David José Machado Ferreira

Procura Estruturada de Textos para Perfis de Utilizadores

Universidade da Beira Interior
Departamento de Informática
Agosto 2009

David José Machado Ferreira

Procura Estruturada de Textos para Perfis de Utilizadores

Tese submetida ao Departamento de Informática para o preenchimento dos requisitos para a concessão do grau de Mestre efectuada sob a supervisão do Doutor Gaël Harry Adélio André Dias,
Universidade da Beira Interior, Covilhã, Portugal

Universidade da Beira Interior
Departamento de Informática
Agosto 2009

Agradecimentos

Primeiramente, começo por agradecer ao meu orientador, Gaël Dias, pelo apoio e ajuda incontestáveis. Foi um prazer realizar o trabalho desta tese na sua companhia, onde me foi possibilitado atingir um novo nível de experiencia com toda a discussão de ideias e explicações dadas sobre assuntos relacionados com o trabalho da tese mas, também, sobre todo um outro conjunto variadíssimo e interessante de temas.

O meu obrigado também aos membros do HULTIG, em especial ao Tiago que me ajudou e fez companhia pelo laboratório durante este longo período.

Agradeço também a todos os meus amigos que sempre me ouviram e apoiaram nos momentos mais complicados durante todos estes anos.

Não menos importante, quero deixar aqui um agradecimento especial á minha família pelo apoio incondicional que sempre me deu. O meu muito obrigado, pois sem vocês não conseguiria e por isso vos dedico este projecto.

Dizem que os últimos são sempre os primeiros, eu concordo, e por isso agradeço finalmente à minha russa de olhinhos azuis. Essa linda namorada cujo sou um felizardo em ter, acompanhou-me sempre em todos os momentos, dando-me o animo e a força necessária para continuar com o trabalho mesmo quando passava por fases menos boas da vida.

O meu muito obrigado!

Resumo

A dimensão cada vez mais colossal e a falta de estruturação da informação existente na *Web*, está a fazer com que os sistemas de *Information Retrieval* enfrentem graves dificuldades no cumprimento dos objectivos para os quais foram criados. Torna-se cada vez mais difícil para estes sistemas encontrarem um conjunto limitado e valioso com a informação procurada pelo utilizador. A consciência deste facto ao longo dos anos tem proporcionado um aumento da comunidade de investigadores que se debatem na exploração de diversos temas e conceitos capazes de solucionarem os problemas.

Das diversas propostas, as mais promissoras são o *clustering* dos resultados e a personalização. O *clustering* permite que a informação devolvida seja organizada em categorias, dando ao utilizador a possibilidade de restringir a sua área de procura com base na escolha das categorias mais apelativas. Por sua vez, a personalização procura escolher automaticamente os resultados que são mais próximos do perfil do utilizador, efectuando uma reordenação da informação de acordo com os interesses de cada um.

A comunidade científica ainda não construiu um sistema que fizesse a união das duas metodologias no que toca à personalização e necessária construção de perfis de utilizadores, com base na categorização dos resultados de pesquisas e das respectivas *queries* introduzidas pelos utilizadores. A criação de *perfis* passa geralmente pela obtenção das categorias de interesse com base na análise dos documentos lidos e classificados como relevantes para o utilizador. Estas categorias poderiam ser extraídas sem a necessidade de efectuar este tipo de análise de documentos, bastando para isso fazer uma relação entre *queries* e *clusters* dos resultados associados a essas *queries*.

Este é o tema da proposta de trabalho apresentada nesta tese. A criação de perfis de utilizadores tendo por base a análise do histórico das pesquisas efectuadas pelos mesmos, bem como a categorização dos resultados para extrair conhecimento oculto e auxiliar a criação destes perfis. Ao serem criados perfis de utilizadores, torna-se possível para um sistema identificar o conteúdo que está mais associado aos interesses de cada pessoa, potencializando-se assim a interactividade entre o sistema e o utilizador.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos e definições

Stemming: Processo de transformar uma palavra na sua palavra raiz

IR: Information Retrieval, metodologias que procuram obter de um conjunto de informação um subconjunto com informação relativa a algo que se procure. No português corrente esta área designa-se por recuperação de informação

WebIR: Aplicação das metodologias de recuperação de informação ao conteúdo existente na Web.

Web: World Wide Web

String : Conjunto de caracteres que representam informação sobre a forma de palavras ou outro tipo de simbologias.

Meta-motor de pesquisa: Sistema que providencia resultados para uma querie a partir da aglomeração dos resultados de diversos motores de pesquisa.

Search-Engine: Sistema que recebe uma querie, efectua a procura de conteúdo que esteja relacionado à querie introduzida e efectua a devolução (mostragem) desse conteúdo.

Stop-words: Conjunto de palavras ou termos que não representam conhecimento.

Querie: Conjunto de palavras-chave que identificam um assunto para o qual se procura informação.

SVM: Support Vector Machine.

Cluster: Grupo.

STC: Suffix Tree Clustering

Snippet: Conjunto de expressões que procuram de forma muito resumida elucidar o conteúdo de um documento.

Crawler: Componente de um motor de pesquisa que percorre os documentos (percorrendo as hiperligações), adicionando a um índice os urls, palavras e texto que encontra.

Capítulo 1

1 - Introdução

1.1 - Origem dos sistemas de recuperação de informação

Dwight D. Eisenhower quando iniciou o projecto ARPA e a criação da rede ARPANET com certeza não imaginou a revolução a que estaria a dar início. Os protocolos e conhecimentos extraídos deste projecto deram lugar ao que hoje é conhecido por internet, a rede que mudou o mundo. Inicialmente, esta rede mapeava um único directório e praticamente todos os servidores eram conhecidos mas, expandiu-se num instante e sendo o conteúdo editado manualmente tornou-se obvia a dificuldade em manter uma hierarquia desta estrutura. Como resultado, foi necessário adoptar técnicas de recuperação de informação desenvolvidas para outras áreas e adapta-las à Web.

Os primeiros sistemas de pesquisa que surgiram não mantinham informação relativa ao conteúdo das páginas Web, apenas indexavam os títulos. Em 1994, o próximo passo foi dado e em vez dos títulos, todo o conteúdo dos documentos passou a ser utilizado na indexação, de modo a que o utilizador pudesse procurar dentro do conteúdo das páginas e não apenas nos títulos. O mercado expandiu-se, tornando empresas como a Yahoo [2u], Altavista [3u] referências no sector mas, em 1998, aconteceu aquilo que Bill Gates mais temia [4u] o Google [1u] apareceu e veio mudar tudo. Era um motor de pesquisa criado no anonimato e revolucionário, capaz de proporcionar resultados bem melhores que os motores de busca existentes. Este novo sistema considerava a estrutura de hiperligações ente os documentos na Web e não apenas o seu conteúdo. O algoritmo denominado de PageRank, introduziu o conceito de citação na Web: quanto mais citações um documento da Web tenha, maior importância lhe é dado. Ainda, quanto maior importância tiver a fonte citadora, mais importante se torna este documento alvo.

Nos tempos correntes os motores de pesquisa estão massificados e o seu uso e importância não pode ser negado. Estudos no mercado dos Estados Unidos, [4] revelam terem sido efectuadas 9.4 biliões pesquisas nos principais motores de busca apenas no

mês de Maio 2009 e mostram existir uma crescente afluência por parte dos utilizadores aos motores de pesquisa a cada ano que passa na ordem dos 20%.

1.2 - Problemática dos motores de pesquisa na Web

A dimensão cada vez mais colossal e a falta de estruturação da informação está a fazer com que os sistemas de recuperação de informação (Information Retrieval **IR**) enfrentem graves dificuldades no cumprimento dos objectivos para os quais foram criados. A consciência deste facto ao longo dos anos tem proporcionado um aumento da comunidade de investigadores que se debatem na exploração de diversos temas e conceitos capazes de solucionarem os problemas.

O objectivo puro de um motor de pesquisa é devolver os documentos considerados relevantes para uma sequência de palavras-chave (querie) fornecida por um utilizador. Por documento, considera-se qualquer tipo de ficheiro cujo possa ser particionado num conjunto de palavras ou expressões regulares sendo tipicamente uma pagina Web, PDF, um documento de texto, tabela ou PS. A querie geralmente é representada por um conjunto de palavras pertencente a uma língua e cujas são consideradas palavras com uma forte relação á informação pretendida. Os motores de busca podem-se classificar como sendo **globais**, quando devolvem resultados provenientes de uma escolha face a um conjunto muito disperso e numeroso de documentos existentes na internet. Acabam por funcionar como portais para qualquer página Web. Caso estejam indexados num domínio específico como um site, são denominados por **locais** visto que a sua área de informação é restrita.

Ponderando um pouco sobre os biliões de páginas existentes acessíveis na rede e a complexidade da linguagem, é fácil perceber que os problemas com que se deparam os sistemas WebIR não são poucos nem triviais. Algoritmos eficientes de crawling, capacidade de processamento distribuído, capacidade de indexação distribuída, adaptação ao ritmo alucinante de novos protocolos Web e conteúdos multimédia, algoritmos de filtragem linguísticos, algoritmos de reconhecimento de spam, são apenas alguns exemplos do quanto é exigente e complexo montar um sistema de recuperação de informação.

Parte de todos estes componentes, servem para resolver problemas de outra ordem. Os próximos cinco parágrafos apresentam os principais desafios, segundo Gulli, com que se deparam estes sistemas.

O primeiro, está na própria definição da palavra "relevante". Por muito tempo se estudaram diversas abordagens e muito esforço foi dispendido na criação de algoritmos capazes de providenciar uma selecção de documentos da Web mais relevantes para um determinado conjunto de palavras-chave. A mais potente metodologia foi introduzida com a adopção do conceito de redes sociais e análise de hiperligações como o caso dos algoritmos Pagerank e Hits. Contudo, em muitas situações esta não é a solução mais perfeita, pois não consegue ultrapassar certos problemas inerentes á complexidade da estrutura da Web. Para queries específicas, onde o número de documentos relacionados existentes na rede é reduzido, torna-se um problema encontra-los. Para queries mais genéricas, que apontam para diversos assuntos, acabam por existir milhões de documentos relevantes e torna-se extremamente complexo saber qual a melhor ordem de ordenação mesmo recorrendo a estes algoritmos. Um dos outros aspectos relevantes para o processo de ranking é que este se baseia numa metodologia. Tal efeito leva a que muitas entidades (Empresas cujo comercio depende muito da visibilidade das suas paginas num motor de busca) explorem o comportamento dos algoritmos de modo a encontrarem a técnica que lhes possibilite um melhor ranking das suas páginas. Este tipo de comportamento é conhecido por optimização para os motores de busca e é legal. Infelizmente, existe quem se aproveite destas técnicas para dar ênfase a páginas cujo conteúdo não é digno de interesse mas dada a sua estrutura, aos olhos dos motores de pesquisa podem ser classificadas como relevantes, deteriorando os resultados.

