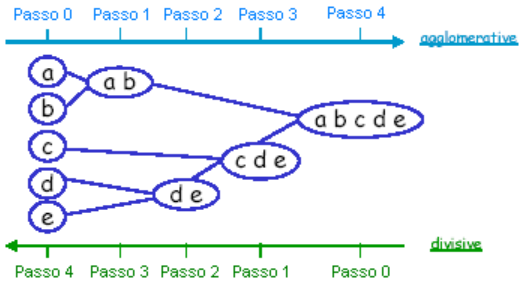

Figura 2.6: Duas abordagens de clustering hierárquico: *agglomerative e divise*.

Com a utilização destes algoritmos, a representação dos documentos com base num vector, exige a escolha prévia de k números de palavras da totalidade dos documentos. Cada documento é depois representado por um vector de tamanho k, com cada palavra da colecção sendo um atributo do mesmo. O *clustering* dos documentos é feito a partir da similaridade dos seus conteúdos, tipicamente calculada com base no TD.IDF (mas como vimos na secção anterior com cada vez mais indicadores além do TF.IDF, utilizando métodos de *Web Content Mining*), e medida com recurso à distância do tipo *Cosine*, Euclidiana ou *Jaccard* (ver Manning & Shutze, 1999).

Métodos Tradicionais

Dos trabalhos estudados no âmbito da utilização de técnicas tradicionais de *clustering*, Hearst *et al* (1996), usam um algoritmo não-hierárquico chamado *Fractionation* que agrupa **n** documentos em **k** *clusters* (ambas as variáveis definidas previamente), usando a distância *Cosine* para avaliar a similaridade entre os mesmos. O sistema que Hearst *et al* (1996), designaram como **Scatter/Gather**, desenvolvido no âmbito de colecções de texto e nunca testado em sistemas de *IR (Information Retrieval)*, agrupa documentos e apresenta-os aos utilizadores, que escolhem os que lhes interessam. Os documentos são colocados juntos e agrupados novamente por forma a produzirem um novo conjunto de *clusters*, apresentando apenas o que o utilizador seleccionou.

É obvio que os algoritmos hierárquicos destacam-se dos não-hierárquicos quando avaliados no âmbito de sistemas de IR, na medida em que podem resultar numa estrutura hierárquica (por produzirem várias partições da informação), facilitando a navegação do utilizador por entre os resultados. Leouski *et al* (1996), assumem como conteúdo os títulos e os 50 termos que aparecem no texto com maior frequência (assumindo-se no entanto a eliminação dos termos que ultrapassam um determinado *threshold* superior a 50%). Com base neste conteúdo utilizam um método hierárquico aglomerativo que resulta numa contrução em árvore, uma estratégia *bottom-up* que agrupa documentos mais pequenos em maiores.

Estas técnicas tradicionais, amplamente exploradas, revelam-se no entanto inapropriadas no âmbito da IR. Ignoram-se dados importantes que estejam próximos das palavras com consequência ao nível da perca de informação. De facto, têm a dificuldade, como referem Zeng *et al* (2004) e Fung *et al* (2003), em gerar *clusters* com nomes legíveis e são também demasiado dependentes de parâmetros de entrada e saída (definição prévia de determinados critérios para o funcionamento dos algoritmos). Como veremos mais tarde, estas críticas não são de todo verdadeiras. De facto, aplicaremos um algoritmo tradicional de *soft clustering* que permite resolver todos os problemas acima mencionados.

Métodos não Tradicionais

Da constatação deste problema, surge uma nova dinâmica associada à investigação de técnicas de *clustering* no âmbito dos sistemas de IR, que produzam melhores resultados. A partir de 1998 (ver Zamir et al, 1998), decorrente deste novo rumo na área, começam a surgir trabalhos que adoptam técnicas não-tradicionais na área do *on-line clustering* (agrupamento de resultados em *real-time* devolvidos por uma pesquisa), cada qual com as suas particularidades, inovações, diferentes abordagens e interrogações. Até hoje, no entanto, o *clustering* como alternativa à organização de resultados ainda não foi implementado na

maioria dos motores de busca, sendo o sistema proprietário *Vivissimo*[18] (galardoado pela SearchEngineWatch.com[19] como o melhor *Meta-Search Engine* no período 2001-2003) uma das poucas excepções conhecidas (ver Figura 2.7), do qual não existem no entanto publicações disponíveis.

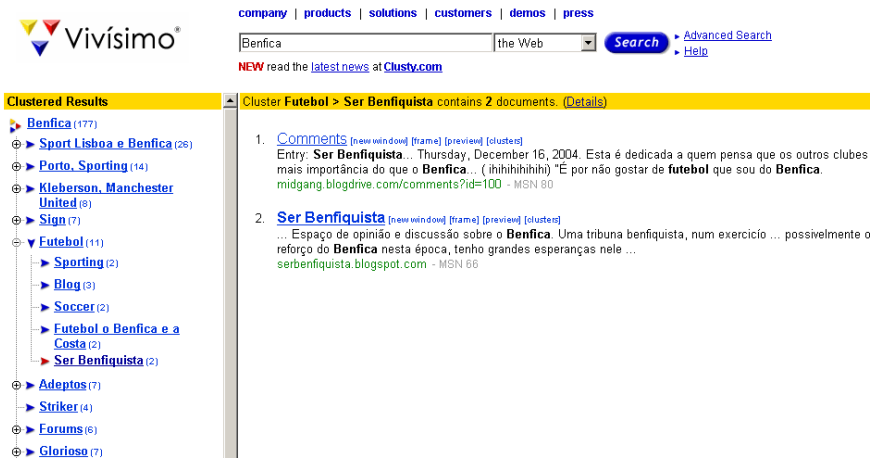

Figura 2.7: Resultados da pesquisa benfica no motor de busca Vivissimo.

Os trabalhos estudados no âmbito de técnicas não tradicionais, têm em comum o facto de partilharem o mesmo conceito: usam como base para o cálculo da similaridade dos documentos, apenas o título e os *snippets* de cada resultado, ou seja, o *abstract* e não o documento[20] inteiro. Na perspectiva de cada um deles, esta informação é informativa o bastante para poder gerar um número de *clusters* suficiente com designações legíveis.

Jiang *et al* (2002), que também fazem uso desta abordagem, referem no entanto que os resultados são obviamente inferiores quando comparados com o uso de todo o texto de um documento e justificam a adopção desta abordagem, que sacrifica em muito a qualidade dos resultados, com a necessidade de os produzir em tempo reduzido.

[18] http://www.vivissimo.com

[19] http://searchenginewatch.com

[20] Nesta secção, por forma a simplificar a leitura, utilizaremos o termo documento para nos referirmos tantos aos *snippets* como aos documentos inteiros

Cada um dos algoritmos, dada uma *query* e a lista de resultados devolvida por um qualquer motor de busca, analisa o conteúdo HTML e extrai a lista de títulos e *snippets*, procedendo ao cálculo do conjunto de palavras caracterizadoras dos documentos. Com base neste conjunto de palavras podem-se utilizar 2 abordagens de clustering:

(1) *flat clustering* (nível único);

(2) *hierarchical clustering* (estrutura hierárquica).

A este propósito leia-se a seguinte descrição dos seguintes trabalhos que utilizam *flat clustering*.

Nos trabalhos de Zeng *et al* (2004) e Zamir *et al* (1998), essas palavras são os n-gramas partilhados por mais do que um documento, assumindo-se cada uma como candidato a nome de *cluster*. Estes trabalhos, limitam-se a atribuir um *score* aos nomes de *clusters* candidatos (*phrases* partilhadas em mais do que um documento), através da multiplicação entre o número de documentos que contêm a *phrase* pelo número de palavras dessa mesma *phrase*.

Os *clusters* finais advêm da junção entre os nomes de *clusters* candidatos (ver figura 2.8), com base no número de documentos que partilham. A tentativa através desta forma é a de eliminar possíveis *clusters* duplicados, ou bastante similares.

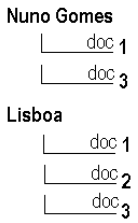

Figura 2.8: *Clusters* candidatos.

Se a parte comum aos 2 *clusters* exceder um certo *threshold* (acima dos 50 ou 75%, dependendo dos trabalhos), os *clusters* candidatos são juntos (ver figura 2.9), com a adap-

tação dos respectivos nomes.

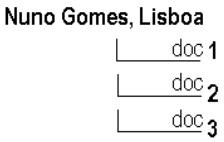

Figura 2.9: *Clustering* final: junção de *clusters* candidatos.

O trabalho de Jiang *et al* (2002), que utilizam um algoritmo *Robust Fuzzy*, baseia-se no uso de um n-grama caracter a caracter, com o valor da similaridade entre 2 *snippets* dado pelo coeficiente *Dice*:

$$Dice = \frac{2 * C}{A + B} \tag{2.1}$$

onde A e B são os números de n-gramas nos respectivos *snippets* e C o número de n-gramas partilhado por ambos. Estes comparam o seu trabalho ao trabalho de Zamir *et al* (1998), e referem como vantagem o facto de através do algoritmo de *clustering* usado, detectarem a formação de *overlapping clusters* sem a necessidade de utilizarem um *threshold*.

A utilização de um *threshold* que os diferentes trabalhos utilizam em diferentes fases do processo, é vista como uma decisão arbitrária que tanto desconsidera valores de 0,5 como considera valores de 0,51. Por via deste facto, Jiang *et al* (2002), procuram utilizar um algoritmo de *clustering* que lhe garanta a não utilização de um *threshold*, para agrupar documentos em clusters distintos.

Todos os algoritmos mencionados até agora são algoritmos de *flat clustering*, isto é, não existe uma estrutura hierárquica de apresentação dos documentos. Os próximos, reflectem metedologias de *hierarchical clustering*.

No contexto português, o TUMBA[21] (ver figura 2.10), é um motor de busca direccionado para uma comunidade específica: captura sites com domínio .pt ou escritos em português alojados noutros domínios (excepto .br) e com um *incoming link* de um domínio .pt.

Figura 2.10: Resultados da pesquisa benfica no motor de busca Tumba.

Esta implementação descrita por Martins *et al* (2003), baseia-se em dois trabalhos. Em primeiro lugar, utiliza a metodologia de Zamir *et al* (1998), para formar nomes de *clusters*. Em segundo lugar, para enriquecer a estrutura dos nomes de *clusters* propõe combinar com este trabalho, a metodologia de hierarquização proposta por Sanderson *et al* (1999), com base numa medida de co-ocorrência, denominada de *subsumption* (ver figura 2.11), que mede o grau de associação entre os termos.

[21] http://www.tumba.pt

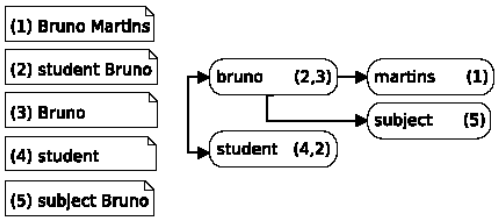

Figura 2.11: Exemplo de uma hierárquia *subsumption*.

Uma vez considerados todos os termos relevantes, cada termo é comparado com todos os outros. Se os documentos, nos quais um dado termo y ocorre, são um subconjunto dos documentos, nos quais um dado termo x ocorre, assume-se que x é hierarquicamente superior a y. Esta co-ocorrência, que no fundo avalia a especificidade ou generalidade dos termos, baseia-se no cálculo do DF[22] para cada termo, sendo que em quantos mais documentos um termo ocorre, mais geral ele é considerado (x *subsumes* y, relação hierárquica e x é mais frequente, logo é mais geral).

Fung *et al* (2003), propõem um algoritmo designado por FIHC - *Frequent Itemset-based Hierarchical Clustering*. Um *Frequent Itemset* é um conjunto de palavras que ocorrem num conjunto de documentos. Assim, um *Frequent Itemset* descreve algo comum aos documentos, podendo agrupá-los em *clusters* e posteriormente organizá-los segundo uma hierarquia. Seguindo o conceito já descrito por outros trabalhos, e ainda que com recurso a diferentes implementações, a formação dos *clusters* iniciais advêm da junção de documentos que partilham, não uma, mas várias palavras distintas. O *label* do *cluster*, que é *polythetic*, é formado por esse conjunto de palavras. Um documento não pode no entanto pertencer a mais do que um *cluster*, assim, depois de estarem formados os *clusters* iniciais, procede-se para cada documento ao cálculo de qual o melhor *cluster* onde o documento se pode integrar. O melhor *cluster*, será aquele em que existirem mais documentos, com termos iguais ao do documento a integrar.

[22]Document Frequency

O nível hierárquico é construído com base na similaridade existente entre os *clusters*. O tópico do *parente cluster* terá de ser mais geral que o tópico do *child cluster* para se poder contruir a árvore. Assim, os potenciais *parent clusters*, serão aqueles cujo *label* seja um subconjunto do *label* do *child cluster*. O próximo passo é igual ao efectuado para o *flat clustering*: seleccionar o melhor *parent cluster*, de entre todos os potenciais *parent clusters*.

Martins *et al* (2003), como analisado anteriormente, também propõem uma estrutura hierárquica, mas inferior à de Fung *et al* (2003), na medida em que neste último, a junção de *clusters* (que está na base da organização hierárquica) é feita comparando os conteúdos (termos) dos documentos já existentes no *cluster* e conteúdos (termos) dos documentos a integrar.

O trabalho de Zhang *et al* (2001), é no que diz respeito ao agrupamento, similar ao trabalho de Fung *et al* (2003), na medida em que é analisada a associação entre termos e documentos (ver figura 2.12), inserindo-se um documento num *cluster* que tenha termos iguais ao do documento a inserir. Considera-se assim que 2 documentos que partilhem grande parte dos termos estão relacionados entre si, logo devem pertencer ao mesmo *cluster*.

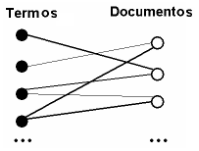

Figura 2.12: Associação entre termos e documentos

A organização hierárquica neste trabalho é feita com base num *threshold*. A validação é feita para cada par de *clusters*: conforme o limiar seja ou não ultrapassado, os *clusters* X e Y podem ser juntos ou então assume-se uma relação *parent-child*.

De forma a evitar a redução de qualidade por via do uso de *snippets*, Ferragina *et*

al (2005), sugerem como referido anteriormente, a utilização de bases de conhecimento que complementem o processo. Do processo de *rank* a cada um das *gapped sentences* extraídas dos *snippets* e das bases de conhecimento, saem os *labels* do *cluster*. *Snippets* que partilham as mesmas *gapped sentences* falam de um mesmo tema e portanto devem ser agrupados juntos num *cluster*.

Para aglomerar as folhas da árvore com vista a uma construção hierárquica, enriquece-se o *cluster* anteriormente construído com um *label* mais geral, avaliando essa generalidade na medida em que uma dada *gapped sentence* ocorra em 80% dos *snippets* do *cluster* (ver figura 2.13 extraída do site *http://snaket.di.unipi.it*).

Figura 2.13: Resultados da pesquisa benfica no motor de busca Snaket.

Assim o *label* do *cluster* assume uma descrição mais geral seguido de uma descrição específica (separada por um caracter). Quando os *labels* dos vários *clusters* partilharem uma *gapped sentence* estará encontrado o pai desses *clusters*, em termos hierárquicos.

Depois de ter estudado o estado da arte no domínio da nossa investigação, propomos no próximo capítulo detalhar a nossa contribuição na área.

Capítulo 3

Contribuição

3.1 Selecção de Páginas Relevantes

Os algoritmos da literatura estudada, tratam todos os documentos como sendo iguais, mas eles não são: todos têm diferente relevância para com a *query*, que diminui à medida que mais documentos são devolvidos. Produzir *clusters* em muitos documentos de pouca relevância pode reduzir a qualidade dos resultados. Excluímos por isso à partida alguns resultados (ver Secção 4.2), sacrificando o *recall* em detrimento da precisão.

Ao contrário da diferente literatura estudada que apenas considera os *snippets*, a nossa aplicação desconsidera todas aquelas páginas que apesar de devolvidas pelo sistema, contenham um reduzido número de palavras. Não considerar estas páginas, justifica-se pelo pouco conhecimento que as mesmas acrescentam ao sistema e consequente degradação da qualidade dos resultados.

Por outro lado, acrescentamos conhecimento ao nosso sistema, ao considerar um conjunto de páginas não devolvidas pelo mesmo, mas relacionados com a *query*. Utilizamos uma funcionalidade dos motores de busca, que para os endereços absolutos devolvem as N melhores páginas do site de acordo com a *query*.

3.2 Web Content Mining e Representação de Documentos

No contexto de *Web Content Mining*, *Mining the web process* pode ser visto como um sub-problema de *retrieval process*, mas o intuito deste trabalho é que eles se complementem. A utilização do conhecimento entendendo factos que há primeira vista e explicitamente nada representam, extraídos do repositório de informação imensa que a *Web* suporta, deverá complementar o *retrieval process* e melhorar a pesquisa de informação no seu todo.

O conceito que propomos é no seu objectivo final semelhante aos trabalhos referidos no capítulo anterior. No entanto a solução que apresentamos difere em muito, do proposto pela diferente literatura.

3.2.1 Representação dos Documentos

O facto de muitos trabalhos assumirem como potenciais *clusters* os termos partilhados por mais do que uma página, sem avaliação ou interpretação do conteúdo semântico do texto, faz com que os trabalhos de Zamir *et al* (1998), Martins *et al* (2003), Jiang *et al* (2002) e Fung *et al* (2003), fiquem sujeitos a definir como *label* do *cluster* um nome que apesar de aparecer nos correspondentes documentos, ou melhor no seus *snippets*, possa não ser um tópico indicativo dos mesmos, com consequências também ao nível do *clustering* nos mesmos trabalhos.

A este nível (*on-line clustering*) Zeng *et al* (2004) e Zhang *et al* (2001), fazem um pouco melhor, agrupando documentos que partilhem um tópico a nível semântico o que é aferido com recurso a 5 e 3 medidas respectivamente. Convém no entanto referir que a fase prévia é dada pela determinação de quais as palavras partilhadas entre os documentos. Nesse ponto a única medida que se regista é o TF.IDF não havendo lugar ao uso de mais nenhuma medida que permita aferir o conteúdo do documento, pois tal como referido

anteriormente, as restantes medidas são apenas aplicadas para escolher de entre o reduzido leque de palavras entretanto determinadas, as candidatas, as melhores palavras entre as melhores.

Ferragina *et al* (2005), por seu lado complementa o uso do TF.IDF com recurso à utilização de duas *Knowledge Bases*, tentando desta forma oferecer maior consistência ao sistema, enriquecendo os seus *snippets*. A aplicação torna-se por isso dependente da existência destas bases de conhecimento que alimentam o seu processo, ao contrário da nossa solução que é auto-suficiente. Também a determinação de *phrases* é dependente da existência de conhecimento linguístico prévio.

De nenhum destes trabalhos se pode por isso dizer que sejam trabalhos puros de *Web Content Mining*, quando a única medida de aferição do conteúdo é o TF.IDF e as bases de conhecimento no caso deste último.

O facto da nossa solução considerar o documento todo ao invés de considerar (como na maioria dos trabalhos estudados) apenas os títulos e os *snippets* (pequenos e pobremente formatados), torna-a uma solução mais abrangente pelo número e essencialmente pela extensão dos documentos no qual se baseia. Apenas Liu *et al* (2003), apresentam um trabalho em que todo o texto das páginas *web* é analisado, mas a este nível as técnicas de *Web Mining*[1] aplicadas são demasiado simplistas. Por outro lado, este mesmo trabalho, reduz o seu contexto a uma definição dos tópicos encontrados, com base em técnicas linguísticas que o tornam dependente das mesmas e da língua em que são definidas.

Para inferir conhecimento, a nossa aplicação utiliza um conjunto de árvores de decisão. Estas classificam um determinado exemplo distribuindo os atributos numa árvore, começando pela raiz até às folhas. Cada nó na árvore de decisão é um teste e uma instância é classificada, começando na raiz da árvore, testando o atributo nesse nó em específico e movendo-se para baixo no ramo especificado pelo atributo seguinte. Cada caminho na

[1] Análise das *tags* HTML.

árvore de decisão corresponde a uma conjunção de testes, geridas por um conjunto de regras *if-then*.

Esta árvore de decisão foi implementada no software WebSpy (ver Veiga *et al*, 2004) no qual cada palavra é definida por um conjunto de 12 atributos.

Assim, o conteúdo semântico do conjunto de textos devolvidos em resposta ao tema pesquisado, pode ser encontrado utilizando 12 propriedades:

(1) **IDF (Inverse Document Frequency)**: o IDF (ver Sparck-Jones, 1972) calcula a dispersão da palavra dentro de um conjunto de textos. Quanto mais dispersa menos importante essa palavra será relevante para um determinado texto;

(2) **TF (Term Frequency)**: frequência média de cada palavra na colecção de textos;

(3) **PP (Primeira Posição)**: é a média calculada de entre todos os documentos onde uma palavra X aparece, através da contagem do número de termos desde o início do documento, até ao primeiro aparecimento da palavra X;

(4) **Tamanho**: número de caracteres de uma determinada palavra;

(5) **Maiúsculas**: representa o número de caracteres maiúsculos existente numa determinada palavra;

(6) **MDM (Média da Distância Média)**: média de entre todos os documentos onde a palavra X aparece de um subatributo chamado DM (Distância Média), em que DM é a distância existente entre as várias ocorrências da palavra X;

(7) **MMD (Média da Menor Distância)**: média de entre todos os documentos onde a palavra X aparece, das menores distâncias entre a palavra X e o tema pesquisado Z;

(8) **SuperSTR**: número de termos que contêm a palavra X. Exemplo: Matemática Informática e Informática Ensino são 2 termos que contêm o termo Informática;

(9) **BigSuperSTR**: definido à custa do atributo anterior, o atributo BigSuperSTR é igual ao maior TF da totalidade dos termos inseridos no conjunto do SuperSTR;

(10) **SCP**: medida de co-ocorrência. O objectivo desta medida (ver Muller *et al*, 1997; Silva *et al*, 1999) é avaliar a força que interliga uma determinada palavra X com o tema

pesquisado Z;

(11) **Tipo**: resultado de uma medidade de similaridade (MESim) entre os documentos recuperados, que pretende agrupar os textos mais similares e definir o tipo de texto em que uma palavra aparece;

(12) **Grama**: representa o número de palavras que um dado termo X tem. Exemplo: Grama (Inteligência Artificial) = 2.

Em particular, as árvores de decisão demonstraram melhores resultados do que as redes neuronais, a regressão linear e a aprendizagem *bayesiana* (ver Veiga *et al*, 2004).

3.2.2 Normalização dos Textos

Extracção de Phrases

Os termos relevantes tanto podem ser palavras simples como palavras compostas, expressões, n-gramas ou *phrases*. Jiang *et al* (2002) e Fung *et al* (2003), não fazem uso de *phrases*, utilizando apenas palavras simples, com natural perca de informação.

O nosso sistema, utiliza o software SENTA (ver Dias *et al*, 2002) para detectar possíveis *phrases*. Os resultados apontam para a identificação de nomes e determinantes compostos, assim como locuções verbais, adjectivais, adverbiais, conjuntivas e proposicionais. Este sistema baseado exclusivamente em estatística, conjuga uma nova medida de associação, a Expectativa Mútua (ver Dias *et al*, 1999) e um novo processo de extracção baseado num algoritmo de máximos locais, o GenLocalMaxs (ver Silva *et al*, 1999).

A expectativa mútua avalia a coesão que existe entre as palavras de um n-grama e o algoritmo de máximos locais elege as unidades (candidatas a partir do conjunto de todos os n-gramas associados com as suas respectivas medidas de associação). O algoritmo GenLocalMaxs é além do mais flexível, na medida em que permite ser testado com todas as medidas de associação existentes e teoricamente bem fundamentado por não basear o processo de selecção, consoante o valor da medidade de associação, ultrapasse ou não,

um dado valor previamente estipulado.

As *phrases* são assim extraídas a partir de regularidades estatísticas, uma abordagem cuja principal vantagem é a flexibilidade, extraíndo todo o tipo de unidades polilexicais (referidas acima), a independência da língua e inexistência de conhecimento linguístico *a priori*. Em qualquer tipo de aplicação de *Information Retrieval*, a rapidez de resposta e de tratamento deve ser tido em conta. Por isso, incorporámos a versão do SENTA implementada por Gil & Dias (2003) que usa *suffix-arrays* e permite a extracção de unidades polilexicais em tempo real.

Stopwords e Stemming

As *stopwords* afectam a eficiência dos métodos baseados na frequência de termos (ver Martins *et al*, 2003). Por esta razão, todos os sistemas objecto do nosso estudo e acima referidos, removem-nas. Ao contrário da diferente literatura estudada, o nosso algoritmo não usa uma lista de *stopwords*. De facto, o software WebSpy elimina de uma forma automática todo o conjunto de palavras que apareçam várias vezes em diferentes documentos e que não revelam qualquer relevância para a representação dos mesmos. Assim, conseguimos desenvolver um sistema completamente flexível e não dependente da língua.

O *stemming* é o processo de reduzir uma palavra à sua forma básica. Reduz o espaço ocupado pelas palavras e é vantajoso no retorno de resultados na medida em que aumenta o espectro de palavras a comparar. Tem no entanto 2 problemas (ver Martins *et al*, 2003):

(1) pode reduzir duas palavras distintas à mesma forma básica;

(2) e torna o sistema que o utiliza, dependente da língua.

A não utilização no sistema que desenvolvemos de algoritmos de *stemming* ao contrário dos trabalhos estudados, é portanto um factor que juntamente com a ausência de lista de *stopwords*, permite manter a aplicação independente do domínio e da língua de qualquer um dos documentos que possam vir a ser analisados.

3.3 Clustering de Termos Relevantes para Apresentação Hierárquica dos Documentos

A organização de documentos facilita a navegação por entre o conjunto de resultados devolvidos pelo motor de busca. O processo de *clustering* não termina no entanto com o agrupamento e a criação de *clusters* coerentes. Como referem Zamir *et al* (1998), a actual procura pela lista de resultados não deve ser substituída por uma procura por entre os *clusters* e por isso o método deve permitir óptimas descrições dos mesmos. Não obstante esta consideração, Zamir *et al* (1998), assumem uma abordagem *polythetic cluster*, uma descrição com mais do que um *label*, da qual, só com algum trabalho o utilizador poderá deduzir/entender o tópico a que o *cluster* se refere.

No nosso trabalho utilizaremos também, cada um dos termos devolvidos como termos relevantes, como possíveis sub-tópicos ou *labels* do resultado do *clustering*, mas ao contrário da literatura acima referenciada (ver Zamir *et al*, 1998; Fung *et al*, 2003), não assumiremos a abordagem *polythetic cluster*, mas sim uma abordagem *monothetic cluster*. Por outro lado, a junção de *clusters* candidatos não será feita com base no número de documentos que partilham, como em Zeng *et al* (2004) e Zamir *et al* (1998), mas sim a partir da similaridade entre os termos relevantes que os documentos contêm. Como se pode observar pela figura 3.1, a abordagem de Zeng *et al* (2004) e Zamir *et al* (1998) corre o risco de juntar num só *cluster* documentos (caso do documento 2), que possivelmente pouco têm a ver com os restantes, como já referimos anteriormente.

Figura 3.1: *Clusters* candidatos.

Na perspeciva de Zeng *et al* (2004), Zamir *et al* (1998) e Jiang *et al* (2002), a definição de *clustering* é a apresentação (ver Figura 3.2), única e exclusiva de termos relevantes relacionados com documentos (*flat clustering*).