O segundo desafio, depreende-se com o já referido tamanho da quantidade de informação disponível na internet. A qualidade de um motor de pesquisa depende em muito da completude e frescura do seu índice de páginas que deve conter poucos documentos desactualizados e hiperligações quebradas. Infelizmente, a explosão de informação digital existente na rede está a tornar esta tarefa impraticável. Se por um lado começa a ser impossível indexar todos as paginas, por outro, o número de documentos relevantes devolvidos pelo sistema para uma pesquisa também aumenta (milhões de paginas relevantes não é um bom indicador de eficiência), visto ser proporcional ao tamanho da rede. O problema da indexação de todos os documentos, pode ser resolvido com a recorrência a um meta-motor de pesquisa [9] que explora um

conjunto de resultados provenientes de múltiplos motores e como tal a área de cobertura da rede também aumenta. Contudo, isto leva a um outro problema que é a reordenação por grau de interesse de todo o conjunto de documentos e o tamanho ainda maior deste.

A terceira dificuldade depreende-se com o facto da relevância de um documento ser subjectivo e dependente do contexto pretendido. O mesmo conjunto de palavras-chave pode representar interesses diferentes conforme os interesses de cada utilizador e pode ate, ser dependente do tempo. Imagine-se o seguinte caso: dois utilizadores introduzem a palavra programação. Porem um deles é completamente alheio ao termo e procura simplesmente a definição. Um outro utilizador, já mais familiarizado pode pretender conhecer linguagens de programação e até num outro momento do espaço temporal simplesmente querer dicas sobre programação.

O quarto desafio centra-se no interesse por parte do utilizador em obter informação actual. Basta dar-se um acontecimento invulgar, que suscite o interesse das pessoas e estas tendem instantaneamente a colocar no motor de pesquisa uma querie relativa ao assunto com o intuito de obter informação actual. Neste caso, o interesse é colocado em artigos noticiosos, que para poderem ser dados como relevantes não podem ser classificados com as mesmas técnicas de análise por relação entre hiperligações, visto que, muito normalmente uma notícia recente aponta para praticamente nenhum outro documento. Um caso bastante recente que confirma este tipo de situação foi a da morte do cantor Michael Jackson, cujo número de pesquisas efectuadas ao mesmo tempo foi tão elevado que um dos mais famosos motores de busca foi obrigado a ficar indisponível por uns minutos [5u].

Por fim, um outro entrave relevante, está relacionado com os sinónimos (diferentes palavras podem ter o mesmo significado) e palavras polissémicas (uma palavra pode ter diferentes significados) presentes nas diversas linguagens, o que faz com que nem sempre uma querie possa ser bem definida e explícita. Por exemplo, a palavra sol pode ser relativa à estrela principal do nosso sistema solar, ou a um jornal conhecido. Os documentos devolvidos para uma querie deste tipo abordarão um pouco de todas as possibilidades, resultando numa lista, algo confusa, para um utilizador que procure apenas um destes assuntos.

1.3 - A nova geração de sistemas WebIR

Face aos problemas correntes e previamente referidos com que os sistemas de recuperação de informação se deparam, torna-se fulcral encontrar novas metodologias capazes de responder e satisfazer os objectivos básicos de qualquer motor de pesquisa: devolver os resultados mais relevantes para o utilizador. Uma análise pela literatura actual e somos capazes de constatar uma grande contribuição neste âmbito, por parte de variadíssimos autores, onde múltiplas teorias e experiencias parecem fazer crer que os sistemas WebIR serão capazes de respirar novamente e fazer aquilo para o qual foram construídos. Apesar de diversas, as abordagens recaem sobre três tópicos: ordenação (ranking), categorização (clustering) e personalização [1] [3] [5] [6] [7] [10] [23] [24] [25]. Muitas das soluções passam pela colaboração aos pares destes tópicos, ordenação e categorização, ordenação e personalização, mas poucas fazem o uso colaborativo de todas elas [10]. A nossa aposta vai para esta última abordagem, na qual pensamos estar a chave para o sucesso da nova era dos sistemas WebIR, ainda que haja um longo caminho a percorrer.

Categorização pode ser definido como, o processo de agrupamento de objectos existentes num conjunto inicial em novos subconjuntos cujos partilham de propriedades semelhantes. No caso concreto da categorização de documentos para uma pesquisa Web, (Web search clustering) isto significa dizer que, após a recepção de um conjunto de documentos relevantes para uma querie, estes são agrupados em novos conjuntos tendo por base a semelhança que os documentos têm entre si. Esta geração é derivada de um processo automático e não deve ser confundido com classificação de documentos para o qual, são pré definidas categorias manualmente e cujos documentos são atribuídos à categoria cujas suas propriedades lhe são mais próximas. A categorização, é um processo fulcral quando se trata de estruturar e organizar amplos conjuntos de informação heterogénea, (a Web é um claro exemplo disso) como os resultados provenientes de uma pesquisa. A natureza navegável da hierarquia criada dinamicamente quando aliada a uma boa discrição sobre o conteúdo de cada grupo, ajuda o utilizador a identificar mais rapidamente os resultados mais relevantes para o seu interesse temporal. A pessoa, tem a possibilidade de consultar num espaço compactado os vários assuntos inicialmente dispersos por entre os resultados, navegar para aquele que lhe mais é chamativo e analisar um conjunto mais restrito, mas bem

mais representativo do tópico seleccionado. No fundo, a categorização pode actuar como um catalisador que ajuda o utilizador a resolver em tempo real os problemas derivados da polissemia e dos sinónimos, para além de extrair conhecimento inicialmente oculto.

A literatura apresenta uma diversidade de algoritmos capazes de gerar uma boa categorização dos resultados (provenientes de uma primeira fase de extracção dos documentos mais relevantes para uma querie) baseados em múltiplas ideologias [5] [6] [7] [10] [30] [31] [32] [33] [34]. Estas passam por efectuar primeiro os agrupamentos e depois ver qual a melhor expressão que representa cada conjunto, a, analisar primeiro todo o conjunto de documentos para extrair as melhores expressões e depois agrupar os documentos com proximidade a essas expressões. Facto que praticamente todas as abordagens impõe é mesmo o processo de categorização da informação, como sendo uma boa alternativa á actual lista de resultados devolvida pelos sistemas WebIR.

A personalização é todo o processo efectuado para adaptar um qualquer conteúdo a uma entidade específica. Visto sobre o contexto de um sistema WebIR, significa realçar todos os resultados da pesquisa que estejam mais relacionados com o utilizador ou, como utopia, devolver exclusivamente os conteúdos que no seu imaginário ele pretende. Isto só é passível de ser efectuado quando o sistema possui conhecimento sobre a pessoa para a qual esta a devolver os resultados. Tal conhecimento é representado através de um modelo ou perfil de utilizador, expressão regularmente utilizada por diversas entidades como psicólogos, sociólogos, investigadores criminais entre outros de forma a caracterizarem um grupo ou individualidade de acordo com aquilo que lhes é mais específico. Um psicólogo caracterizará certamente um paciente com um registo da sua infância, dos seus hábitos sociais, dos seus problemas saúde ou familiares, para poder mais tarde tomar uma medida capaz de reequilibrar esse ser. Um modelo de utilizador para um sistema de WebIR é todo este registo de informação que permite identificar aquilo que o usuário é de modo a que o sistema possa, com o auxílio desta informação, ultrapassar os problemas da ambiguidade e subjectividade, apresentando os resultados com real interesse. Imaginem-se, dois utilizadores apaixonados pelos seus clubes respectivamente FC Porto e SL Benfica, a introduzirem a mesma querie “futebol”. Sem a interacção com os moledos de utilizador, o sistema retornará os mesmos resultados para ambos os interessados. Porém, recorrendo aos perfis de cada utilizador, o sistema

será capaz de apresentar resultados diferentes a cada utilizador indo ao encontro das suas pretensões e facilitando a descoberta mais rápida de documentos de interesse para qualquer uma das partes.

Este tipo de personalização no campo das ciências computacionais enfrenta dois grandes desafios.

O primeiro depreende-se com a dificuldade em criar perfis de forma automática e com um grau de precisão considerável. Diversas metodologias [17] [18] [19] [20] aspiram a resolver o problema recorrendo a múltiplos recursos de carácter explicito e não explicito para o utilizador. Se [24] [27] recorrem á indexação das páginas Web referentes ao histórico de navegação dos utilizadores para extraírem a informação representativa dos interesses com base em técnicas de categorização, outros preferem faze-lo através da análise e extracção de conhecimento a partir da estrutura das queries [26]. Outras abordagens, que permitem ver o quanto qualquer área da ciência pode contribuir para uma melhoria dos sistemas são as abordagens de [11] onde algoritmos genéticos são utilizados para extrair conhecimento dos documentos consultados e até mesmo tentar prever o comportamento do utilizador. De modo a tentar melhorar os seus resultados, muitas das abordagens permitem ao utilizador modelar e interagir com os perfis propostos, mas num mundo onde são poucos os utilizadores que perdem tempo a passar á próxima página de resultados, será algo estranho observa-los a optimizar o seu próprio perfil.

O segundo deve-se a questões éticas mais concretamente a privacidade, motivo pelo qual cada vez mais movimentos de protesto são realizados pelo mundo [8u].