1.	**war** (32)	1. AlterNet: War on Iraq 2. War in Iraq - Christianity Today Magazine 3. End The War 4. Iraq Aftermath: The Human Face of War: AFSC 5. NOLA.com: War on Iraq	6.	**country** (19)	1. Library of Congress / Federal Research Division / Country ... 2. U.S. Department of State: Iraq Country Information 3. Iraq Country Analysis Brief 4. ArabBay.com: Arab Countries/Iraq 5. Countries: Iraq: Arabic Search Engine: Directory of arabic ...
2.	**middle east** (31)	1. Middle East Studies: Iraq 2. Amnesty International Report 2002 - Middle East and North ... 3. Human Rights Watch: Middle East and Northern Africa : ... 4. Columbus World Travel Guide - Middle East - Iraq - Overview 5. Iraq/Middle East	7.	**special report** (13)	1. Guardian Unlimited | Special reports | Special report: Iraq 2. Operation Iraqi Freedom - A White House Special Report 3. RFE/RL Iraq Report 4. Amnesty International Report 2002 - Middle East and North ... 5. Ethnologue report for Iraq
3.	**map** (18)	1. UT Library Online - Perry-Casta?eda Map Collection - Iraq ... 2. ABC Maps of Iraq; Flag, Map, Economy, Geography, ... 3. Flags of Iraq - geography; Flags, Map, Economy, Geography, ... 4. Lonely Planet - Iraq Map 5. Map of Iraq	8.	**guide** (13)	1. Lonely Planet World Guide | Destination Iraq | Introduction 2. Columbus World Travel Guide - Middle East - Iraq - Overview 3. Herald.com - Your Miami Everything Guide 4. Kansas.com - Your Kansas Everything Guide 5. Kansascity.com - Your Kansas City Everything Guide
4.	**saddam hussein** (13)	1. Iraq Resource Information Site - News History Culture People ... 2. U.S. Department of State - Saddam Hussein's Iraq 3. Iraq Crisis - Global Policy Forum - UN Security Council 4. New Scientist | Conflict in Iraq 5. Almuajaha - The Iraqi Witness: home	9.	**united nations** (11)	1. Mission of Iraq to the United Nations 2. united nations 3. United for Peace and Justice 4. U.S. Department of State: Iraq Country Information 5. Iraq Crisis - Global Policy Forum - UN Security Council
5.	**human rights** (11)	1. Human Rights Watch: Middle East and Northern Africa : Iraq ... 2. Iraq: Amnesty International's Human Rights Concerns 3. Human Rights Watch: Background on the Crisis in Iraq 4. Iraq:Amnesty International's Human Rights Concerns for 5. Iraq Aftermath: The Human Face of War: AFSC	10.	**travel, business** (16)	1. Iraq: Complete travel information to Iraq, travel facts, ... 2. EIN news - Iraq - Political, Business and Breaking ... 3. Iraq - Travel Warning 4. Columbus World Travel Guide - Middle East - Iraq - ... 5. Iraq Visa Application - Tourist Visas, Business Visas, ...

Figura 3.2: Resultados de *clustering* para a palavra Iraq. Trabalho de Zeng *et al*, (2004)

Já Martins *et al* (2003), propõem uma organização hierárquica. A mesma tem por base o *clustering* dos termos que co-ocorrem. Esta similaridade é no entanto avaliada tendo em atenção apenas o IDF e os termos são considerados co-ocorrentes com outros termos, mediante um dado *threshold*, o que implica a intervenção do utilizador (de todo a evitar) para ajustar parâmetros. Por outro lado e pelo facto de não existir qualquer avaliação semântica para a interpretação do seu significado no texto, o agrupamento de termos tem tendência a formar *clusters* potencialmente não relacionados entre si, como facilmente se pode depreender da análise da figura 3.3:

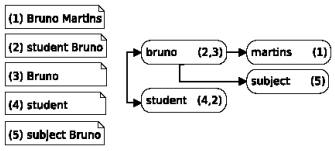

Figura 3.3: Exemplo de *subsumption* hierárquico.

Fung *et al* (2003), tal como Martins *et al* (2003), também propõem uma estrutura hierárquica, mas como visto anteriormente superior à deste último visto que o agrupamento de *clusters* (que está na base da organização hierárquica) dá lugar à comparação entre o conteúdo (termos) dos documentos e o conteúdo (termos) do documento a inserir. Não convém no entanto esquecer, aquele que é um ponto fulcral no que nos distingue destas abordagens: a não assumpção por parte da diferente literatura estudada, de nenhuma ou quase nenhuma técnica de *Web Content Mining*, razão pela qual se podem formar *clusters* potencialmente não relacionados entre si. O algoritmo de Fung *et al* (2003), não admite por outro lado que um documento pertença a mais do que um *cluster*.

Zhang *et al* (2001), têm uma abordagem bastante semelhante a Fung *et al* (2003), como visto anteriormente, mas utilizam comparativamente a este último um *threshold* para validar o agrupamento de *clusters* similares (que está na base da organização hierárquica).

Ferragina *et al* (2005), também utilizam um *threshold* no seu processo de construção hierárquico, onde um *parent* assume os *clusters* que partilhem entre eles (no *label*) uma *gapped sentence*. Tal abordagem pode resultar em agrupamentos de *clusters* potencialmente incorrectos, na medida em que não são usadas técnicas de *Web Content Mining* para avaliar se o conteúdo dos documentos é ou não similar.

De entre todos os trabalhos estudados, este é o único que oferece a possibilidade de efectuar *query expansion* também nos moldes por nós proposto: *Classified Query Expansion*.

A nossa perspectiva de *clustering* difere também de Zeng *et al* (2004) e Zamir *et al* (1998), onde a *phrase*, para se assumir como candidato a *cluster* terá de aparecer partilhada por mais do que um documento (ver Figura 3.4).

Figura 3.4: Candidatos a *clusters*.

No caso da figura 3.4, T1 e Eusébio não poderiam por isso formar *clusters*, ao contrário de Nuno Gomes que aparece partilhado por 2 documentos. De facto o *clustering* da nossa aplicação ultrapassa a limitação acima referida, ao ser feito com base num conjunto de termos relevantes que tanto podem ser relevantes para um só documento como para vários.

Difere também de Zeng *et al* (2004), Zamir *et al* (1998) e Zhang *et al* (2001), na medida em que não fazemos um *merge* de *clusters* mediante um dado *threshold*, mas sim de forma completamente automática, utilizando o algotimo de *Soft Clustering*, Poboc, desenvolvido por Cleuziou *et al*, (2004).

De todos os trabalhos, apenas Martins *et al* (2003), Fung *et al* (2003), Zhang *et al* (2001) e Ferragina *et al* (2005), implementam uma organização hierárquica dos resultados. A nossa implementação também sugere uma estrutura hierárquica para apresentação dos resultados e dá um passo em frente na sumarização dos mesmos, aplicando o algoritmo de *soft clustering* Poboc (mais detalhes no capítulo 5) sobre o conjunto de termos relevantes associados a cada documento, com a vantagem de agrupar num só *cluster*, termos directamente relacionados com o mesmo assunto, possibilitando-se desta forma a distinção entre tópicos diferentes (disambiguidade dos termos) e a organização/sumarização hierárquica aos olhos do utilizador (ver Figura 3.5).

Figura 3.5: Sumarização dos *clusters*.

A vantagem das técnicas de *soft clustering* sobre as técnicas de *hard clustering* baseia-se no facto de um determinado texto poder encontrar-se em vários *clusters*. Esta particularidade é um pré-requisito evidente, sabendo que um texto pode abordar vários temas ao mesmo tempo e assim pertencer a diferentes *clusters*.

3.4 Resumo do Trabalho Relacionado e Contribuição

A comunidade de IR, através da literatura científica publicada, sugere diferentes soluções para o problema da organização de resultados, mas todos os trabalhos têm em comum o facto de apenas considerarem os títulos e os *snippets* de cada um dos resultados devolvidos da execução de uma pesquisa.

Ferragina *et al* (2005), é o único que ainda assim tenta enriquecer os *snippets* com o conhecimento de duas bases de conhecimento.

A representação dos documentos é feita com recurso ao modelo de Espaços Vectoriais e com implementação usando *suffix-arrays* quando a medida de co-ocorrência mede a importância dos termos ou *suffix-trees* quando os termos advêm simplesmente do conjunto de palavras partilhadas por mais do que um documento.

Os algoritmos de *clustering* distinguem-se pela combinação dos dois níveis seguintes:

(1) os que consideram como termos caracterizadores dos documentos palavras simples e os que consideram expressões (palavras simples; expressões);

(2) os que apenas fazem *flat clustering* e os que fazem *clustering* hierárquico.

Da literatura estudada, Hearst *et al* (1996), ainda que não tenha sido testado em ambiente *Web* e Jiang *et al* (2002), proporcionam *flat clustering* com palavras simples, enquanto Zamir *et al* (1998) e Zeng *et al* (2004), o fazem com expressões/*phrases*.

Zhang *et al* (2001), é o primeiro a introduzir o *hierarchical clustering* com expressões, ao que se segue Martins *et al* (2003) e Ferragina *et al* (2005).

Fung *et al* (2003), sugere também uma abordagem de *hierarchical clustering* mas com palavras simples.

De todos estes trabalhos apenas Zeng *et al* (2004) e Zhang *et al* (2001), utilizam algumas propriedades no âmbito do *Web Content Mining*, mas como referido anteriormente, não para definir termos relevantes caracterizadores dos documentos, mas sim para extraír *phrases*. Apenas Liu *et al* (2003), tem uma abordagem pura de *Web Content Mining* ainda que demasiado simplista e fora do âmbito de *Web Clustering*.

A utilização de algoritmos de *clustering* que não usam a informação constante dos documentos, pode até produzir *clusters* entendíveis para o utilizador, mas o mais natural é que não correspondam aos seus desejos. O *label* dos *clusters* é normalmente escolhido por uma medida teórica e até pode ser suficientemente descritivo se os documentos pertencentes aos *clusters* forem relacionados entre si. Esta relação é no entanto subjectiva de avaliar, a não ser que se utilizem medidas que permitam entender o conteúdo dos documentos. Se tal não acontecer, os algoritmos podem estar a formar *clusters* com documentos não relacionados entre si.

Esta tese tenta ultrapassar as limitações acima descritas. O nosso algoritmo não considera todos os URLs devolvidos pelo motor de busca em resposta à *query* de pesquisa. Aplica uma função que escolhe de entre os documentos devolvidos, os melhores. Ao contrário de toda a literatura estudada considera todo o texto dos documentos ao invés de

seleccionar apenas os *snippets* (sendo auto-suficiente na medida em que não alimenta o seu saber com recurso a bases de conhecimento) e não é dependente da língua. Por entre outras características, não utiliza nem lista de *stopwords*, nem algoritmos de *stemming*, tornando-o completamente flexível e aplicável a qualquer domínio, género ou língua.

De entre o nosso melhor conhecimento somos os primeiros a propor um entendimento dos documentos, usando para isso técnicas de *Web Content Mining*, utilizando esse conhecimento como base de formação dos *clusters*, hierarquicamente estruturados. A este nível apresentamos de uma forma distinta, tópicos diferentes (um passo no sentido de resolver o problema da ambiguidade dos termos).

No capítulo seguinte detalharemos os fundamentos da nossa arquitectura.

Capítulo 4

Representação dos Documentos

4.1 Arquitectura Global

A arquitectura global da aplicação designa-se por WISE (Web Interactive Search Engine) e é composta por 4 componentes principais:

(1) a selecção de páginas relevantes para uma dada *query*;

(2) a normalização dos documentos por integração de um módulo de extracção de palavras compostas, o SENTA (ver Dias, 2002);

(3) a identificação das palavras/*phrases* relevantes de cada documento utilizando o software WebSpy (ver Veiga *et al*, 2004);

(4) a apresentação hierárquica dos documentos utilizando o algoritmo de *Soft Clustering*, Poboc (ver Cleuziou *et al*, 2004).

Este trabalho tem uma parte técnica e uma parte teórica. Na parte técnica, integramos num processo de Engenharia de Software um conjunto de metodologias já existentes. Na parte teórica, foi pensada uma arquitectura totalmente flexível para o *clustering* hierárquico de páginas *Web*. Em particular, desenvolveu-se um novo método de extracção de páginas relevantes e uma representação dos documentos baseada em termos relevantes

para a aplicação de algoritmos de *clustering*.

O nosso algoritmo é assim composto pelos seguintes passos:

1. devolver a lista de resultados de acordo com a *query* e com um dado motor de busca;
2. seleccionar os resultados mais importantes;
3. normalizar as páginas seleccionadas e identificar os n-gramas/*phrases* presentes;
4. calcular os termos relevantes para cada documento;
5. agrupar hierárquicamente os resultados;
6. apresentar os resultados ao utilizador.

Utilizámos para tal técnicas de *Web Content Mining*, *Clustering* e Processamento da Linguagem Natural.

A Figura 4.1 ilustra a arquitectura do software:

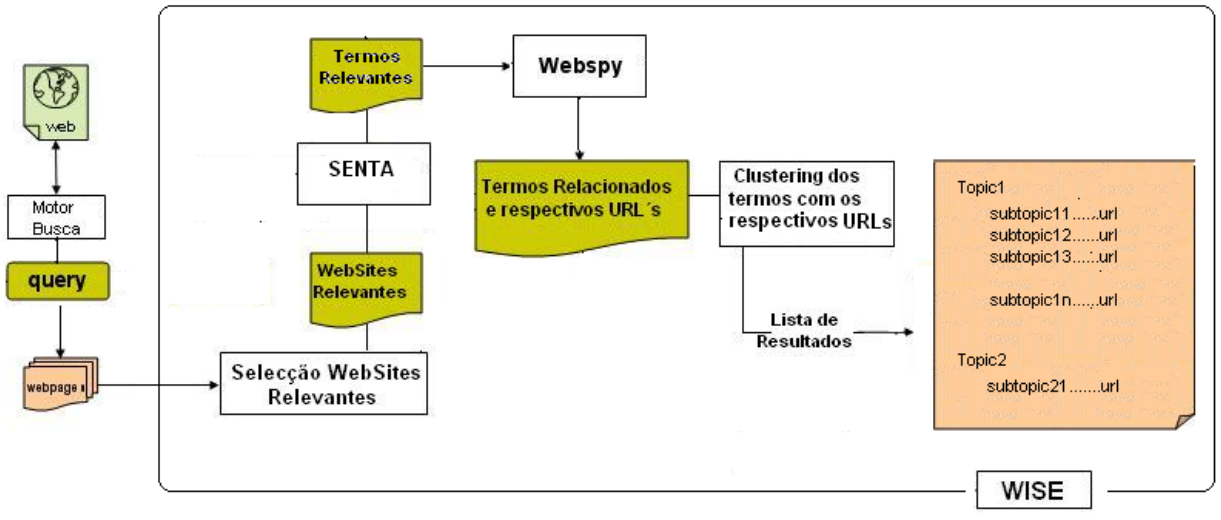

Figura 4.1: Arquitectura do software WISE.

A pesquisa de informação desenvolve-se como em qualquer *meta-crawler*. O utilizador deve especificar no sistema o tema a pesquisar (ver Figura 4.2), assim como o motor de busca pretendido.

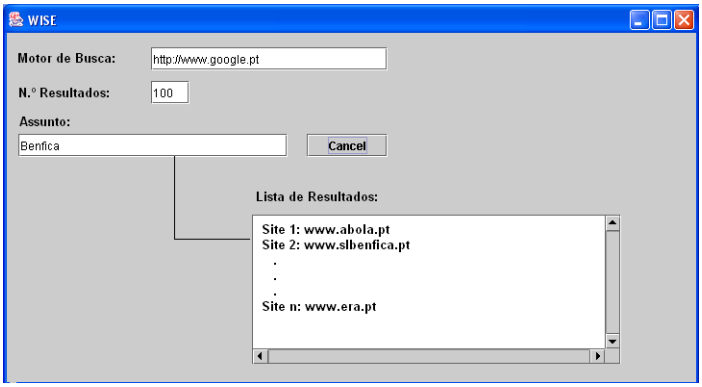

Figura 4.2: Especificação de um tema a procurar e respectiva lista de resultados.

O WISE interroga então um motor de busca ou um meta-motor de busca (ver Wegrzyn-Wolska, 2004) definido como parâmetro do sistema. Da lista dos documentos devolvidos, consideraremos como resultados apenas um número limitado de sites, os mais relevantes (ver secção 4.2).

Cada um dos *urls* considerados como mais relevantes, é no seguimento passado como parâmetro para o software SENTA (ver secção 4.3). Sobre estes, o SENTA devolverá um conjunto de expressões ou palavras compostas que substituírão as correspondentes palavras simples em cada um dos textos, referente ao *url*.

De seguida a utilização do software WebSpy (ver secção 4.4), devolverá do conjunto de textos obtidos, como resposta na etapa anterior (guardados num repositório de *Web Pages*), os termos relacionados (palavras ou expressões) com o tema a pesquisar, no exemplo seguinte com o tema Benfica (ver Figura 4.3).

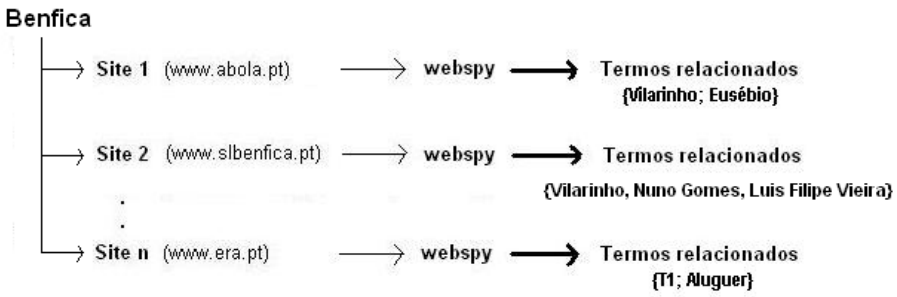

Figura 4.3: Utilização do software WebSpy na procura de termos relacionados.

A lista dos resultados, ou seja, dos termos relevantes provenientes do WebSpy com a referência às respectivas páginas onde os mesmos se encontram, é geralmente designada por *flat clustering* (ver Figura 4.4). Note-se pela observação da figura 4.3 e da figura 4.4, que permitimos a este nível o *overlap*[1] dos documentos. Assim, a informação relativa aos termos relacionados com o tema benfica, tomando como exemplo a Figura 4.4, seria a seguinte lista: (Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luís Filipe Vieira).

Termo	Site(s)
Vilarinho	http://www.abola.pt
	http://www.slbenfica.pt
Eusébio	http://www.abola.pt
T1	http://www.era.pt
Aluguer	http://www.era.pt
Nuno Gomes	http://www.slbenfica.pt
Luis Filipe Vieira	http://www.slbenfica.pt

Figura 4.4: *Flat Clustering*.

No entanto, esta lista da Figura 4.4 que poderia ser devolvida por um motor de busca num contexto de *Global Document Analysis* (ver Xu & Croft, 1996) não é a mais ade-

[1]Possibilidade de um documento pertencer a mais do que um *cluster*.

quada nem para fazer *Interactive Query Expansion* (o utilizador fica perdido no meio de informação díspar) nem para *Automatic Query Expansion* (que envolveria conceitos diferentes na pesquisa de informação) devido à ambiguidade do termo Benfica, que tanto pode referenciar o clube de futebol como o bairro de Lisboa. Assim, precisamos de efectuar a classificação dos termos de forma a juntar numa mesma classe os termos referentes a um mesmo conceito. Em termos práticos a lista dos termos relevantes Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luís Filipe Vieira e os respectivos sites onde os mesmos se encontram referenciados, servirá como *input* para este próximo passo.

O software WISE, utilizando de forma recursiva o WebSpy, determinará que termos estão relacionados com cada um dos elementos do conjunto Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luís Filipe Vieira (ver Figura 4.5).

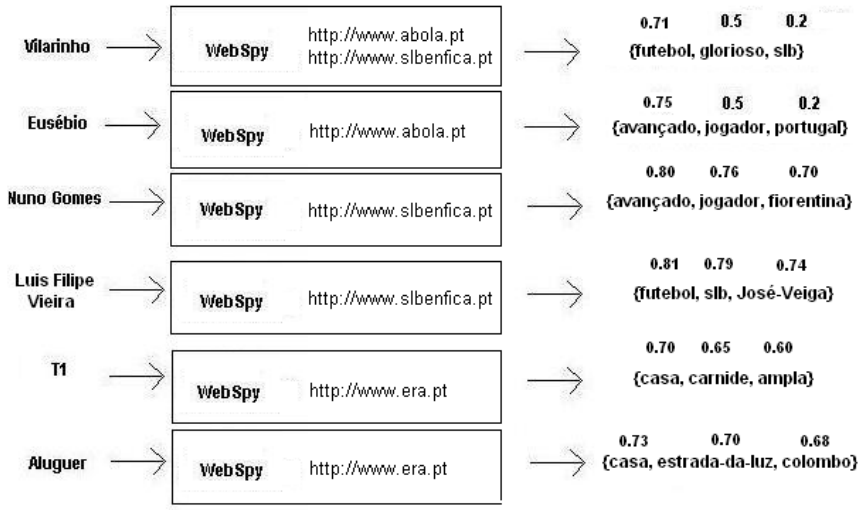

Figura 4.5: Lista dos termos relacionados com cada elemento do conjunto Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luis Filipe Vieira, com indicação da respectiva probabilidade de relevância devolvida pelo WebSpy (árvore de decisão).

Finalmente, com recurso ao algoritmo de *soft clustering* hierárquico Poboc (ver Cleuziou *et al*, 2004), o software WISE determinará grupos ou *clusters* de termos relacionados, podendo desta forma dissociar os diferentes conceitos existentes e permitir uma classificação

automática dos dados para uma melhor visualização em relação à *query* (ver Figura 4.6).

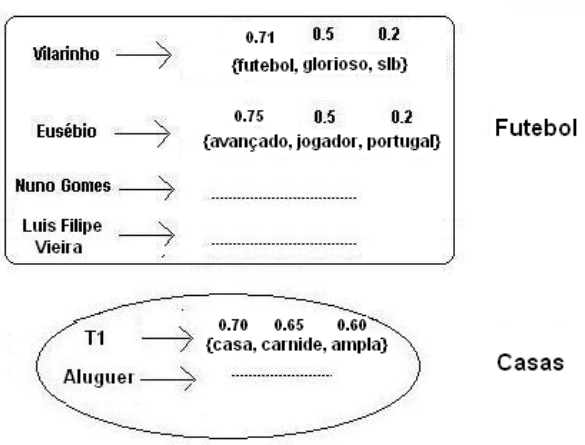

Figura 4.6: Classificação das palavras relacionadas em 2 *clusters*.

Observe-se a apresentação final dos resultados da parte do WISE na Figura seguinte:

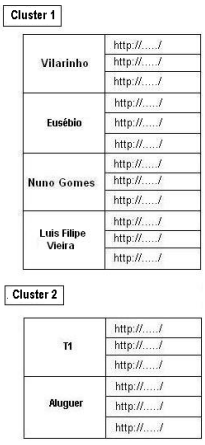

Figura 4.7: Apresentação dos resultados com os respectivos *URLs*.

Começamos por explicar o primeiro passo da nossa arquitectura.

4.2 Selecção de Páginas

Não obstante o bom funcionamento de alguns motores de busca, acreditamos ser possível aumentar a precisão destes sistemas na obtenção de páginas relevantes. Assumindo o descrito na secção relativa à forma como funcionam, em particular o funcionamento do Google, sentímos necessidade de introduzir uma fase de pre-processamento na selecção de páginas. De acordo com a lista de resultados devolvidos pelo motor de busca definido pelo utilizador no uso da aplicação WISE, procedemos ao cálculo da média de relevância global, dada pela equação 4.1:

$$\text{média de relevância} = \frac{\sum \text{URLs devolvidos}}{\sum \text{EndereçoAbsoluto de URLs diferentes}} \tag{4.1}$$

e seleccionamos como relevantes todos os *urls* cujo número de ocorrência for superior à média calculada e *urls* cujos endereços são absolutos. O número de ocorrências de um *url* X, é dado pela soma de todos os endereços absolutos devolvidos, iguais ao do *url* X.

Tomando como exemplo a Figura 4.8, o cálculo da
média de relevância = $\frac{4}{3}$ produz um resultado aproximado de 1,3.

Site	N.º Ocorrências
http://www.publico.pt	2
http://www.publico.pt/noticias/01.html	2
http://www.portoalegre.pt/campanha/1.html	1
http://www.fcporto.pt	1

Figura 4.8: Lista de Resultados relativos à *query* Porto.

Com base no número de ocorrências evitamos a selecção do *url* relativo a Porto Alegre, que apesar de devolvido pelo motor de busca não deve ser considerado relevante de acordo com a *query* definida, o que facilmente se entende. A verdade é que se o site ao

qual a página pertence, fosse de acordo com a *query* especificada, realmente relevante, mais do que uma ocorrência teria sido devolvida.

O nosso sistema desconsidera por outro lado todos os *urls* que devolvidos e seleccionados pelo sistema como relevantes, contenham um reduzido número de palavras. Não consideramos desta forma um conjunto de páginas que degradariam a qualidade dos resultados.

Esta fase de pre-processamento ao filtrar todo o conjunto de resultados devolvidos pelo motor de busca, seleccionando os melhores *urls* de entre os devolvidos, já de si os melhores, aumenta a precisão a nível de relevância, evitando a selecção de *urls* que apesar de devolvidos podem ser mais facilmente irrelevantes de acordo com o assunto especificado. Privilegiamos desta forma a precisão em detrimento do *recall*.

Nesse sentido, o sistema extrairá, com base no exemplo anterior, os seguintes *urls*:

- http://www.publico.pt
- http://www.publico.pt/noticias/01.html
- http://www.fcporto.pt

Depois, usando uma funcionalidade do motor de busca Google, para os *urls* com endereço absoluto, o sistema extraí os N melhores resultados (as N melhores páginas) do site, de acordo com a *query* previamente definida. Para isso, desenvolvemos um *spider*, que extraí o texto para cada uma dessas N páginas, não considerando novos *urls* encontrados (ver Figura 4.9).

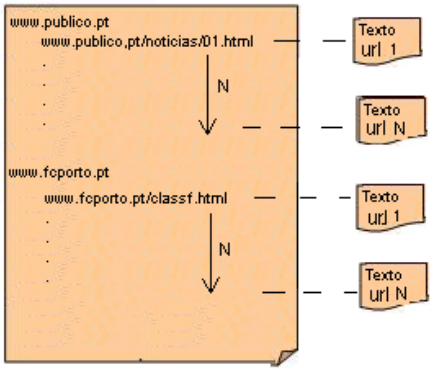

Figura 4.9: Obtenção do texto para cada um dos *urls* relativos.

Para os resultados devolvidos da execução da *query* com endereço relativo, seleccionamos apenas o texto dessa mesma página. De outra forma, se considerassemos outros *urls* aí encontrados estaríamos provavelmente a considerar páginas de todo irrelevantes, sem qualquer tipo de relação com o assunto especificado. Outra hipótese, seria a de considerar novamente a utilização da funcionalidade do Google anteriormente usada para os endereços absolutos. Mas imaginemos que a página proveniente da fase de pre-processamento fosse a seguinte: *http://geocities/hobbies/adeptoPorto*, aplicar aquela funcionalidade devolveria de entre o conhecido alojamento de páginas *geocities*, um conjunto de links que apesar de relacionados com o assunto, seriam de todo irrelevantes, caso contrário já teriam sido devolvidos pelo motor de busca.