Actualmente, no âmbito comercial os principais motores WebIR, devolvem o mesmo resultado independentemente da pessoa que efectuou a pesquisa. Os sistemas de categorização como o do Vivíssimo [6u] dão provas das potencialidades desta tecnologia como forma de ajudar o utilizador a navegar pela quantidade imensa de resultados. Sistemas capazes de modelar o comportamento e associa-lo a um motor de pesquisa global começam a surgir, contudo ainda que numa fase de protótipo. Os seus resultados porém mostram melhorias relativas ao nível da ordenação dos documentos, quando se tem em conta o perfil dos utilizadores. Normalmente, o processo de categorização serve para gerar e mostrar as categorias inicialmente ocultas por entre os resultados ou, no caso da personalização para gerar os modelos de utilizador através da

sua sobreposição aos documentos cujo utilizador achou importantes. Nós pensamos haver uma outra forma de utilizar a categorização. Fazer dela uma ponte de ligação entre tudo aquilo que o navegador pesquisa na Web (histórico das pesquisas) e é considerado relevante, de forma a criar um modelo de utilizador construído simplesmente a partir deste histórico e das categorias associadas a estas pesquisas. Este modelo de utilizador pode por fim, ser utilizado para diversos fins, desde pesquisas na Web ao comércio centralizado.

1.4 – Motivação e Objectivos

Sendo informáticos, uma das ferramentas que mais utilizamos é a recorrência a um motor de pesquisa para encontrar informação sobre todo um tipo de coisas relativas á área (e não só claro) que muitas vezes ainda não estão sequer num formato físico, passível de ser folheado. A natureza de constante inovação na área leva a que isto aconteça, saindo a cada dia que passa uma nova tecnologia que apenas é analisada em formato electrónico. Quem pretende estar na vanguarda da arte tem mesmo de recorrer a um sistema que encontre informação sobre estas tecnologias. Ninguém pode negar que um motor de pesquisa não funciona ou não cumpre os seus objectivos, contudo praticamente toda a gente afirma que muitas vezes é bastante difícil encontrar informação sobre aquilo pelo que se procura. Aos sermos um dos grupos de pessoas que talvez mais utilizam os motores de busca, somos os primeiros a apercebermo-nos do quanto estas afirmações são verdadeiras.

O objectivo desta tese é o de melhorar os resultados devolvidos por um sistema de recuperação de informação fazendo uso da unificação das duas soluções actualmente mais em voga no que toca ultrapassar as barreiras impostas pela ambiguidade das queries, quer seja pela subjectividade ou polissemia. São elas a categorização automática dos resultados mais frequentemente denominada por clustering e a personalização.

Ambas as soluções estão ainda numa fase muito embrionária de adaptação aos sistemas de IR comerciais porem a categorização leva uma ligeira vantagem no que toca

à implementação nesta área graças ao poderoso motor de busca Vivíssimo [6u] cujo é referencia neste sector. Por outro lado a integração de um sistema de personalização no site de comércio Amazon [7u] demonstra bem as capacidades desta abordagem.

Os motores de pesquisa de uma forma geral passam por três fases até devolverem os resultados ao utilizador. Elas são o crawling, a indexação e a procura. A informação relativa aos interesses de utilizador poderia muito bem ser acoplada em qualquer uma destas fases porem e o custo de implementação seria muito elevado. A forma mais simples e mais abordada [23] [24] [25] é a de adaptar este conhecimento do utilizador aos resultados finais provenientes de um motor (ou meta-motor) de pesquisa, fazendo uma reordenação dos resultados tendo em vista o perfil do utilizador.

Frequentemente as duas soluções não são usadas homogeneamente, muitas das contribuições universitárias quando focam a utilização da categorização para melhoria dos resultados permitindo ao utilizador socorrer-se da escolha da categoria que mais lhe é chamativa, não utilizam a personalização para tentarem reordenar essas categorias e vice-versa. Do nosso ponto de vista a aglomeração destas duas soluções só trás vantagens visto que uma pode auxiliar a outra.

De modo a resolver o problema de imparcialidade em relação ao utilizador e aos seus interesses por parte dos WebIR, pretendemos criar um sistema capaz de categorizar os resultados provenientes de múltiplos motores de pesquisa, melhorando assim a experiencia de utilização ao possibilitar a restrição do numero de documentos face á escolha da categoria mais apelativa.

Com vista em introduzir conhecimento sobre os utilizadores num sistema de recuperação de informação, pretendemos criar automaticamente modelos de utilizadores a partir do armazenamento do seu historial de pesquisas onde para além das queries introduzidas também são guardados os clusters relativos a estas mesmas queries e para as quais documentos foram consultados. Desta forma com o uso destes modelos de utilizador, os sistemas de pesquisa poderão reordenar os resultados e as categorias de modo a colocar em destaque todo o conteúdo que tem mais interesse para o utilizador.

1.5 - Organização da tese

No capítulo 2, serão analisados superficialmente os sistemas de recuperação de informação. Numa primeira fase serão descritas as metodologias dos sistemas IR tradicionais que depois numa segunda fase serão vistos como a base para a criação dos sistemas WebIR, os actuais motores de pesquisa que fazem uso de potentes algoritmos para classificar a informação.

No capítulo 3, são referidas algumas das metodologias envolvidas no processo de categorização. É feita uma análise mais profunda sobre um tipo de categorização mais específica que é a dos Web snippets.

O capítulo 4 aborda todo o processo de construção de um modelo de utilizador. São referidos os processos utilizados que permitem descobrir que informação é importante para o utilizador e o trabalho relacionado com a criação de modelos de utilizador a partir desta informação recolhida.

O capítulo 5 refere-se a todo o trabalho desenvolvido ao longo da investigação onde são descritos os algoritmos utilizados para a criação da categorização e dos modelos de utilizador.

No capítulo 6, são apresentados e discutidos os resultados do trabalho.

Finalmente o capitulo 7 da tese apresenta as conclusões e algum, possível, trabalho futuro.

Capítulo 2

2 - Sistemas de recuperação de informação

Durante muitas décadas as pessoas tornaram-se conscientes da importância de encontrar e armazenar informação. Com o surgimento da era dos computadores, tornou-se possível guardar grandes quantidades de documentos e como tal, encontrar informação útil dentro destas bases de conhecimento tornou-se uma necessidade. O campo dos sistemas de recuperação de informação foi criado em 1945, fruto destas necessidades [8]. Em 1960, Gerard Salton desenvolveu o sistema SMART, protótipo de um sistema IR que serviu de teste a algoritmos que automaticamente indexavam e devolviam documentos de texto. Muita teoria de processamento da linguagem natural corrente estava lá: análises estatísticas de texto, extracção da raiz das palavras, remoção de palavras que não representam conhecimento e teoria da informação, tudo foi experimentado. Como em todos os sistemas desenvolvidos até esta data, os conjuntos de informação analisados pertenciam a um ambiente controlado onde o conteúdo existente era realmente importante e bem estruturado, tinha dimensões grandes mas nada de extraordinário e era bastante estático no sentido de que o conteúdo dos documentos não variava.

O surgimento da Web mudou a situação por completo. A multiplicidade de conteúdos heterogéneos, dinâmicos, objectivos e subjectivos, aliada ao crescimento exponencial das fontes que os disponibilizam, fez com que uma nova realidade fora de controlo emergisse. De modo a existir uma adaptação a esta nova situação, foi criada uma nova disciplina: os sistemas de recuperação de informação na Web (Web Information Retrieval ou WebIR). Esta tem por base muitos conceitos tradicionais de recuperação de informação, mas foi capaz de introduzir muitos mais engenhos inovadores capazes de se adaptarem ao meio e às circunstâncias.

2.1 - Os sistemas tradicionais de IR

Os sistemas de recuperação de informação tradicionais propuseram diversos modelos capazes de representar o conteúdo dos documentos [8]: Booleanos, probabilísticos, de inferência e o modelo de espaço vectorial. Os últimos três são os mais conhecidos devido ao permitirem calcular e atribuir um valor de interesse para cada documento face a uma querie, levando a que no final os resultados a devolver possam ser ordenados por esta qualificação. Os modelos booleanos apesar de também conseguirem ordenar resultados por uma certa característica, não têm esta visão de quantificação da proximidade entre documento e querie. É contudo, o modelo de espaço vectorial "vector space model" que merece mais destaque face á sua maior aceitação por parte dos sistemas IR.

A representação dos documentos no modelo de espaço vectorial [29] é feita do seguinte modo: Os documentos são processados e transformados numa lista de termos. Estes termos, (geralmente palavras) muitas vezes são submetidos a uma filtragem onde são eliminados os que não contém significado ou reduzidos á sua forma canónica. Cada documento ***d*** é representado no espaço Euclidiano sob a forma de um vector e cada uma dos termos ***t*** do conjunto da intersecção entre todos os conjuntos de termos representativos dos documentos, é um eixo desse espaço. Assim, se um termo ***t*** aparece ***n(t,d)*** vezes num documento ***d,*** então a ***t*** coordenada deste é simplesmente ***n(t,d)***. Contudo, apenas o número de vezes que um termo ocorre num documento não pode aludir a conclusões, se não for introduzida informação sobre o tamanho desse mesmo documento. O cálculo da frequência do termo (*TF*) permite fazer isto. $TF = \frac{n(t,d)}{||d||}$ onde $||d||$ é o tamanho do documento ***d*** normalizado.

O *TF* é a relação entre o número de vezes que o termo ***t*** ocorre no documento sobre o número de termos pertencentes ao documento. Infelizmente esta medida não é suficiente para constatar que certos termos tem uma importância superior a outros. Imagine-se, uma colecção de documentos relacionados com a área automóvel. Os termos carro, automóvel, não podem ter a mesma importância que os termos gasolina, ou MG, visto ser muito provável que os primeiros existam em praticamente todos os documentos. Se o termo ***t*** ocorre ***Nt*** vezes num conjunto ***C*** de ***Nc*** documentos, então $\frac{Nt}{Nc}$ mede a raridade do termo ***t*** e como tal a sua importância face ao conjunto ***C***.

O *IDF*, sinonimo de frequência inversa do documento é uma optimização desta métrica. *IDF(t, C)* = $1 + \log \frac{Nt}{Nc}$.

O *TFIDF* é a junção destas duas medidas e consegue quantificar a importância de um termo t num documento d de um conjunto C. É a medida mais utilizada em recuperação de informação para representar o valor de cada ponto nos vectores $\vec{d}$ representativos dos documentos no espaço euclidiano.

$$TFIDF(t,d,C) = \frac{n(t,d)}{||d||} * 1 + \log \frac{Nt}{Nc}$$

De modo a poder calcular uma aproximação da querie aos documentos, é necessário transforma-la também num vector $\vec{q}$ passível de ser adicionado ao espaço euclidiano. Com o uso de métricas para o cálculo de distância entre vectores, é possível indicar e ordenar por grau de relevância, os documentos que mais se aproximam á querie. As mais utilizadas são a distância Euclidiana ou a distância de Cosine.

Distancia Euclideana(d,q) = $\sqrt{\sum_{t \in (termos(d) \cap termos(q))} (q_t - d_t)^2}$

Cosine(d,q) = $\frac{d.q}{\sqrt{\sum_t n(t,d)^2} . \sqrt{\sum_t n(t,q)^2}}$

Graças a estas heurísticas baseadas na geometria, é possível a um sistema IR retornar de forma ordenada os documentos mais relevantes para uma querie introduzida.

Para efectuar uma avaliação quanto ao desempenho de um sistema de recuperação de informação, diferentes medidas foram definidas. De modo a poder contextualizar, tomemos Q como sendo um conjunto constituído por múltiplas queries q {q1, q2, q3…} submetidas a um sistema de WebIR. Manualmente de todo o conjunto de documentos D são seleccionados Dq {Dq1, Dq2, Dq3…} conjuntos de elementos cujos são mais relevantes para as respectivas queries. O sistema IR devolve para cada querie q um conjunto de documentos Dp ordenados por grau de relevância {Dp1, Dp2, Dp3…}. Pode-se calcular uma lista binária R{r1, r2, r3…} representativa da pertença de um documento (proveniente do sistema) ao conjunto de documentos manualmente definidos para a querie onde ri = 1 se Dpi pertence a Dqi ou ri=0 caso contrario.

A ***precisão*** é o número de documentos relevantes dividido pelo número total de documentos recebidos.

$$Precisão\ (\mathrm{k}) = \frac{\sum_{i \leq k} r_i}{k}$$

O ***recall*** define-se como o número de documentos relevantes recebidos dividido pelo número total de documentos relevantes para a querie.

$$Recall\ (\mathrm{k}) = \frac{\sum_{i \leq k} r_i}{|D_q|}$$

Onde ***k*** representa uma qualquer posição da lista de documentos facultados pelo motor de pesquisa no calculo da *precisão* ou *recall*.

Dado que a *precisão* e o *recall* quantificam propriedades diferentes, existe normalmente uma relação custo-benefício entre elas. Se por um lado para melhorar o *recall* podemos aumentar k, por outro, a *precisão* irá provavelmente ser reduzida devido ao facto de possivelmente surgirem mais documentos não relevantes. Uma medida frequentemente utilizada é o *F-measure*, que é capaz de combinar a *precisão* e o *recall* num único valor.

$$F_b = \frac{(b^2 + 1)(P.R)}{b^2.P + R}$$

Esta medida é passível de receber como entrada um valor ***b*** variável que quantifica o grau de interesse da precisão sobre o *recall*. Se ***b*** = 0 então a *F-measure* torna-se a *precisão* **P**, se ***b*** -> infinito, então **F** toma os valores do *recall* **R** para um ***k*** fixo. Quando ***b*** = 1, os valores de **P** e **F** tem o mesmo nível de importância, se ***b*** = 2 então o *recall* tem o dobro da importância. O uso das medidas de *precisão* e *recall* no caso específico de um sistema de recuperação de informação na Web é bastante difícil pois devido à quantidade de documentos existentes ser de tal ordem colossal, a tarefa de manualmente classificar os documentos que são realmente relevantes para uma querie introduzida é humanamente impossível [12].

2.2 - Os sistemas WebIR

Actualmente a Web é a maior fonte de informação do mundo. Noticias, vídeos, musica, fóruns, jogos, informação militar, análises e explicações detalhadas de todo o tipo de material existente á face da Terra, tudo está lá. Difícil mesmo é encontrar o que se pretende numa rede que não para de crescer e de se actualizar. Os modernos sistemas de recuperação de informação na Web permitiram com a exploração de algumas das metodologias de IR clássicas, desenvolver metodologias inovadoras capazes de encontrar informação relevante nesta rede tão complexa. Um estudo recente [9u] mostra que 80% dos utilizadores que navegam na Web encontram os sites, que visitam depois com frequência, a partir dos motores de pesquisa na WEB como o Google [1u], Yahoo [2u], Bing [10u] ou Ask [11u]. Se um utilizador pretende saber algo sobre um local ou uma tecnologia ou simplesmente encontrar o restaurante mais próximo dele, simplesmente introduz uma querie no motor de busca e espera numa fracção de segundos obter as suas respostas.

Tal facto faz com que esta disciplina se tenha estabelecido como uma das mais importantes áreas de pesquisa e interesse da comunidade científica, cujo intuito é desenvolver sistemas capazes de providenciar cada vez melhores resultados. Nos parágrafos que se seguem, a arquitectura geral destes sistemas bem como algumas das suas tecnologias serão abordadas se bem que de uma forma algo leve dado não ser o âmbito desta tese estuda-las profundamente. Para tal propomos uma leitura de [12] [13] [8].

2.2.1 - Estrutura de um motor de pesquisa Web

O processo de execução de um sistema WebIR pode ser visto como um aglomerar de três fases: Extracção de documentos ou páginas Web, indexação e procura.

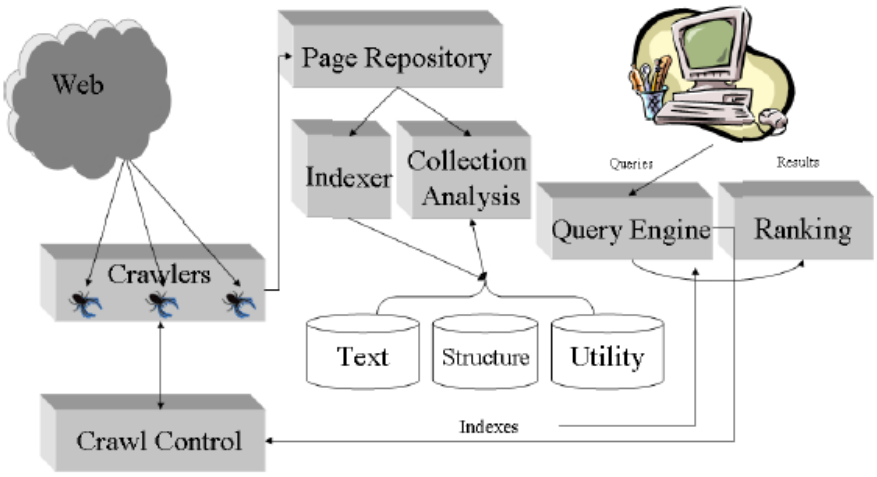

Figura 2.1: Arquitectura de um motor de pesquisa.

A extracção de documentos Web, tarefa talvez mais conhecida por "crawling" passa por percorrer a rede ou parte dela e reunir o conteúdo dos documentos guardando-o num repositório. Tal trabalho só é possível graças à computação distribuída. Este trabalho é feito por agentes controlados que operam de diversas formas conforme os requisitos impostos pela entidade que os controla. Esta pode definir a maneira como os agentes percorrem a Web segundo certas políticas (BSF, DFS, aleatória). Em certos intervalos de tempo, o repositório é processado sendo os documentos analisados e indexados num grafo de forma a permitir um rápido acesso à estrutura e ao conteúdo dos documentos. Neste grafo os documentos são os vértices e as hiperligações as arestas. Esta estrutura é depois analisada tanto do ponto de vista relacional como contextual, permitindo fazer o ranking de cada documento. É na análise da parte relacional que se encontram muitas das inovações face aos tradicionais sistemas de pesquisa. Alem de considerar os termos que o documento contem e a sua importância relativa ao conteúdo do documento (considerar para o efeito como este sendo o conjunto **A**), são considerados também os termos que representam os textos correspondentes às

hiperligações que apontam para este mesmo documento (considerar para o efeito este como sendo o conjunto **B**). Tal deve-se ao facto de muitas vezes as descrições associadas as hiperligações, serem boas representações do conteúdo do documento para as quais se referem. Esta teoria está já bem cimentada como se pode constar [12]. Desta forma a o conteúdo que representa cada documento na estrutura é proveniente da reunião dos conjuntos A e B. Por razoes de eficiência, parte do processo de ranking é feito off-line, antes de a querie ser submetida ao motor de pesquisa.

Actualmente os motores de pesquisa indexam biliões de páginas através do recurso a plataformas de computação distribuída onde os índices da estrutura são replicados por vários clusters de servidores autónomos. Esta distribuição é necessária para sustentar e dar resposta aos milhões de queries introduzidas pelos utilizadores dia após dia.

2.2.2 - Indexação de páginas Web

De modo a efectuar uma pesquisa por conteúdo, o utilizador introduz uma querie. Para seleccionar os documentos que serão devolvidos, o sistema necessita de percorrer todos os documentos existentes, ler o seu conteúdo e verificar se as palavras da querie estão de facto presentes. Quando o estão avalia ainda o seu grau importância no documento, sabendo assim se este vai ser um dos elementos a devolver como resultado. Dada a dimensão da Web, o problema desta técnica é que requer um elevado nível de tempo e poder de processamento para poder disponibilizar os resultados. Para um sistema cujo objectivo é funcionar em tempo-real, a metodologia torna-se impraticável.

Listas invertidas:

O uso de listas invertidas veio solucionar este problema. A ideia é bastante simples: todos os termos extraídos dos documentos são armazenados num léxico. Cada termo deste léxico aponta para uma lista de tamanho variável que contem identificadores dos documentos (opcionalmente também pode conter as posições no texto) onde o termo ocorre. Este léxico contém também muito frequentemente para cada

termo, o valor do IDF ou outras medidas passíveis de indicar um melhor grau de interesse do documento.