4.3 Integração do SENTA

O software SENTA, desenvolvido por (ver Dias, 2002), permite obter nesta fase de pre-processamento, um conjunto de n-gramas ou palavras compostas a partir do texto retirado de cada um *urls* previamente seleccionados. Quando uma só palavra não é suficiente para expressar um conceito, recorre-se a grupos de palavras (duas ou mais). Palavras como Nuno Gomes, Partido Independente, Luis Filipe Vieira, que ocorrem frequentemente em

documentos são, utilizando o software, extraídas automaticamente, assumindo um significado próprio quando comparadas com a sua forma singular. A extracção de um conjunto de n-gramas/*phrases* de entre um *corpus*, é importante não só para a tradução automática (ver Dias *et al*, 1999) como também no processo de classificação e indexação de documentos, aquele que mais nos interessa. Como referido por Baeza-Yates *et al* (1999) e descrito na secção relativa aos motores de busca, o processo de classificação é uma tarefa semi-automática ou totalmente automatizada, mas com baixa precisão e cobertura e que possivelmente beneficiaria da identificação de palavras compostas.

A utilização do software SENTA, que faz a utilização de métodos puramente estatísticos, baseado no número de vezes que cada n-grama ocorre no *corpus*, utiliza o algoritmo *GenLocalMaxs* (ver Silva *et al*, 1999) e a medida de associação Expectativa Mútua (ver Dias *et al*, 1999), baseando-se na procura do máximo local da função de associação, função esta que mede a força da ligação existente entre os vários tokens de um n-grama. Sequências de tokens, contíguas, fortemente ligadas entre si, corresponderão a valores de EM elevados e serão escolhidos pelo algoritmo *GenLocalMaxs* como *phrase*[2]. Em particular, foi utilizada a implementação baseada em *suffix-array* proposta por Gil e Dias (2003) de forma a extraír os n-gramas/*phrases* em tempo real.

A maioria dos sistemas é ainda muito primitiva, classificando os documentos recorrendo a palavras-chave, e é por isso importante que a classificação dos documentos deixe de ser somente baseada em palavras singulares. A utilização do software, que além do mais é independente em relação à língua, permite uma interpretação dos documentos (considerando termos na forma de *phrases* devidamente contextualizados) na medida em que é extraída informação semântica, conferindo ao sistema uma maior base de conhecimento.

[2]O software SENTA foi desenvolvido para extrair também, expressões não contíguas. Neste trabalho, no entanto, só foram utilizadas as palavras compostas contíguas, deixando a utilização das não contíguas para trabalho futuro.

4.4 WebSpy

A utilização do software SENTA conjugada com o software WebSpy, vem no seguimento da teoria de *Web Content Mining*. A interpretação dos documentos e uma orientação centrada na informação são conceitos que se encontram na base da investigação e desenvolvimento de sistemas de IR com maior performance.

O software WebSpy (ver Veiga *et al*, 2004) foi melhorado no contexto da aplicação WISE. Com a integração do software SENTA, deixou de existir a distinção entre palavras e n-gramas e recebe, agora, um conjunto de documentos (representados por *phrases* e ou palavras) provenientes da execução do software SENTA.

Em termos de implementação, como se pode observar do exemplo (ver Figura 4.10), o software WebSpy não recebe os textos provenientes do SENTA, um a um, mas sim o conjunto de todos os textos interligados por *hyperlinks* que digam respeito ao mesmo *host* (3 textos para o *host* www.slbenfica.pt, 2 textos para o *host* www.ojogo.pt e 1 texto para o *host* www.abola.pt).

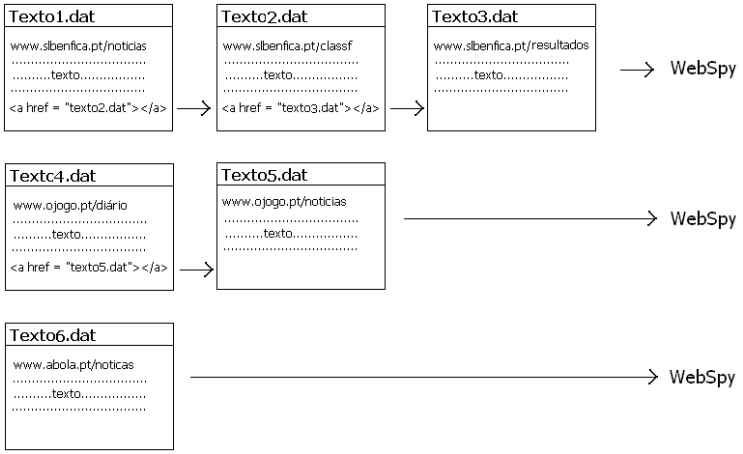

Figura 4.10: Textos provenientes do SENTA.

Tal permite, o *mining* de *phrases* que ocorrem em mais do que um texto e que por isso assumem maior relevância e a detecção de outras, a partir da árvore de decisão im-

plementada no WebSpy, que ocorrem demasiadas vezes em muitos textos e que por isso pouco acrecentam ao conhecimento que temos do documento.

Em particular, o WebSpy procura devolver para cada conjunto de textos, tendo em atenção o contexto em que ocorre, um conjunto de palavras/termos que com um dado nível de confiança (a média das probabilidades resultante da execução das várias árvores de decisão implementadas) se apresentam relacionados com o assunto especificado, num dado texto, texto este relativo a um endereço URL (ver Figura 4.11).

Assunto: Benfica

Termo	Nível de Confiança	URL
Nuno-Gomes	83,4	www.slbenfica.pt/classf www.slbenfica.pt/notícias
Eusébio	89	www.slbenfica.pt/resultados
.		
.		
Benfica-vs-FCP	84	www.ojogo.pt/diário
agressão-a-simão	77	www.ojogo.pt/notícas
.		
.		
José-Veiga	80	www.abola.pt/notícias
.		
.		

Figura 4.11: Termos relevantes provenientes do WebSpy.

Habitualmente retiradas com base numa lista, as *stopwords*, são neste software evitadas automaticamente (ver secção 3.2.2). Mantemos desta forma, a aplicação independente da língua e do domínio a que o texto se refere.

Da utilização da aplicação WISE, decorre que é possível proceder ao agrupamento de termos relacionados. Utilizaremos cada um dos termos devolvidos como relevantes como possíveis sub-tópicos ou *labels* do resultado do *clustering*.

Capítulo 5

Clustering de Páginas Web

5.1 Clustering

A classificação não supervisionada, característica dos métodos de *clustering*, permite uma adaptação em *real time* à *query* do utilizador, razão pela qual é utilizada no *on-line clustering*. Sofre, claro está, das dificuldades inerentes a esse processo às quais acresce, como referem Zeng *et al* (2004), o facto de não se poderem utilizar técnicas tradicionais de *clustering* no domínio da *web*.

Este tipo de dificuldades são no entanto ultrapassadas pela aplicação WISE. O algoritmo de *clustering* Poboc (ver Cleuziou *et al*, 2004), utilizado pela nossa aplicação, enquadra-se na abordagem tradicional de *clustering*, e não em medidas heurísticas como os métodos não-tradicionais. Não obstante esse facto, é possível ultrapassar as dificuldades referidas na secção relativa aos métodos tradicionais (2.4.1), conjungando-o com a aplicação WISE.

O agrupamento dos documentos é feito não com base na sua similaridade, mas com base na semelhança e similaridade do seu conteúdo, i.e., das suas palavras mais relevantes.

O *overlap* é garantido pela aplicação WISE e pelo algoritmo Poboc (ver Cleuziou *et al*, 2004), evitando desta forma confinar um documento a um único *cluster*, o que seria

de todo injustificável na medida em que um documento pode dizer respeito a mais do que um tópico.

Os *labels* dos clusters são designações legíveis, determinadas pela aplicação WISE logo após a determinação dos *clusters*.

Descrevemos de seguida todo o processo de *clustering* levado a efeito pela aplicação WISE.

5.2 Poboc

Os algoritmos de *clustering* que estudámos enquadram-se dentro de 2 perspectivas: os tradicionais e os não-tradicionais. Os primeiros tem a dificuldade em gerar nomes de *clusters* legíveis, a obrigatoriedade de definir previamente um conjunto de variáveis e de confinar um documento a um só *cluster*. Os segundos não assumem qualquer método de *clustering* conhecido, baseando-se em medidas heurísticas, mas produzindo nomes de *clusters* legíveis.

O algoritmo de *clustering* Poboc *(Pole-Based Overlapping Clustering)* desenvolvido por Cleuziou *et al* (2004), é um algoritmo de *soft clustering* que se enquadra na abordagem tradicional, mas que ultrapassa todas as suas dificuldades:

(1) o número de *clusters* é desconhecido à *priori*;

(2) e um objecto pode pertencer a mais do que um *cluster*.

5.2.1 Funcionamento

Dada uma matriz de similaridade e um conjunto de objectos, o Poboc constroi pequenos conjuntos de objectos (os pólos) e de seguida associa os objectos a esses pólos.

As suas 4 principais tarefas são:

(1) procura de pólos;

(2) construção de uma matriz associada dos objectos aos pólos;

(3) atribuição dos objectos a um ou mais pólos;

(4) organização hierárquica dos grupos obtidos.

5.2.2 Matriz de Similaridade

Um dos pontos que nos distingue da maioria dos trabalhos estudados é a implementação de uma estrutura hierárquica designada por *hierarchical clustering*. A partir do *flat clustering* estudado na secção 4.1, o sistema executa novamente o WebSpy para determinar o conjunto de termos relacionados com cada um dos elementos do *flat clustering* (tomando como exemplo a figura 5.1, os *flat clusters* seriam Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luis Filipe Vieira).

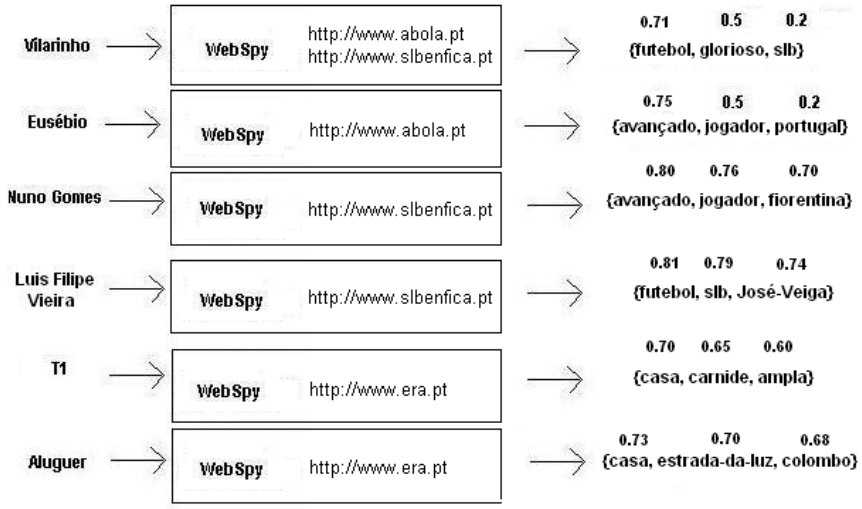

Figura 5.1: Lista dos termos relacionados com cada elemento do conjunto Vilarinho, Eusébio, T1, Aluguer, Nuno Gomes, Luis Filipe Vieira, com indicação da respectiva probabilidade de relevância devolvida pelo WebSpy.

O próximo passo é avaliar a similaridade entre cada um dos *flat clusters*. Essa similaridade é registada numa matriz (simétrica) cuja dimensão é dada pelo número de *flat clusters*. Para a preencher é necessário calcular um número de similaridades dado pela

equação 5.1:

$$n^o \text{ total de similaridades} = \sum_{i=1}^{n-1} i \tag{5.1}$$

Assim, para 6 flat clusters, teriamos uma matriz 6 x 6, e o cálculo de 15 similaridades (ver figura 5.2).

	FlatCluster1	FlatCluster2	FlatCluster3	FlatCluster4	FlatCluster5	FlatCluster6
FlatCluster1		Sim(FC1, FC2)	Sim(FC1, FC3)	Sim(FC1, FC4)	Sim(FC1, FC5)	Sim(FC1, FC6)
FlatCluster2			Sim(FC2, FC3)	Sim(FC2, FC4)	Sim(FC2, FC5)	Sim(FC2, FC6)
FlatCluster3				Sim (FC3, FC4)	Sim(FC3, FC5)	Sim(FC3, FC6)
FlatCluster4					Sim(FC4, FC5)	Sim (FC4, FC6)
FlatCluster5						Sim(FC5, FC6)
FlatCluster6						

Figura 5.2: Exemplo de uma matriz de similaridade para 6 *flat clusters*.

O valor de cada uma das similaridades é obtido com recurso à medida de *Cosine*, que mede a distância entre 2 vectores de dimensão n. No âmbito da nossa aplicação os vectores são constituídos pelos termos relacionados e respectivas probabilidades de relevância obtidas da execução do WebSpy para cada um dos *flat clusters* (ver figura 5.1).

Assim, quando os vectores de 2 *flat clusters* partilham termos iguais, deve proceder-se ao cálculo da similaridade aplicando a medida de *Cosine*, caso contrário se os vectores não partilharem qualquer termo entre eles, o valor da similaridade será zero (ver equação 5.2).

$$Cosine(vector_1, vector_2) = \frac{\sum_{i=1}^{n} pesos(vector_{1,i}) \times pesos(vector_{2,i})}{\sqrt{\sum_{i=1}^{n} pesos(vector_{1,i})^2} \times \sqrt{\sum_{i=1}^{n} pesos(vector_{2,i})^2}} \quad (5.2)$$

O conjunto das similaridades é registado numa matriz simétrica, ponto de entrada para o Poboc, que a partir delas devolve ao sistema um conjunto de *clusters*, organizados de forma hierárquica. Atingimos a este nível o *hierarchical soft clustering* (ver figura 5.3).

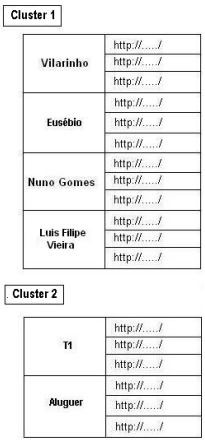

Figura 5.3: Formação de 2 *clusters*.