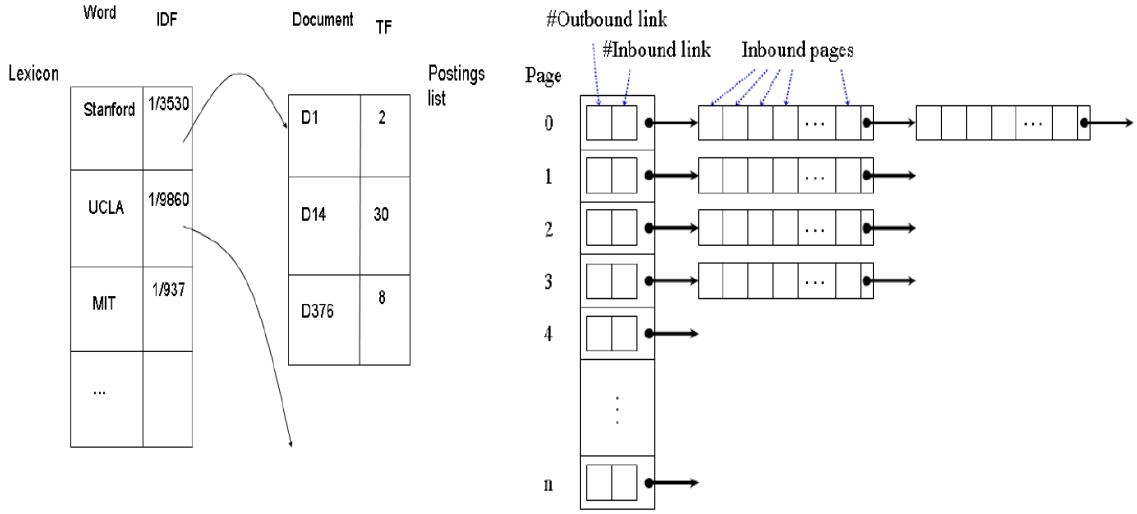

Figura 2.2: Representação de uma lista invertida e de um grafo de indexação.

Desta forma, o número de documentos a analisar é reduzido drasticamente visto que só são contemplados os documentos que estão a ser apontados pelos termos do léxico que compõe a querie.

Indexação da Web em grafos:

Os motores de pesquisa necessitam de armazenar e rapidamente aceder á estrutura de hiperligações presente na Web. A figura 2.1 apresenta um tipo de estrutura que permite esta representação onde para cada página Web existe uma lista com os índices de todas as páginas apontadas pelo documento. Esta representação é denominada por grafo e cada página é um nodo que está ligado ou não a outros nodos por arestas que representam as hiperligações. Diferentes tipos de estruturas, podem ser definidas de acordo com os interesses e objectivos de cada sistema IR. A sua necessidade advém dos novos algoritmos de ranking cujos fundamentos para a classificação se baseiam em redes sociais onde é necessário saber para onde aponta um nodo do grafo e quais os nodos que apontam a ele. A optimização e compressão destes grafos da Web é fonte interesse para muitos investigadores que propõe para isso diversas técnicas como em [14].

2.2.3 - Algoritmos para ranking de páginas Web

O uso de listas invertidas associadas a poderosos grafos para indexação pode resolver os problemas de rapidez mas infelizmente não resolve os problemas de qualidade. Com o número de documentos existentes na rede a ascender aos biliões, as medidas utilizadas na tradicional IR como o cálculo do TFIDF para cada termo não são suficientes, visto não serem capazes de representar sozinhas o real interesse do documento face a todo o enorme conjunto. Diversas metodologias baseadas em redes sociais sobrepuseram-se as medidas tradicionais, como é o caso dos algoritmos PageRank, Hits ou o TrustRank. De todos eles o mais famoso até à data é o PageRank. A próxima secção refere-se a este algoritmo.

PageRank:

O PageRank é o algoritmo utilizado pelo motor de pesquisa Google. Permite classificar a importância de uma página independentemente das queries e baseia-se na estrutura de relações entre páginas (as hiperligações). Em teoria uma página é mais importante se for referenciada por muitas outras páginas também elas importantes. De uma forma superficial iremos explicar a computação deste algoritmo, usando a notação de Brin e Page [12]. Consideremos uma matriz de adjacência (forma de representar um grafo) ***E[u,v]*** associada ao grafo da Web $\boldsymbol{G_w}$. Seja um ***Z[v]*** um numero real positivo representativo da noção de importância de ***v***. Então o vector Z representa a importância de cada um dos nodos de E. A partir da equação $Z = E^T.Z$, o valor da importância de cada vértice é igual à soma da importância de todos os seus parentes. Desta forma a notação da matriz pode ser reescrita por $Z_{i+1}[v] = \sum_u E^T[v,u]Z_i[u] = \sum_u E^T[u,v]Z_i[v]$ O que se pretende não é saber o valor exacto de importância atribuído a cada vértice mas sim a ordem relativa assumida por eles.

De forma a garantir a existência de um ponto fixo para a equação $Z = E^T.Z$ a matriz $\boldsymbol{E}$ tem de ser alterada por forma a satisfazer as condições impostas pelo teorema(A Web não é um). Para tal Larry Page propôs uma solução.

$Z = P^T.Z$ onde $P[u,v] = \frac{E[u,v]}{outdeg(u)}$

As páginas sem hiperligações externas podem ser tratadas de duas maneiras: são removidas de forma a reduzir a complexidade da computação, sendo depois os valores de importância das páginas computadas, propagados pelas removidas. Ou então é efectuado um salto aleatório para uma qualquer página quando se atinge uma sem ligações. Esta solução leva a criação de uma nova matriz $P = P + D$ onde D é definida por:

$$D_{i,j} = \begin{cases} \frac{1}{N} \; se \; outdeg(i) = 0 \\ 0, caso \; contrário \end{cases}$$

Um utilizador ao navegar na Web acaba por parar de saltar de página para página, a proposta de Page baseia-se nisso mesmo impondo um valor probabilístico (1 - α) para efectuar a passagem ao próximo vértice ligado.

Matematicamente, isto corresponde á criação de uma matriz estocástica, irredutível e aperiódica ***P***.

$$P = \; \alpha P + (1 - \alpha)\frac{1}{N} s s^T$$

Onde ***N*** é o número de páginas no grafo da Web e ***s*** é a unidade do vector para que o salto aleatório não seja focado num vértice em particular.

Deste modo, o algoritmo PageRank atribui um valor de importância às páginas ao resolver a equação: $Z^T P = Z^T$.

Mesmo sendo este algoritmo muito poderoso no que toca à ideologia que o representa e estando em constante aperfeiçoamento, a complexidade da rede internet actual, é de tal forma exigente que é necessário explorar outras metodologias de forma a melhorar os resultados devolvidos pelos motores de pesquisa. Os próximos dois capítulos abordam os dois mais importantes conceitos que procuram realizar esta árdua tarefa.

Capítulo 3

3 - Categorização de Web snippets

3.1 - Introdução ao processo de categorização

O termo categorização "clustering" denota um alargado número de metodologias para identificar estruturas comuns escondidas em grandes conjuntos de objectos. Este tipo de metodologias é aplicado nas mais diversas áreas como a física, biologia, estatística, análise numérica, mineração de dados entre muitas outras. Um cluster é um grupo de objectos cujos membros são mais similares entre si, que os elementos dos outros clusters. Pode-se por isso dizer que a similaridade intra-grupo é alta mas a similaridade inter-grupos é baixa. No caso especifico da área de IR, os objectos referidos são os documentos da Web na sua completude ou parcialmente. Salientamos o facto de esta categorização de objectos ser um processo todo automático, não podendo ser confundido com classificação onde existem á partida diversas categorias pré-definidas e depois os documentos são atribuídos à respectiva categoria. Também é de realçar que a categorização que vamos abordar é relativa ao conteúdo Web e como tal foca-se no trabalho com texto. Uma particularidade deste tipo de categorização é que não tem só como objectivo fazer uma boa geração de grupos mas também atribuir a estes grupos um nome que represente bem o seu conteúdo.

3.1.1 - Classificação dos vários métodos de categorização

Os métodos de categorização podem ser classificados atendendo a vários aspectos:

Estrutura:

Plana - termo mais conhecido por "Flat Clustering" onde só existe um primeiro nível de grupos, apresentado como uma lista.

Hierárquica – mais conhecida por "Hierarchical Clustering". Os grupos são organizados numa estrutura em árvore. Os elementos da raiz da árvore, são os grupos de nível 1. Cada um desses grupos pode ainda estar subdividido em mais grupos e assim sucessivamente.

Overlap – Não é representativo da estrutura mas sim de uma particularidade indicativa de que certos elementos podem pertencer a mais que um grupo.

Unidade de indexação:

Os documentos (nesta área em particular) são representados por um conjunto de palavras, vulgarmente referido como saco de palavras ou por um conjunto de frases onde a ordem das palavras é tida em conta.

Duração:

A categorização pode ser efectuada sobre um conjunto de documentos persistente, cujas propriedades (o conteúdo) não variam ao longo do tempo, ou, sobre um conjunto que existe por breves momentos como o caso dos documentos devolvidos por um motor de pesquisa na Web para uma querie fornecida. Denomina-se este último caso por categorização efémera.

Algoritmo:

Tipo de sequência utilizada para efectuar a categorização dos objectos. Esta pode ser por divisão (o processo é iniciado a partir de um conjunto de objectos e efectua-se a subdivisão deste em múltiplos grupos, podendo alguns dos elementos pertencer a diferentes conjuntos) ou por Aglomeração (começa-se individualmente por cada um dos elementos e efectua-se a junção destes em grupos existindo a mesma possibilidade de overlap).

Nos sistemas de recuperação de informação tradicionais o processo de categorização persistente era considerado o mais comum visto "os documentos existirem num ambiente controlado e bastante estático sendo necessário categorizar a informação apenas uma vez e depois manter a estrutura" como refere Salton em [15]. Correntemente tal não acontece pois ainda que existam múltiplas estruturas de

documentos estáticas, a Web não o é e como tal todo o seu conteúdo é muito dinâmico. Os motores de pesquisa na Web por modo a mostrar os resultados em categorias navegáveis muito rapidamente, fazem-se usar da categorização efémera sobre os resultados onde a durabilidade da estrutura é proporcional ao tempo da sessão com o utilizador.

3.1.2 - Noção de similaridade e não similaridade

A ideia chave da categorização é a definição de noção de similaridade entre objectos num grupo. O objectivo principal das técnicas de categorização é a partição dos documentos em k subconjuntos por ordem a maximizar a não similaridade entre os grupos e maximizar a similaridade dentro destes. Seguindo a notação de [16], podemos dizer que a função ***d*** é o índice de não similaridade entre um conjunto Ω de documentos, se satisfaz as seguintes propriedades:

1. $d(d_i, d_i) = 0, \forall\, d_i \in \Omega$
2. $d(d_i, d_i) = d(d_j, d_i), \forall\, (d_i, d_i) \in \Omega * \Omega$

De maneira fácil, dada a medida de não similaridade ***d***, é possível calcular a similaridade ***s***: $s(d_i, d_i) = 1 - d(d_i, d_j), \forall\, (d_i, d_i) \in \Omega * \Omega$

As medidas mais utilizadas para o cálculo de similaridade entre dois documentos é o coeficiente de *Cosine* (C) entre os vectores representativos destes documentos após terem sido normalizados e o coeficiente de *Jaccard* (J).

Cosine: $s(\overrightarrow{d_i}, \overrightarrow{d_j}) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{d_j}}{\|\overrightarrow{d_i}\| \cdot \|\overrightarrow{d_j}\|}$

Jaccard: $s(d_i, d_i) = \frac{|T(d_i) \cap T(d_j)|}{|T(d_i) \cup T(d_j)|}$

Onde $T(d_i)$ é o conjunto de átomos que representam o documento d_i. Na categorização clássica esses átomos são geralmente as palavras que representam o documento, formando o tão conhecido "saco de palavras". No caso mais específico da categorização dos snippets da Web, a reduzida dimensão do conteúdo obriga a que também seja muitas vezes necessário utilizar sequências de palavras como forma de átomos e outro tipo de medidas por modo a ser possível a criação de categorias realmente fortes e valiosas somente a partir de tão pouca informação. Mais para a frente este assunto será abordado com mais detalhe visto ser o tipo de categorização utilizado no nosso trabalho.

3.1.3 - Descrição de algumas abordagens existentes para categorização automática

A aplicação do processo de categorização é comum a diversas ciências. Como tal ao longo do tempo surgiram as mais variadíssimas metodologias específicas às diversas áreas e com o intuito de obter os melhores resultados. Focando-nos no processamento da linguagem natural, existem diversas abordagens: particionais, geométricas, sintácticas, probabilísticas entre outras, mas são contudo as particionais e as geométricas as mais se destacam.

As abordagens particionais podem-se dividir em dois tipos: "Bottom-Up" onde as categorias são obtidas a partir de um conjunto inicial de documentos que depois vai sendo dividido até que se encontre um ponto de estabilidade. "Top-Down" quando existe um número pré-definido de categorias e os documentos vão sendo atribuídos á categoria cujos documentos já existentes nela se aproximam mais do documento que procura adicionar-se a um grupo. Nos próximos parágrafos serão abordados dois algoritmos "Hierarchical" e "K-means" muitos famosos neste meio e que pertencem a cada um destes tipos respectivamente.

Hierarchical clustering:

A categorização hierárquica agrupa os objectos numa estrutura em forma de árvore. A técnica mais frequentemente utilizada neste tipo de categorização pertencente ao tipo "Bottom-Up" das abordagens particionais é o Hierarchical Agglomerative Clustering (HAC). Esta técnica consiste na criação de uma sequência de partições do conjunto original Ω de documentos criada com base no pseudo-código apresentado na figura 3.1. Cada grupo criado é resultante da união dos dois conjuntos que estavam no nível anterior.

```
Let Ω the group of documents organized in singletons {d};
while  ((|Ω| > 1) AND (merged == 1)){
   merged = 0;
   choose (δ, γ)  ∈  Ω × Ω to maximize the similarity measure s(δ, γ);
   if exist (δ, γ) {
      remove (δ, γ) from Ω × Ω;
      add θ = δ ⋃ γ to Ω;
      merged = 1;
   }|
}
```

Figura 3.1: Pseudo-código do algoritmo HAC.

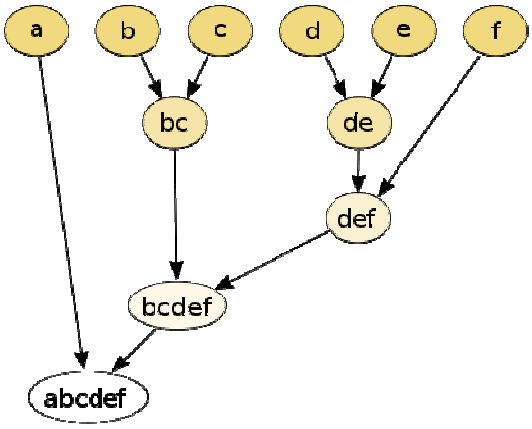

Figura 3.2: Árvore representativa das iterações do algoritmo HAC sobre um conjunto de objectos.

Para saber qual o melhor grupo a juntar em cada ciclo, o HAC necessita da definição de uma medida de similaridade entre grupos. Esta similaridade pode ser definida como:

Single Link: A similaridade entre dois grupos é dada pelo par de documentos cujo apresenta a melhor similaridade do conjunto.

Complete Link: A similaridade entre dois grupos é dada pelo par de documentos cujo apresenta a pior similaridade do conjunto.

Average Link: É unido o par de grupos que a cada iteração tenha a maior coesão. Para cada grupo $\boldsymbol{G}$ a coesão $\boldsymbol{C}$ calcula-se da seguinte forma: $C(G) = \frac{1}{n(n-1)} (R(G) - n)$ onde $n = |C|$, $R(G) = \sum v \in G \sum w \in G < v, w >$ em que $< v, w >$ é o produto escalar de v com w.

Minimum Variance: O método da variância mínima junta o par de grupos cuja união resultará no incremento mínimo do valor do erro quadrático dentro do novo grupo.

A categorização hierárquica é geralmente a mais acolhida pela comunidade pois permite navegar pela árvore construída e a cada nível de profundidade é possível encontrar um conjunto de documentos cada vez mais específico.

K-Means:

O K-Means é um algoritmo que necessita de um parâmetro de entrada (para alem dos documentos) relativo ao número de categorias a gerar. Funciona sobre o espaço Euclidiano onde todos os documentos são representados sob a forma de um vector cujos pontos contem a relação com todos os termos existentes (Vector Space Model). Por cada grupo é calculado um documento virtual designado de centroide que representa todo o conjunto de documentos pertencentes ao grupo. Percorre-se novamente o conjunto inicial de todos os documentos e calcula-se a distância de cada um aos centroides. O documento é adicionado ao grupo cujo centroide está mais próximo deste. Depois de todos os documentos terem sido recolocados, calculam-se novamente os centroides e o algoritmo só termina quando não tiverem existido alterações quanto a posição dos centroides ou com um outro tipo de regra imposta (o numero de iterações por ex.). Se a regra imposta não for cumprida, repete-se o processo.

A computação dos centroides pode ser feita de diversas formas mas a mais comum é a criação de um vector que é a representação geométrica da média do conjunto dos termos dos documentos que existem no grupo. A complexidade deste algoritmo é proporcional ao número de documentos (***d***) e ao número de grupos (***g***) pré estabelecido e visto que cada documento tem de ser equiparado com cada grupo, calcula-se como sendo ***d.g***.

As abordagens geométricas representam os documentos num enorme espaço Euclidiano sendo depois aglomeradas certas dimensões (termos) cujas ocorrem muito frequentemente num certo conjunto de documentos, gerando-se assim as categorias. Exemplos de algoritmos que se enquadram nesta metodologia: Self Organizing Maps (SOM), Multidimensional Scaling, Fast Map e Latent Semantic Indexing.

3.2 Categorização de Web snippets

No decorrer deste capítulo temos vindo a analisar o processo de categorização, mais concretamente, a categorização de documentos da Web. Contudo existe ainda um tipo de categorização mais específico, de Web snippets. Neste caso, o conteúdo original de uma página Web é substituído por um conjunto de expressões representativas do documento, sendo esse conjunto de texto muito mais reduzido, dado como entrada para o algoritmo de categorização. A categorização de snippets foi primeiramente introduzida no sistema Northernlight (funcionava mais como um classificador) mas tornou-se famosa com o motor de pesquisa Vivíssimo. Tudo o que se pretende de um sistema de categorização de documentos Web, pretende-se também na categorização de snippets. Contudo, as dificuldades deste tipo de categorização são ainda mais acentuadas visto que o conjunto de informação com que se trabalha é muito mais reduzido.

O termo snippet pode não ser muito conhecido para quem não seja da área, porém praticamente todas as pessoas que já utilizaram um motor de pesquisa, já olharam para conteúdo deste tipo. O snippet é todo o conjunto de expressões que existem no título e descrição, associados a um url. Ou seja, é a informação associada a cada url existente na lista de resultados apresentada, para uma qualquer querie introduzida, em um qualquer motor de pesquisa.

A vantagem associada ao uso dos snippets para efectuar o processo de categorização prende-se fundamentalmente com a rapidez de execução do algoritmo. Qualquer sistema que faça uso dos resultados (sem categorização) de um motor de pesquisa e pretenda efectuar a devida categorização, recebe normalmente estes snippets numa lista ordenada por relevância. O sistema pode então efectuar duas coisas: carregar para cada url o contudo completo da página que este representa ou, trabalhar directamente com esta informação mais reduzida. O primeiro caso, implica que sejam descarregados da Web o contudo de cerca de 100 a 500 páginas para só depois ser efectuada a categorização. Mesmo que já exista uma grande quantidade de páginas indexadas, a quantidade de informação, torna o sistema demasiado lento para poder trabalhar em tempo real e satisfazer milhares de queries por dia.

O trabalho com os snippets permite a criação de sistemas capazes de categorização em tempo real, contudo e como já referido a informação é muito menor, existindo dificuldade acrescida na criação de bons clusters e boas expressões identificadoras do conteúdo destes.

Para resolver os problemas inerentes a este tipo de categorização, muitas das abordagens existentes passam não só pelo calculo da similaridade entre o conteúdo dos documentos para depois formar grupos (document-derived clustering), mas também através da valorização de termos ou expressões compostas que façam parte do conjunto de snippets (label-derived clustering, como referido em [34]). Estes termos mais fortes servem depois para aglomerar documentos que as contenham, gerando assim clusters. O algoritmo para categorização de snippets proposto nesta tese segue este segundo tipo de metodologia.

3.3 - Trabalho relacionado

O trabalho relacionado pode ser dividido em duas categorias: comercial e académico. Tal acontece porque a categorização de Web snippets já é uma metodologia bastante implementada no mundo comercial e apesar de muitas ideias surgirem do mundo académico, é importante realçar o esforço que algumas entidades fazem em explorar este potencial e disponibiliza-lo aos utilizadores.

No mundo comercial o primeiro sistema a efectuar a categorização de snippets foi o Northernlight, cujo a cada documento era atribuído um conjunto de tags identificadores de categorias que pudessem estar associadas a esse documento. Esta atribuição era feita manualmente, tornando este sistema mais como um classificador. Os resultados eram devolvidos e consoante as tags associadas a cada documento, também eram apresentadas as categorias que representavam esses documentos. O sistema foi desactivado em 2002.