O último passo que o sistema realiza é o de determinar *labels* para os *clusters* devolvidos:

(1) assim, o nome do *label* de um dado *cluster* é aquele termo que mais vezes aparece partilhado, pelo conjunto de vectores pertencentes ao grupo. Assim para o *cluster* 2 da figura 5.4 e tomando como referência os vectores da figura 5.1, o termo escolhido seria *casa*, dado ocorrer 2 vezes, mais do que qualquer um dos outros termos;

(2) nos casos em que existe mais do que um termo com o mesmo número de ocorrência máxima, o sistema escolhe aquele que obtém a maior soma de pesos (nos vectores em que

ocorre). Assim para o *cluster* 1 da figura 5.4 e tomando como referência os vectores da figura 5.1, existem 4 termos (futebol, slb, jogador e avançado) que ocorrem duas vezes. A escolha recairía no termo *futebol* que se distingue dos restantes por ter uma soma de pesos superior (em concreto 1,52).

Os labels destes clusters seriam portanto os termos *futebol* e *casa* respectivamente (ver figura 5.4).

Figura 5.4: Formação de 2 *clusters* com os respectivos *labels*.

Outras possibilidades de *labelização* foram pensadas mas deixamos esta discussão para o próximo capítulo de conclusões e trabalho futuro.

5.3 Avaliação e Resultados

5.3.1 Avaliação de Sistemas de Tecnologia da Linguagem Humana

A avaliação no âmbito da investigação serve para estabelecer as características do desempenho de um sistema.

Uma avaliação pode ser:

(1) Qualitativa: se envolver observações do sistema ou entrevistas com utilizadores;

(2) Quantitativa: se envolver uma análise estatística para estabelecer a significância e a importância, por exemplo, a nível das diferenças no desempenho.

Existem três tipos de avaliação apropriados para 3 objectivos diferentes:

(1) Avaliação de Adequação (*Adequacy Evaluation*);

(2) Avaliação Diagnóstica (*Diagnostic Evaluation*);

(3) Avaliação de Desempenho (*Performance Evaluation*).

Avaliação de Adequação

É necessário ter conhecimento das necessidades dos potenciais utilizadores do sistema, determinar se os sistemas estão adequados à tarefa a que se propõem, e se sim, averiguar quais os que estão mais adequados. Este tipo de avaliação não se destina necessariamente a identificar o melhor sistema, mas em fornecer informação comparativa que o utilizador pode usar para fazer uma escolha mais acertada. Por isso ela tem de permitir tirar conclusões se um sistema é eficaz ou se ainda é necessário melhorá-lo.

As metodologias para a avaliação de adequação são difíceis de aplicar e não são aceites de forma geral.

Avaliação Diagnóstica

A avaliação diagnóstica é uma metodologia de desenvolvimento comum, que emprega um conjunto de teste de *input*, cujo objectivo é o de constituir as combinações mais importantes e prováveis e definir um conjunto de *inputs* válidos e inválidos. Estes conjuntos de testes são particularmente valiosos para os programadores dos sistemas, permitindo a identificação das suas limitações, erros ou deficiências.

Avaliação de Desempenho

Existe uma longa tradição na avaliação quantitativa do desempenho a nível da pesquisa de informação (*Information Retrieval*). Quando se considera a metodologia para a medição numa dada área, é feita uma distinção entre critério, medida e método.

O critério descreve o que estamos interessados em avaliar: precisão, velocidade, nível de erros.

A medida é a propriedade de performance do sistema à qual nos referimos numa tentativa de chegar ao critério escolhido.

O método especifica como calcular a medida segundo o critério escolhido.

Sucessos e Limitações da Avaliação

A avaliação desempenha um papel crítico no âmbito da investigação, tendo sido dado até ao momento um maior ênfase à Avaliação de Desempenho, com sucesso em problemas específicos:

(1) extracção de informação a partir de textos;

(2) reconhecimento contínuo da fala;

(3) tradução automática;

(4) recolha de informação de larga escala.

Exemplos de sucesso foram a criação de pelo menos 4 conferências e *workshops* relacionadas com a avaliação de desempenho, que atraem um número cada vez maior de investigadores, empresas e governos:

(1) MUC (*Message Understanding Conferences*);

(2) TREC (*Text Retrieval Conferences*);

(3) DUC (*Document Understanding Conference*);

(4) CLEF (*Cross-Language Evaluation Forum*).

A nível de limitações os sistemas de avaliação possuem algumas falhas menos boas:

(1) tem sido prestada até ao momento pouca atenção à avaliação de situações que envolvem várias línguas e continua a ser dado muito ênfase ao inglês à excepção do CLEF.

(2) a avaliação requer muito trabalho, tempo e recursos.

A avaliação de sistemas tornou-se tão importante que se pode tornar numa área de investigação por si só, e assim tentar ultrapassar as falhas com que hoje ainda se debate.

5.3.2 Trabalho Relacionado

Como referem Costa *et al* (2001), a avaliação de resultados é provavelmente a parte mais difícil no desenvolvimento de sistemas IR. Esta dificuldade agrava-se quando se utilizam algoritmos de *clustering*, na medida em que é difícil classificar um *cluster* como sendo bom ou mau.

Técnicas de IR comuns como a precisão e a cobertura, são frequentemente usadas no âmbito da avaliação Quantitativa, mas requerem a formação de um conjunto base de documentos *ground truth* que servirão de comparação para com os resultados provenientes da execução dos sistemas, para além de exigirem a intervenção humana. A avaliação destes sistemas pode ser por outro lado Qualitativa, quando é feita com recurso a questionários, situação que também implica a intervenção humana.

Zamir *et al* (1998) comparam o sistema com uma lista organizada de resultados (classificando manualmente os documentos como relevantes e não-relevantes), provenientes da execução de um conjunto de *queries* num dado motor de busca, aplicando a medida de precisão.

Costa *et al* (2001), recorrem a 2 utilizadores para formarem o *ground truth*, com base em 5 *queries* (50 resultados devolvidos para cada). Esta informação é manualmente agrupada por estes 2 utilizadores e comparada com os resultados devolvidos da execução

da aplicação. Para o complemento desta avaliação, procedem a um estudo qualitativo com a elaboração de um questionário.

Fung *et al* (2003) consideram o seu *ground truth*, com base num conjunto de *corporas* e aplicam uma medida designada por *F-Measure* que tem por base o cálculo da precisão e da cobertura.

Zeng *et al* (2004) utilizam a precisão, como medida para avaliar a performance do sistema, recorrendo à ajuda de 3 utilizadores para validarem a relevância dos documentos.

Ferragina *et al* (2005) comparam o seu sistema com o Vivissimo, seleccionado 20 estudantes da Universidade de Pisa para executarem um conjunto de *queries* nos 2 sistemas, tentando posteriormente aferir de entre os alunos, qual dos 2 sistemas preferiram.

Jiang *et al* (2002) apenas fazem umas experiências preliminares enquanto Zhang *et al* (2001) não referem a esse propósito qualquer tipo de experiência.

5.3.3 Proposta de Avaliação para o TREC

A avaliação do nosso sistema encontra-se numa fase de elaboração, mas temos a este propósito um conjunto de ideias sobre como proceder, razão pela qual propomos para trabalho futuro um conjunto de experiências que tencionamos desenvolver:

(1) avaliar o sistema a curto prazo, tal como ele se encontra implementado. A ideia é verificar a adequação dos resultados com as necessidades do utilizador, ou seja, fazer uma avaliação de adequação. Este tipo de avaliação é qualitativa e tem associado problemas como a subjectividade da análise humana, problemas de logística (muitos recursos, tempo) e problemas de definição dos critérios de avaliação (número de *clusters*?, a ordem dos *clusters*?, etc...);

(2) avaliar o sistema nas conferências TREC, tal como ele se encontra implementado.

Esta avaliação da relevância dos resultados é particularmente influenciada pelo motor de busca escolhido para efectuar a pesquisa da informação. Na prática, a única diferença entre a lista de resultados devolvida pelo motor de busca e a lista de resultados devolvida pelo nosso sistema, é dada a este ponto, pela filtragem de resultados que o nosso sistema faz, com recurso ao cálculo da média de relevância global (ver secção 4.2). Poderemos avaliar assim, da eficiência desta implementação;

(3) avaliar o sistema nas conferências TREC, depois de desenvolvido e implementado o *query expansion*[1]. A este nível estaremos a avaliar a qualidade dos nossos resultados quando aplicados numa outra aplicação, neste caso em aplicações de *Query Expansion*.

As MUC e as TREC têm impulsionado o aparecimento de novas arquitecturas, técnicas e ferramentas. O MUC apresenta-se mais virado para metodologias de avaliação de extracção de informação e o TREC para metodologias de avaliação de recolha de textos.

Uma abordagem do TREC é o HARD (*High Accuracy Retrieval from Documents*) TREC. Esta abordagem consiste na facultação em forma de metadata[2] (ver figura 5.5), por parte dos organizadores, de uma série de *queries* e respectiva informação (descrição da *query*, o tipo de documentos que o utilizador procura, o tipo de informação que os documentos devem devolver, a familariedade do utilizador com o tópico, a quantidade de texto que o utilizador espera em resposta à *query*, etc...)

[1] No âmbito de trabalho futuro, como referido anteriormente.

[2] No total o *corpus* do HARD TREC é de 372,219 documentos, ocupando 1.7Gb de espaço em disco.

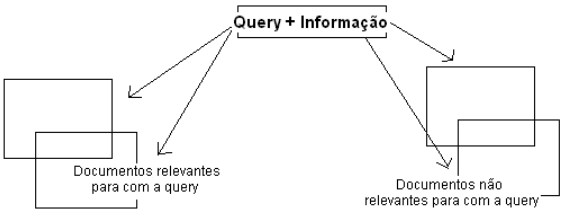

Figura 5.5: *Metadata facultada ao utilizador.*

Para que a exactidão da recolha de textos possa ser medida, os *outputs* têm de ser comparados com a verdade base (*grounded truth*) determinada por humanos. A nossa proposta de avaliação baseia-se nesta informação e na metadata disponibilizada. Assim, o nosso sistema poderá ser avaliado na óptica de *Automatic Query Expansion* com a implementação futura de uma *Web Warehouse* ou na óptica de *Interactive Query Expansion* onde o utilizador escolherá um *cluster* para reformular a *query*. Com base nos resultados produzidos serão então aplicadas as medidas de precisão e cobertura (ver secção 1.1). Veja-se a esse propósito a figura 5.6 que resume a nossa proposta de avaliação futura.

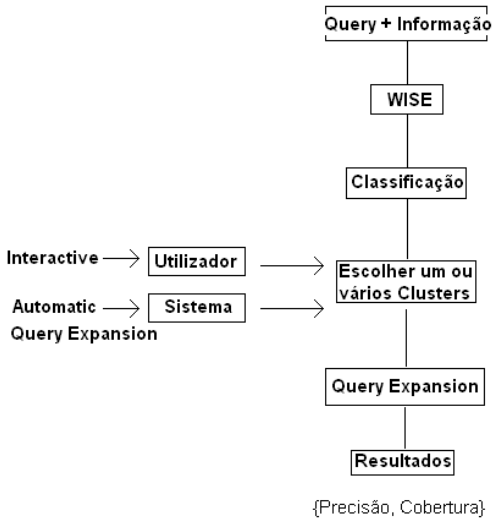

Figura 5.6: *Processo de Avaliação.*

A implementação destas propostas será objecto de trabalho futuro, que não cabe

no âmbito desta dissertação por razões logísticas (tempo e recursos humanos). Assim, disponibilizamos no capítulo seguinte um conjunto de resultados mais conseguidos e outros menos conseguidos, resultantes da execução da aplicação WISE.

5.3.4 Resultados

Os resultados que mostramos em baixo são *clusters* devolvidos da execução da aplicação WISE, para a *query* Benfica no dia 31 de Maio de 2005, tendo por base o motor de busca Google e uma lista de resultados inicial de 100 *URLs*.

O *cluster* que pode ser visto na figura 5.7 refere-se a um conjunto de *URLs* relacionados com José-António-Camacho, antigo treinador do Real-Madrid de quem se fala poder vir a ser o sucessor de Giovanni-Trapattoni no comando técnico do Benfica. Note-se a qualidade dos *labels* José-António-Camacho, Real-Madrid e Giovanni-Trapattoni, *labels monothetic* e extremamente descritivos o que decorre do facto do sistema considerar o cálculo de palavras compostas. É de sublinhar a capacidade do nosso sistema lidar com alguns erros ortográficos (Giovanni-Trapattoni e não Geovanni-Trapattoni). De facto, juntando estes dois termos num mesmo *cluster* poderemos vir a propor termos mais abragentes para a *query expansion*.

Figura 5.7: *Labels monotéticos e palavras compostas.*

Outro dos pontos positivos do sistema referido ao longo da dissertação diz respeito à desambiguição dos termos. Assim, o *cluster* José-António-Camacho (ver figura 5.7) diz respeito ao clube Benfica, o *cluster* PS (ver figura 5.8) ao Partido Socialista com sede em Benfica e o *cluster* universitários (ver figura 5.8) à venda de casas no bairro de Benfica.

Figura 5.8: *Desambiguação de termos.*

A possibilidade de um documento falar de dois ou mais tópicos distintos é um facto que o *overlap* resolve como se pode observar na figura 5.9, onde o *URL http://www.slbenfica* é referenciado por 2 *clusters*.

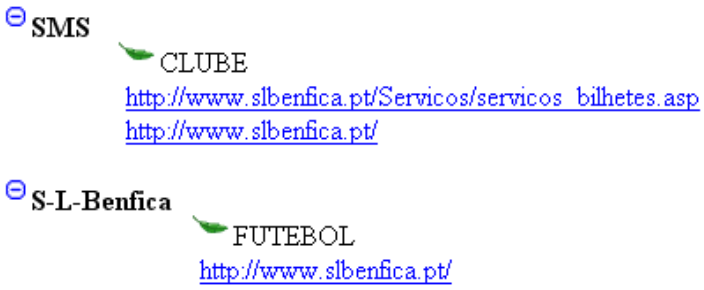

Figura 5.9: *Overlap.*

O facto do sistema ser independente a nível da língua permite ter num mesmo *cluster*

referências a *URLs* em diferentes idiomas. Assim, observe-se da figura 5.10 os *clusters* SL-Benfica e *Champions-League* relacionados com um conjunto de termos expressos na língua portuguesa (o Sporting, a participação do Benfica na taça UEFA e o jogo com o Porto), termos expressos na língua húngara (*Segítségét-köszönjük-Startlap-team*) e na língua inglesa (*Clubs*).