Desde esta data, diversos motores de categorização de Web snippets foram surgindo. Funcionam na sua globalidade como meta-motores de busca visto que os resultados relevantes para uma querie são obtidos a partir de outros motores de pesquisa como o Google, o Live, Yahoo entre outros. Entre os mais conhecidos estão o Kartoo

[15u], Mooter [13u], iboogie [14u], Vivíssimo [6u] e Carrot2 [12u]. É contudo o Vivíssimo que apresenta melhores resultados tendo até sido distinguido com diversos prémios na área. Infelizmente pouco se conhece sobre a metodologia implementada neste sistema que a cada dia que passa apresenta resultados melhores.

O Carro2, é também um dos meta-motores de pesquisa com categorização que apresenta resultados relevantes e de uma forma visualmente atractiva (quando se visualizam as categorias no modo gráfico). Possibilita a geração dos clusters com base na escolha entre um algoritmo de árvore de sufixos ou com o uso do Lingo [35], um algoritmo que também efectua a geração de clusters com base na descoberta das frases mais fortes.

No mundo académico, são muitas as propostas existentes que visam conseguir o melhor processo de categorização tendo em conta um ou vários factores mais importantes como a qualidade, o tempo de execução, a dependência de fontes externas de conhecimento entre outros. Segue-se uma análise a algumas das abordagens mais bem sucedidas.

Ferragina P. e Gulli A. [10] propõem um sistema de categorização hierárquica de snippets denominado Snaket. Este tem como base para a geração de clusters a classificação de termos e expressões compostas existentes nos snippets. Por modo a ultrapassar o problema da normalmente pouca informação existente nos snipptes, fazem o enriquecimento destes com o uso de duas bases de conhecimento.

Estas bases de conhecimento são construídas offline e são utilizadas em conjunto com o *TFIDF* no processo de atribuição de um valor de importância aos termos. Uma, guarda os títulos das hiperligações extraídos a partir de mais de 200 milhões de páginas Web e a outra indexa o motor de busca *DMOZ* que classifica 3,500,000 sites em mais de 460,000 categorias (é com o auxilio desta base de dados que é dado um valor de importância aos termos, consoante a sua ocorrência nas diversas categorias).

Ao adicionarem informação a cada snippet, estes passam a ser mais completos visto geralmente os títulos das hiperligações serem bons descritores do conteúdo dos sites para onde apontam. Contudo ao fazerem uso das bases de conhecimento, estão a limitar área de acção visto que não podem obter informação para todos os documentos existentes na Web.

Após este enriquecimento dos snippets, são extraídas aquilo que os autores chamam de "gapped sentences". Expressões compostas por termos não consecutivos. Ou seja fizeram uso de um algoritmo capaz de analisar as frases associadas aos snippets e extrair expressões cujos termos por entre o seu conteúdo não tem de ser exactamente os mesmos. Como exemplo muito simples, "George Bush", "George W. Bush" são expressões que na totalidade dos seus termos não são iguais mas o algoritmo reconhece como sendo referentes à mesma coisa.

No final, a cada url é associado, se possível, um conjunto de expressões compostas não consecutivas. A geração dos clusters parte destas expressões. Snippets que partilham as mesmas "gapped sentences" candidatas, falam de um mesmo tema e portanto são agrupados.

Para a geração de uma estrutura hierárquica, são extraídos do conjunto de urls de cada grupo expressões compostas secundárias que ocorram em 80% dos membros de cada grupo. Passa assim, cada conjunto a ser representado por uma expressão composta primária, seguindo-se de um conjunto de outras expressões compostas que representam um conjunto de documentos mais específico.

Estando a estrutura hierárquica de pais e filhos construída, é feito o ranking dos documentos de cada cluster com base no ranking das suas expressões.

O sistema Snaket ambiciona ainda ser um sistema personalizado e como tal, permite que o utilizador após receber a estrutura hierárquica das categorias associadas à querie pesquisada, possa seleccionar um conjunto de títulos para assim filtrar a lista de documentos devolvidos cujo conteúdo pertença somente a esses títulos.

Salienta-se ainda que este foi um dos trabalhos que foi fonte de inspiração ao trabalho proposto nesta tese. Tal facto deve-se ao agrupamento dos urls com base na força das suas expressões e não na simples similaridade entre termos.

Geraci F., Pellegrini e restantes [33], efectuam também eles a categorização de Web snippets com o seu sistema Armil (http://ubi8.imc.pi.cnr.it). Este é um sistema com o intuito de ser muito rápido e como para tal, é apenas dependente do conteúdo dos snippets. A obtenção dos documentos relevantes para uma querie é feita a partir de um meta-motor de pesquisa. Ao contrário de Ferragina que propõe a criação dos clusters com base na força das expressões sendo a sua descrição já efectuada á priori, os autores tomam como sendo mais importante para eles primeiramente a fase de agrupamento dos

documentos e só depois a fase de procura dos melhores descritores do conteúdo de cada cluster.

O agrupamento dos documentos tendo por base os snippets, é feito através do uso de um algoritmo "furthest-point-first" (FPF) para um k número de clusters. Os snippets são por isso representados num modelo de espaço vectorial e é calculada a sua similaridade com base numa métrica não referida pelos autores. É depois aplicado sobre estes documentos uma versão modificada do FPF denominada M-FPF que para um k dado, constrói k conjuntos de documentos. Cada documento só pode existir num cluster, classificando-se assim o algoritmo com "non-overlap".

A escolha dos nomes representativos dos clusters é feita com base numa medida IG_m(Information Gain Measure) atribuída a cada termo que maximiza a existência desse termo dentro do cluster e desvaloriza conforme a sua existência em outros grupos. A expressão atribuída como descritiva do grupo é não um destes termos, mas sim, uma expressão retirada de um dos títulos que exista no conjunto e que contenha um ou mais dos melhores termos.

Lipai [32], apresenta a construção de uma interface completa que permite a categorização dos resultados com base num algoritmo k-means. Contudo para a geração das categorias, todo o conteúdo dos documentos é considerado, sendo por isso o algoritmo algo exigente em termos de tempo e processamento. Os documentos são representados no modelo de espaço vectorial, sendo depois calculados os centroides correspondentes a número de clusters definido previamente. Após sucessivas iterações, obtêm-se os conjuntos finais que são associados a um identificador, também este simples, resultante da escolha dos termos com mais peso.

Também Zamiz e Etzioni [6] propõem a categorização de snippets mas com recurso a metodologias bastante diferentes. Os clusters são gerados através de um algoritmo de árvore de sufixos denominado "Suffix Tree Clustering" (STC) que possibilita a sua construção com base nas frases comuns a cada documento. Uma frase segundo o contexto dos autores, é uma sequência ordenada de uma ou mais palavras.

O algoritmo processa-se em três fases: Limpeza do documento, identificação dos clusters base e combinação de clusters base.

Na primeira fase, é efectuada a remoção de certas tags e caracteres especiais, bem como é aplicado o processo de stemming a cada palavra.

A segunda fase, passa pela criação da árvore representativa das frases e da ligação entre as palavras. Esta árvore apresenta as seguintes propriedades:

- A árvore é enraizada e direccionada.
- Cada nodo interno contém pelo menos 2 filhos.
- Cada aresta é nomeada com uma sub-string de S.
- Nenhuma das arestas que ligam a cada nodo, podem conter palavras que estejam nas arestas acima do nodo.

Onde S, é uma string representativa das palavras de uma frase.

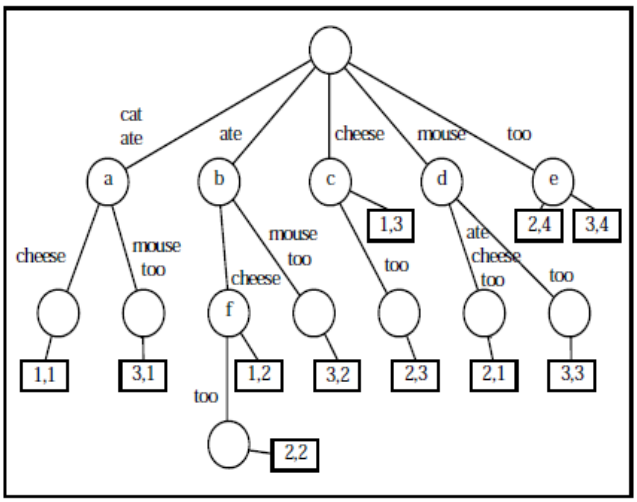

Figura 3.3: A árvore de sufixos para as strings: “cat ate cheese”, “mouse ate cheese” e “cat ate mouse too”.

Cada nodo na árvore representa um conjunto de documentos e uma frase comum a todos eles. É por isso considerado que cada nodo na árvore é um cluster base. A cada um destes clusters é atribuído um valor de importância consoante o número de documentos que contem e as palavras que compõe a frase.

Na última fase, os clusters base são aglomerados conforme uma função definida que agrupa conjuntos cuja grande parte dos documentos que pertencem a cada um, esteja presente em ambos os conjuntos. A escolha dos clusters finais é feita através da selecção dos 10 melhores classificados.

Capítulo 4

4 - Modelos de utilizador para personalização dos resultados de um motor de pesquisa na Web.

O constante e alucinante crescimento da informação disponível na internet, impõe altos requisitos no que toca a conceitos e tecnologias que proporcionem aos utilizadores uma filtragem relevante dessa informação por forma satisfazer os seus interesses. Ao longo do capítulo 2, foram abordados os sistemas de recuperação de informação na Web, motores de pesquisa que permitem através de um conjunto de metodologias actualmente muito baseadas em redes sociais, devolver informação relativa a um tópico requerido pelo utilizador através da introdução de uma querie. Estes sistemas conseguem de facto devolver documentos cuja relação com a querie não se questiona. Porém também que é facto as queries são muitas vezes ambíguas e isso aliado ao facto de serem devolvidos muitas vezes milhares de resultados relacionados faz com que o utilizador diversas vezes não encontre nas primeiras paginas aquilo que pretende. De forma a tentar suavizar esta lacuna, o processo de categorização destes resultados foi implementado em muitos sistemas académicos e também a uma já considerável fatia de motores de pesquisa comerciais. É assim extraído conhecimento oculto por entre os resultados, que é colocado de forma visível ao utilizador, tendo apenas este de navegar na estrutura para o tópico que lhe mais é chamativo. A geração e aplicação de categorização automática nos documentos Web, abordada no capítulo 3, apesar de trazer grandes vantagens ao utilizador apresenta ainda alguns problemas. Muitas vezes, os tópicos apresentados não são suficientes fortes para que o utilizador possa saber exactamente qual a categoria que pretende. Ao navegar por esta estrutura acaba por perder o mesmo tempo que a navegar pela lista de resultados inicial. A consciência deste facto levou a ciência a adaptar novamente uma metodologia já existente noutras áreas, o uso de modelos de utilizador, para a personalização dos resultados dos motores de pesquisa de forma a dar relevo aos que mais vão de encontro ao perfil do utilizador.