Figura 5.10: *Independente em relação à língua.*

Por outro lado o utilizador perderá menos tempo de pesquisa, se encontrar num *cluster* a junção de um conjunto de referências a um mesmo assunto. A esse propósito veja-se a figura 5.11 com 3 *URLs* para o termo Sporting.

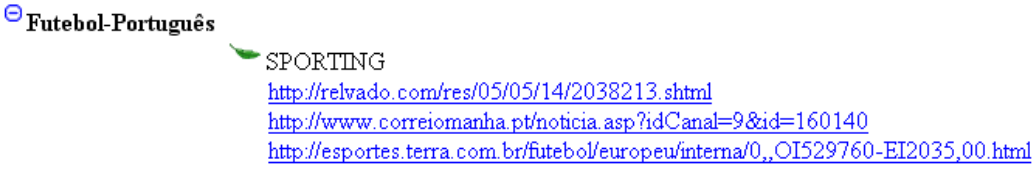

Figura 5.11: *Várias referências para um mesmo assunto.*

Além do mais, por basearmos a nossa pesquisa num conjunto alargado de *URLs*, mais do que os devolvidos pelo motor de busca escolhido para a execução da *query* (ver secção

4.2), a nossa cobertura de resultados é muito maior, como se pode observar da figura 5.12

Figura 5.12: *Recall dos resultados.*

Com base nas figuras seguintes podemos também detectar possíveis melhorias a desenvolver no âmbito da aplicação, para as quais pensamos ter a resposta a implementar no contexto de trabalho futuro. Assim, da figura 5.13 constata-se um *cluster* com o nome Miklos-Fehér que tem como termo relevante associado um termo com o mesmo nome.

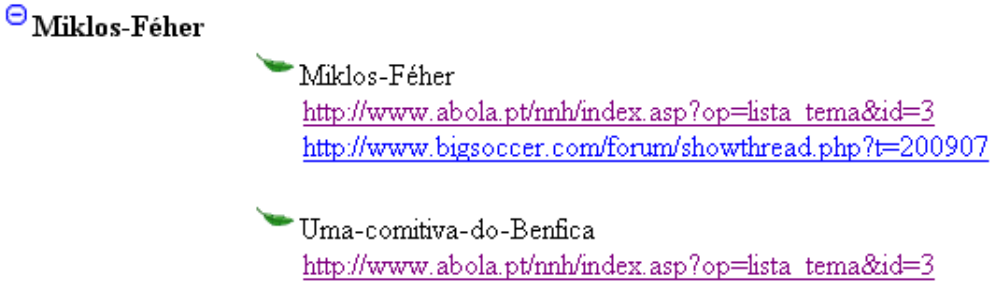

Figura 5.13: *Labels e termos com o mesmo nome.*

Por outro lado o facto de atribuirmos muita importância a palavras maiúsculas (ver figura 5.14) no âmbito do processo de *Web Content Mining* tem o efeito, não obstante a correcta desambiguação do termo Benfica relacionado com o shopping Benfica no Brasil, de assumir como *labels* palavras insuficientemente descritivas dos termos.

Figura 5.14: *Processo de Web Content Mining.*

A proliferação de *clusters* potencialmente relacionados como se pode observar da figura 5.15, é uma deficiência da aplicação que pensamos corrigir com a implementação de uma nova medida de similaridade (ver secção Trabalhos Futuros).

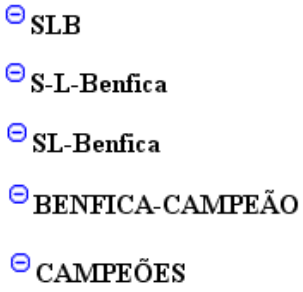

Figura 5.15: *Clusters dispersos.*

Com a descrição dos resultados fechamos um capítulo da dissertação, mas da observação atenta dos mesmos, conclui-se que estes abrem portas para uma nova fase de investigação. A esse propósito veja-se o capítulo Conclusão e Trabalhos futuros.

Capítulo 6

Conclusão e Trabalhos Futuros

6.1 Conclusão

Quando alguém se inicia no desempenho de novas funções deve procurar entender domínios que não domina, iniciar conhecimentos que não conhece e solidificar ideias que pouco entende. O início de qualquer actividade prática deve ser sempre precedido de um estudo teórico.

Para o início da actividade científica é necessário, também, formar uma base de conhecimento. Para isso, estudámos detalhadamente a produção científica existente na área da pesquisa de informação, na área de *web content mining* e na área de *clustering* de documentos *web*. Compilámos toda a informação, aprofundámos conhecimentos, estudámos o que existe publicado na área e avaliámos a possibilidade de propor novas soluções.

A solução que propusemos e implementámos permite mostrar ao utilizador, de forma organizada numa topologia hierárquica, as páginas mais importantes de acordo com a sua pesquisa. Para isso, utilizámos um algoritmo de restrição de páginas, que desconsidera documentos pouco relevantes para com a pesquisa do utilizador. Utilizámos um conjunto de técnicas de *web content mining*, que permitem uma análise semântica dos documentos *web*, entendendo factos que nenhum outro motor de busca, até ao momento, permite

entender.

Igualmente importante para a eficiência da solução foi a adopção do conceito de *phrases*. A utilização de palavras compostas para definir conceitos, quando comparado com a utilização de palavras simples, tem o mérito de permitir um maior entendimento dos mesmos, integrando mais uma vez conhecimento semântico dos documentos.

Esta nova representação da informação, permite subir, no que diz respeito ao entendimento dos documentos, para um nível até agora não alcançado por nenhum outro trabalho. Estes factores, em conjunto com a implementação no domínio da *web*, de um novo algoritmo de *clustering* de documentos, permite apresentar ao utilizador, a informação de forma hierárquica, em tempo real, com metodologias tradicionais de *clustering*.

A arquitectura e os algoritmos propostos dão assim resposta a um dos maiores problemas dos motores de busca: a devolução de resultados com qualidade, através de uma estrutura organizada e desambiguada de conceitos.

A avaliação dos sistemas de pesquisa de informação é um problema decorrente da subjectividade e âmbiguidade da língua natural. No âmbito desta dissertação propomos uma metodologia de avaliação baseada na aplicação do WISE em sistemas de *Query Expansion*. Para este efeito, definimos uma arquitectura para participar no *HARD TRACK* do *TREC* (*Text Retrieval Evaluation Conference*).

No âmbito da tese de mestrado, concluo esta dissertação com o início da minha actividade científica e portanto nada acaba onde tudo pode começar.

6.2 Trabalhos Futuros

Para realizarmos este trabalho de investigação, definimos um plano de projecto que não se esgota com o finalizar desta tese. São definidas em baixo um conjunto de ideias decorrentes da pesquisa de bibliografia, trabalho relacionado e implementação do software, que a curto prazo permitirá uma continuação do estudo na área relacionada.

Recentemente surgiu um novo conceito descrito por Hackathorn (1998), como uma poderosa combinação entre a *Web* e o conceito de *Data Warehouse*: o processo de *Web Farming*. Este consiste numa procura sistemática de conteúdos relevantes na *Web* que alimentem a *Data Warehouse*, complementado-os com os dados provenientes dos sistemas operacionais (ver Figura 6.1).

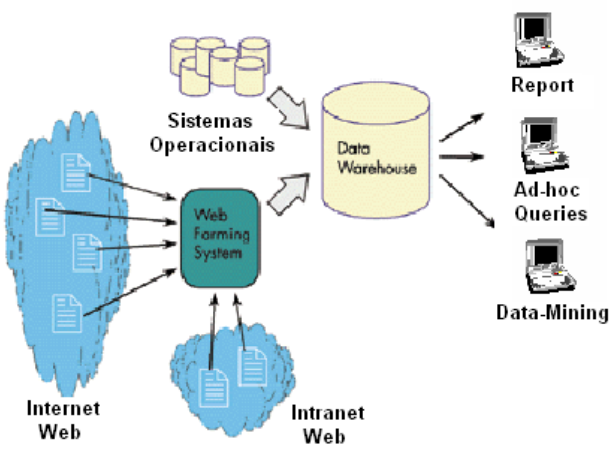

Figura 6.1: O processo de *Web Farming*.

Apesar dos seus imensos recursos, pouca gente considerou usar técnicas de *Web Content Mining* como *input* para uma *Data Warehouse*. O facto do paradigma da *Web* ser radicalmente diferente do que o da *Data Warehouse* exige um trabalho muito apurado: muitos links, pouca disciplina, informação volátil e dispersa, no que usalmente se designa por *spaghetti data*.

A pesquisa de informação e a área de bases de dados, disponibilizam pontos comuns e são uma direcção interessante no contexto de *Web Content Mining*. No seguimento do estudo e da compilação de informação dos artigos relacionados, surge a ideia de implementar uma base de dados com informação relativa ao processo de *Web Content Mining*. Toda esta teoria foi na prática pensada para o conceito do projecto. O seu objectivo é criar uma *DataWarehouse*, um *Thesaurus* com base na pesquisa de termos relacionados, efectuada pela *Web* através de técnicas de *Web Content Mining*. Designamos este conceito de

Web Warehouse.

A integração de informação com base na *WebWarehouse,* poderá ser utilizada, uma vez implementada, em 2 vertentes:

(1) na óptica do utilizador ela pode ser usada para sugerir termos (*Automatic Query Expansion*) relacionados com o tema a pesquisar, obtendo-se desta forma um suporte inicial de resultados mais de acordo com as pretensões do utilizador derivado do refinamento da *query*.

(2) por outro lado e numa vertente técnica, a existência na *WebWarehouse* de termos relacionados com o tema a pesquisar, evitaria o uso repetido do *Spider*, com evidentes melhorias a nível do desempenho do software.

Mas a constante mutação da sociedade, não permite que se tome por definitivo o relacionamento de um termo com outro e torna-se por isso evidente a necessidade de em permanência acompanhar a evolução do dia a dia. Tome-se como exemplo o termo Benfica que com naturalidade estará hoje mais frequentemente relacionado com Luis Filipe Vieira, mas a curto prazo poderá estar relacionado com outro presidente do Benfica que não este. Assim, deverá ter-se em conta aquando da definição e implementação da *WebWarehouse* a problemática da actualização dos dados.

A *WebWarehouse* permitirá também a construção automática e gradual de um *thesaurus* a partir de dados reais, ao invés de dados introduzidos manualmente com a subjectividade humana subjacente a este facto como o *Rodget's Thesaurus* (ver Rodget, 1852) ou a base lexico-semântica *WordNet* (ver Miller, 1995).

No contexto da avaliação da aplicação WISE, no âmbito das conferências HARD TREC, pretendemos implementar num futuro imediato a funcionalidade de *Interactive Query Expansion*. Por outro lado, a sua implementação, tornará o sistema mais funcional ao permitir ao utilizador um refinamento da *query* na procura de novos resultados. Atingiremos a este nível, em conjunto com a implementação do *Automatic Query Expansion*

o *Classified Query Expansion*.

Pretendemos também avaliar a viabilidade em implementar uma nova medida de similaridade, na exacta medida em que a aplicação da medida de *Cosine*, como visto anteriormente, apenas considera dois vectores de contexto de um termo como sendo similares, se os mesmos partilharem palavras comuns entre eles, não considerando a existência de qualquer similaridade no caso contrário. Semelhante avaliação, não pode no entanto, ser de todo considerada eficaz. De facto, com a utilização desta medida estaremos a desconsiderar vectores de contexto que sendo similares não partilham nenhuma palavra em comum, como se depreende da leitura dos seguintes vectores:

(1) Termo Marcador com o vector de contexto: Nuno-Gomes | marcar | ponta-de-lança;

(2) Termo Avançado com o vector de contexto: volta-aos-golos | avançado-centro | Benfica.

A solução passa por considerar informação semântica, que neste caso concreto identificasse Nuno Gomes como o Avançado do Benfica. De outra forma, e a menos que se lide com documentos muito específicos onde os sinónimos raramente existam e a ambiguidade de vocabulário seja reduzida, estaremos a desconsiderar potenciais vectores similares.

Uma forma de evitar a necessidade de detectar a ocorrência de palavras iguais, como forma de avaliação da similaridade entre dois vectores, é a identificação da coesão lexical entre palavras. A esse propósito, muitos autores sugeriram a identificação de relações através do uso de recursos linguísticos tais como as já referidas bases lexico-semânticas *WordNet* (ver Miller, 1995) ou *Rodget's Thesaurus* (ver Rodget, 1852). A utilização destes recursos, tem no entanto o contra de estar apenas disponível para línguas dominantes como o Inglês, o que por consequência torna qualquer tipo de sistema que as use, dependente das mesmas.

A nossa proposta, ao contrário das anteriores, vai no sentido de utilizar uma nova

medida de similaridade, o *InfoSimba* (ver Dias & Alves, 2005), que acreditamos poder vir melhorar o desempenho do sistema WISE, ao utilizar um modelo matemático bem fundamentado, que lida com a co-ocorrência de palavras. Esta medida de similaridade informativa, inclui na sua definição a medida de associação *Equivalence Index Association*, proposta por Muller *et al* (1997) e Silva *et al* (1999), que avalia o grau de coesão entre palavras, que serão tanto mais coesas quanto maior for o valor obtido.

A ideia básica da medida *InfoSimba* é integrar na medida de *Cosine* o factor de co-ocorrência de palavras inferido de uma colecção de documentos, com recurso à medida de associação *Equivalence Index Association*. Assim, $EI(W_{i,k}, W_{j,l})$ é o valor entre a palavra da posição k no vector de contexto i, e a palavra na posição l no vector de contexto j, obtido com recurso a esta medida de associação. Veja-se a equação 6.1, relativa à aplicação do *InfoSimba* a 2 vectores (X_i, X_j):

$$\frac{\sum_{k=1}^{p}\sum_{l=1}^{p} X_{i,k} \times X_{j,l} \times EI(W_{i,k}, W_{j,l})}{\sqrt{\sum_{k=1}^{p}\sum_{l=1}^{p} X_{i,k} \times X_{i,l} \times EI(W_{i,k}, W_{i,l})} \times \sqrt{\sum_{k=1}^{p}\sum_{l=1}^{p} X_{j,k} \times X_{j,l} \times EI(W_{j,k}, W_{j,l})}} \tag{6.1}$$

A medida de similaridade *InfoSimba*, pode simplesmente ser explicada da seguinte forma: Suponha-se um vector de contexto X_i e um outro vector X_j. O valor calculado, resulta da multiplicação entre todos os pesos de todas palavras, multiplicado de seguida pelo grau de coesão existente entre as palavras, dado pela medida de associação *EI*. Como resultado, quanto mais coesas as palavras forem, mais elas contribuirão para a coesão de entre os vectores de contexto.

Da aplicação da medida *InfoSimba* em detrimento da medida *Cosine*, acreditamos que resulte uma melhor determinação de vectores de contexto similares, com consequências positivas também ao nível do *clustering* dos documentos.

Noutra direcção, uma medida que poderá resultar, também, numa valorização dos *clusters*, é um possível *merge* dos mesmos. De facto, a ocorrência de resultados (devolvidos pelo Poboc) muito próximos/similares uns dos outros, não faz qualquer sentido e um *merge* ou um novo agrupamento, poderão resultar num melhor agrupamento de documentos.

Em média, executar a aplicação para uma *query* que devolva muitos resultados, demora cerca de 45 minutos[1], o que inclui a leitura da página, o seu *parsing*, pre-processamento de resultados, a procura de palavras compostas, a procura de termos relacionados e o *clustering* hierárquico de resultados. Garantir a rapidez da aplicação não foi o focus desta dissertação. Tal não invalida que este facto não seja tomado em consideração. Propomos por isso uma optimização da aplicação, paralelizando o seu código ou utilizando técnicas de computação distribuída (*Grid Computing ou Cluster Computing*) que permitirão um aumento da velocidade de execução.

O nível estável em que os Motores de Busca actualmente se encontram, permitiu à comunidade científica o estudo de novas técnicas que esperamos revolucionem a curto prazo, este tipo de ferramentas. Gostariamos de validar no futuro, até que ponto, os actuais Motores de Busca adoptarão estes conceitos e se novos Motores de Busca surgirão por via deste tipo de estudos.

[1]Num computador com sistema operativo Windows XP, equipado com 256Mb de RAM, processador PentiumIV a 2,4 GHz e executado com base numa rede a 100Mbps.

Bibliografia

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, A. (1996). *Fast Discovery of Association Rules*. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, chapter 12, AAAI Press, 307-328

[2] Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley, ACM Press, New York, USA.

[3] Berry, M. & Linoff, G. (2000). *Mastering Data Mining*. John Wiley and Sons, New York, USA.

[4] Chekuri, C., Goldwasser, M., Raghavan, P. & Upfal, E. (1997). *Web Search Using Automated Classification*. In 6th International World Wide Web Conference, (Poster no. POS725), Santa Clara, California, April.

[5] Church K.W. & Hanks P. (1990). *Word Association Norms Mutual Information and Lexicography*. In Computational Linguistics, 16(1), 23-29.

[6] Cleuziou, G., Martin, L. & Vrain, C. (2003). *PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classification and Textual Data*. In Proceedings of the 16th biennial European Conference on Artificial Intelligence (ECAI'04), Valencia, Spain, August, 440-444.

[7] Cooley, R., Mobasher, B. & Srivastava, J. (1997). *Web Mining: Information and Pattern Discovery on the World Wide Web*. In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97).

[8] Costa, M. & Silva, M. (2001). *Ranking no Motor de Busca TUMBA*. In Proceedings of the 4^a Conferência de Redes de Computadores, Tecnologias e Aplicações, Covilhã, Portugal, November.

[9] Daille, B. (1996). *Study and Implementation of Combined Techniques for Automatic Extraction of Terminology*. In P. Resnik and J. Klavans, editors, The Balancing Act: Combining Symbolic and Statistical Approaches to Language, MIT Press, Cambridge, MA, 49-66.

[10] Dias, G. (2002). *Extraction Automatique d'Associations Lexicales à partir de Corpora*. PhD Thesis. DI/FCT New University of Lisbon (Portugal) and LIFO University of Orléans (France).

[11] Dias, G. & Nunes, S. (2004). *Evaluation of Different Similarity Measures for the Extraction of Multiword Units in a Reinforcement Learning Environment*. In Proceedings of the 4th International Conference On Languages Resources and Evaluation, M.T. Lino, M.F. Xavier, F. Pereira, R. Costa and R. Silva (eds), Lisbon, Portugal, May 26-28. ISBN: 2-9517408-1-6. EAN: 0782951740815. 1717-1721.

[12] Dias, G. & Alves, E. (2005). *Language-Independent Informative Topic Segmentation*. In Proceedings of the 9th International Symposium on Social Communication. Center of Applied Linguistics, Santiago de Cuba, Cuba, January 24-28.

[13] Dias, G., Guilloré, S., & Lopes, J.G.P. (1999). *Language Independent Automatic Acquisition of Rigid Multiword Units from Unrestricted Text Corpora*. In Proceedings of 6th Conférence Annuelle du Traitement Automatique des Langues Naturelles. Institut dt'Etudes Scientifiques, Cargèse, France, July 12-17.

[14] Díaz-Galiano, M.C, Martín-Valdivia, M.T., Martínez-Santiago, F. & Ureña-López, L.A. (2004). *Multiword Expressions Recognition with the LVQ Algorithm*. Workshop on Methodologies and Evaluation of Multiword Units in Real-world Applications (MEMURA Workshop) associated with the 4th International Conference Languages Resources and Evaluation. Dias, G., Lopes, J.G.L. & Vintar, S. (eds), Lisbon, Portugal. May 25. ISBN: 2-9517408-1-6. EAN: 0782951740815. 12-17.

[15] Dunning, T. (1993). *Accurate Methods for the Statistics of Surprise and Coincidence*. In Computational Linguistics, 19(1), 61-74.

[16] Fagan, J. L (1987). *Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods*. Ph.D. Thesis, Cornell University.

[17] Ferragina, P. & Gulli, A. (2005) *A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering*. In the Proceedings of the 14th international conference on World Wide Web, Chiba, Japan, ISBN:1-59593-051-5. 801-810.

[18] Fung, B., Wang, K. & Ester, M. (2003) *Large hierarchical document clustering using frequent itemsets*. In Proceedings of the SIAM International Conference on Data Mining, Cathedral Hill Hotel, San Francisco, CA, May 1-3.

[19] Gil, A. & Dias, G. (2003). *Using Masks, Suffix Array-based Data Structures and Multidimensional Arrays to Compute Positional Ngram Statistics from Corpora*. In Proceedings of the Workshop on Multiword Expressions of the 41st Annual Meeting of the Association of Computational Linguistics, Sapporo, Japan, July 7-12, ISBN: 1-932432-20-5, 25-33.

[20] Hackathorn, R. (1998). *Web Farming for the Data Warehouse*. Morgan Kaufmann Publishers, New York. USA.

[21] Hearst, M. & Pedersen, J. (1996). *Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results*. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), Zurich, Switzerland, August, 76-84.

[22] Jeff, H. (2002). *Programming Spiders, Bots and Aggregators in Java*. ISBN: 0782140408. Sybex. USA.

[23] Jiang, Z., Joshi, A., Krishnapuram, R. & Yi, L. (2002). *Retriever: Improving web search engine results using clustering*. In Managing Business with Electronic Commerce.

[24] Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley and Sons, New York. USA.

[25] Kosala, R. & Blockeel, H. (2000). *Web Mining Research: a Survey*. ACM SIGKDD Exploration 2(1). 1-15.

[26] Leouski, A. & Croft, B. (1996). *An Evaluation of Techniques for Clustering Search Results*. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst.

[27] Liu, B., Chin, C.W., & NG, H.T. (2003). *Mining Topic-Specific Concepts and Definitions on the Web*. In Proceedings of the 12th International WWWConference (WWW03), Budapest, Hungary.

[28] Manning, C.D. & Shutze, H. (1999). *Foundations of Statistical Natural Language Processing*. London. MIT Press.

[29] Martins, B. & Silva, M. (2003). *Web Information Retrieval with Result Set Clustering*. In Proceedings of NLTR 2003 - Natural Language and Text Retrieval Workshop at EPIA03, December.

[30] Miller, G. (1995). *Lexical Database for English*. In Communications of the ACM, 38(11). 39-41.

[31] Muller, C., Polanco, X., Royauté, J. & Toussaint, Y. (1997). *Acquisition et Structuration des Connaissances en Corpus: Éléments Méthodologiques*. Technical Report RR-3198, Inria, Institut National de Recherche en Informatique et en Automatique. http://www.inria.fr/rrrt/rr-3198.html

[32] Ogata, T., Terao, K. & Umemura, K. (2004). *Japanese Multiword Extraction using SVM and Adaptation*. In Workshop on Methodologies and Evaluation of Multiword Units in Real-world Applications (MEMURA Workshop) associated with the 4th International Conference Languages Resources and Evaluation. Dias, G., Lopes, J.G.L. & Vintar, S. (eds), Lisbon, Portugal. May 25. ISBN: 2-9517408-1-6. EAN: 0782951740815. 8-12.

[33] Page, L. & Brin, S.(1998). *The Anatomy of a Large-Scale hypertextual Web Search Engine*. In Proceedings of the 7th World Wide Web Conference (WWW7), Bisbane, Australia, 107-117.

[34] Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). *The PageRank Citation: Bringing Order to the Web*. Computer Science Departament, Standford University.

[35] Rodget's, P. (1852). *Thesaurus of English Words and Phrases*. Longman, London.

[36] Salem, A. (1987). *La Pratique des Segments Répétés*. Klincksieck, Paris.

[37] Salton, G., Yang, C. S. & Yu, C. T. (1975) *A theory of Term Importance in Automatic Text Analysis*. In Journal of the American Society for Information Science, 26(1). 33-44.

[38] Salton, G., Wong, A. & Yang, C. S. (1975). *A Vector Space Model for Automatic Indexing*. In Communications of the ACM 18(11). 613-620.

[39] Sparck-Jones, K. (1972). *A Statistical Interpretation of Term Specificity and its Application in Retrieval*. In Journal of Documentation, 28(1). 11-21.

[40] Sanderson, M. & Croft, W. (1999). *Deriving Concept Hierarchies from Text*. In Research and Development in IR. 206-213.

[41] Shimohata S. (1997). *Retrieving Collocations by Co-occurrences and Word Order Constraints*. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, 1997, 476-481.

[42] Silva, J. & Lopes, J.G.P. (1999). *A Local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units*. In Proceedings of the 6th International Conference on Mathematics of Language. Orlando, USA. July 23-25.

[43] Silva, J., Dias, G., Guilloré, S. & Lopes, J.G.P. (1999). *Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units*. In Proceedings of 9th Portuguese Conference in Artificial Intelligence, Pedro Barahona and Júlio Alferes (eds), Lecture Notes in Artificial Intelligence n^{o}1695, Springer-Verlag, Évora, Portugal, September 21-24. ISBN: 3-540-66548-X, 113-132.

[44] Silverstein, C., Henzinger, M., Marais, H. & Moricz, M. (1998). *Analysis of a Very Large Altavista Query Log*. Technical Report 1998-014, Digital SRC.

[45] Smadja F. (1993). *Retrieving Collocations From Text: XTRACT*. In Computational Linguistics, 19(1). 143-177.

[46] Tomokiyo, T. & Hurst, M. (2003). *A Language Model Approach to Keyphrase Extraction*. In Proceedings of Workshop on Multiword Expressions of the 41st ACL meeting. July 7-12, Sapporo, Japan, 33-41.

[47] Vechtomova, O., Karamuftuoglu M. & Skomorowski J. (2004). *Approaches to High Accuracy Retrieval*. In the Proceedings of the 13th Text Retrieval Conference, Gaithersburg, November 16-19.

[48] Veiga, H., Madeira, S. & Dias, G. (2004). *Webspy*. Technical Report n^o 1/2004. http://webspy.di.ubi.pt

[49] Wegrzyn-Wolska, K. (2004). *FIM-Metaindexer: a Meta Search Engine Purpose-Built for the French Civil Administration and Statistical Evaluation of Search Engines*. WSS04 The Second International Workshop on Web-based Support Systems avecle IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China, September.

[50] Willet, P. (1990). *Parallel Database Processing, Text Retrieval and Cluster Analysis*. Pitman Publishers, London. UK.

[51] Xu, J. & Croft, W.B. (1996). *Query Expansion Using Local and Global Document Analysis*. In Proceedings of the 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval.

[52] Yang, S. (2003). *Machine Learning for Collocation Identification*. In Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, Chengqing Zong (eds), Beijing. China, IEEE Press, October 26-29. ISBN: 0-7803-7902-0, 315-321.

[53] Zaiane, O. (1998).*From Resource Discovery to Knowledge Discovery on the Internet*. School of Computing Science, Simon Fraser University, Canada.

[54] Zain, A., Murty, M. & Flynn, P. (1999). *Data Clustering: A Review*. In ACM Computing Surveys, 31(3), September.

[55] Zamir, O. & Etzioni, O. (1998). *Web Document Clustering: A Feasibility Demonstration*. In Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR98), 46-54.

[56] Zeng, H., He, Q., Chen, Z. & Ma, W.(2004). *Learning to cluster web search results*. In the Proceedings of the 27th annual international conference on Research and development in information retrieval, Sheffield, UK, ISBN:1-58113-881-4, 210-217.

[57] Zhang, D. & Dong, Y. (2001). *Semantic, Hierarchical, Online Clustering of Web Search Results*. In Proceedings of the 6th Asia Pacific Web Conference (APWEB), Hangzhou, China, April.