4.1 - O que é um modelo de utilizador?

"Todos diferentes, todos iguais" é uma frase bem passível de ser adaptada á filosofia dos utilizadores de um sistema WebIR. Todos tem as suas convicções, interesses e objectivos o que os torna diferentes mas quando frente a um motor de pesquisa, todos aspiram a encontrar a informação que mais lhes convêm de forma rápida e concisa. Os modelos de utilizador focam-se na primeira parte da frase. Cada um de nós sem excepção tem preferências, gostos, hábitos, que em toda a sua extensão fazem com que tenhamos comportamentos diferentes. Um modelo de utilizador pretende mapear estas características únicas de cada um.

Casos muito frequentes de utilização de modelos de utilizador, estão presentes no nosso quotidiano. Quando vamos a um médico pela primeira vez, ele faz-nos um inquérito sobre o nosso historial de doenças, doenças na família, sintomas actuais, registando estes dados numa possível ficha clínica ou base de dados do computador. Este registo permite depois numa outra consulta, saber quem eu sou sem ter de me estar a pedir novamente essas informações. Este registo é um modelo ou perfil de utilizador.

Um modelo de utilizador pode não ser só referente a uma individualidade. Neste caso o modelo serve para abranger um conjunto de pessoas que tem um interesse mútuo em certas áreas. Existem actualmente algumas implementações de metodologias de procura personalizada [22] a fazem uso deste tipo de modelos que ao não serem muito detalhados ou específicos, permitem enquadrar certos grupos de pessoas. Uma outra área que pretende optimizar a eficácia dos motores de pesquisa e faz uso da mutualidade de interesses entre utilizadores é a pesquisa colaborativa. Se uma pessoa A tem os mesmos interesses que outra B e está a procurar informação relativa a assuntos (expressos sobre a forma de uma querie) que já foram procurados por B e B sabe de alguns documentos D relevantes sobre esse mesmo assunto, então é possível indicar a A que os documentos D provavelmente lhe interessarão.

4.2 - O uso de modelos de utilizador para a personalização dos resultados de um sistema IR

A adaptação dos modelos de utilizador aos sistemas de WebIR, tem como objectivo, introduzir conhecimento nos resultados devolvidos. Sabendo aquilo que o utilizador é, pode reordenar a listagem inicial de documentos similares com a querie por modo a colocar no topo da lista os resultados mais próximos dos interesses do utilizador. Imaginemos dois utilizadores distintos. Ambos introduzem num motor de pesquisa a querie "madonna". Um dos utilizadores tem por hábito ir a concertos musicais e outra tem interesse em saber as líricas de diversas músicas. Com base nos seus perfis, onde estão presentes estas particularidades, é possível reordenar os resultados por base no cálculo de uma distância desses resultados aos interesses de cada utilizador.

4.3 - Construção de um modelo de utilizador

Muita pesquisa foi feita para reunir e reconhecer os interesses de um utilizador. Os sistemas construídos para efectuar tal tarefa podem recorrer a indicadores de carácter implícito ou explícito. Explicito, quando os utilizadores indicam ao sistema quais os documentos ou outro tipo de informação lhe é de interesse ou não. Implícito, se de alguma forma é extraído conhecimento das acções do utilizador sem que este se aperceba disso.

No exemplo dado anteriormente referente ao questionário efectuado por um médico para a construção de um perfil do utilizador, o utilizador provavelmente não se mostrará relutante quanto ao facto de ter de responder às perguntas. No contexto da Web o caso é bastante diferente pois a maior parte das pessoas quando lhes é proposta a participação em algum tipo de inquérito não está disposta a perder tempo com isso. É por isso necessário encontrar uma metodologia que seja capaz de indicar automaticamente que tipos de documentos são ou não relevantes para um utilizador. Quando estes documentos são encontrados, é possível analisa-los e extrair informação que sendo comum a um conjunto de documentos ou outro tipo de informação, representa os prováveis interesses do utilizador. Imaginemos uma pessoa que consulte

diversas páginas sobre notícias relativas ao mundo do desporto e uma grande fatia dessas páginas é relativa a futebol. Ao analisar estes dados, será muito provável que os termos desporto e futebol tenham uma grande frequência por entre o conjunto de documentos, o que realmente é representativo dos interesses do utilizador. De forma a extrair este tipo de conhecimento dos documentos lidos ou de um histórico de queries pesquisadas por um utilizador, são utilizadas diversas metodologias que serão descritas no sub-capítulo Trabalho Relacionado. Muitas destas metodologias tem por base o uso do processo de categorização e como este já foi estudado no capítulo 3, não faz sentido estar novamente a entrar em detalhes nesta secção.

4.3.1 – Reconhecimento automático da informação relevante para um utilizador

Reconhecer a informação relevante para o utilizador automaticamente, implica fazer uma análise aos seus padrões de interacção com o sistema. Só desta forma é possível reconhecer que informação é relevante para ele. Para tal é necessário a criação de uma ferramenta que lhe permita a realização normal das suas tarefas mas que consiga também efectuar um registo das suas acções. No caso concreto da análise do comportamento do utilizador dentro do mundo Web, é necessário um browser que seja capaz de permitir ao utilizador navegar para qualquer página e efectuar as suas pesquisas mas, que também consiga indicar que documentos o utilizador leu e achou relevantes. Saber que pesquisas efectuou com resultados úteis, é também muito importante para a obtenção de dados que permitam depois a construção de um modelo de utilizador.

Por modo a saber que documentos e consequentemente, pesquisas, foram relevantes, existem muitas variáveis que podem ser tidas em conta:

- O tempo que decorre desde a abertura de um documento ao seu fecho.

- A movimentação do cursor sobre um documento.

- O conteúdo seleccionado num documento.

- O scroll efectuado no documento.

- O numero de cliques dado no rato.

- A impressão do documento.

- A adição aos favoritos do documento.

Ao relacionar os valores de algumas destas variáveis é possível saber que documentos foram de facto interessantes (ou não) para o utilizador. De salientar o facto que uma delas usualmente é muito mais representativa do interesse de um utilizador num determinado documento. Esta variável é o tempo. Como se poderá constar na abordagem a alguns dos estudos referentes ao assunto apresentados de seguida, todos referem o tempo como sendo o principal indicador do interesse de um utilizador num documento.

4.3.2 - Estudos sobre os melhores indicadores para encontrar documentos relevantes

Os próximos parágrafos referem diversos estudos cujo intuito foi de verificar quais os melhores indicadores implícitos que permitem reconhecer o que é interessante ou não, para o utilizador.

Morita e Shinoda [17], focam-se nos documentos do Usenet News Articles e na investigação com o intuito de demonstrar o interesse dos utilizadores nestes mesmos artigos. Concluem que apenas o tempo disponibilizado pelo leitor na leitura dos documentos é um indicador desse interesse. Salienta-se o facto dos resultados apresentados mostrarem que os tempos mais indicadores de documentos não relevantes (menor tempo gasto), quando confrontados com a leitura percentual do documento, não representam uma leitura total (indicada pelo scroll ate ao final do documento) o que é algo intrigante. Deixa de o ser quando é referido um conceito bastante interessante: Normalmente um utilizador consegue saber se um documento é interessante ou não pela leitura do texto presente nos primeiros parágrafos, não necessitando de observar todo o documento quando este não é relevante.

Kim, Oard e Romanik [18], analisam a relevância do tempo de leitura de um documento como forma de previsão para preferência de leitura de artigos de jornais profissionais ou académicos. Também é estudada a relação entre o tempo dispendido a consultar um documento e a acção de impressão desse mesmo documento. Concluem que os utilizadores tendem a perder mais tempo na leitura dos documentos relevantes para o estudo efectuado do que os outros documentos. Porém é referido que este tempo varia consoante o tipo de artigo, se noticioso, académico ou de um jornal profissional.

Claypool, Le, Waseda e Brown [19], estudam a correlação entre a indicação explícita de interesse e diversos factores implícitos depreendidos do comportamento de navegação do utilizador num browser denominado Curious Browser. São tomados em conta o tempo total dispendido na visualização de um documento, o tempo perdido a mover o rato, o tempo de scroll e o número de cliques nesse documento. Neste estudo é mostrado que o tempo perdido na leitura e o tempo de scroll são bons indicadores de interesse, o número de cliques no rato não é um bom indicador e que existe uma boa correlação entre o tempo de movimentação do rato e o valor de importância dado ao documento. Contudo, os movimentos do rato sozinhos parecem ser só relevantes para a determinação dos documentos que tem menos importância para o utilizador. O contrário não acontece.

Goecks e Shavlik [20], desenharam uma rede neuronal para aprender os interesses do utilizador a partir dos cliques no rato, do scroll pelos documentos e da movimentação do rato. Concluíram apenas que a monitorização da actividade do rato não era suficiente para existir uma correlação com o interesse do utilizador em algum documento.

Chan [21], introduziu algumas métricas para estimar o interesse para cada página consultada. O interesse de um utilizador numa determinada página é calculado através de uma função f que recebe 5 parâmetros de entrada:

1 - Frequência de visitas a essa mesma página.

2 - Booleano a indicar se a página foi marcada como sendo favorita.

3 - O tempo gasto na página normalizado pelo seu tamanho.

4 - O tempo passado desde que a página foi visitada pela ultima vez.

5 – O número de hiperligações visitadas sobre o número de hiperligações existente.

4.3.3 - Trabalho relacionado

Aktas, Nacar e Menczer [22] propõem a criação de um sistema de personalização baseado na alteração do algoritmo PageRank (abordado no capitulo 2). Este passa por introduzir no cálculo do algoritmo informação relativa aos interesses do utilizador por certos domínios específicos. Focam-se por isso especificamente na análise das características do url de cada página.

Na sua implementação foram escolhidas nove categorias das quais três são geográficas e as restantes cinco relativas aos tópicos comercial (.com), militar (.gov), organizações sem fins lucrativos (.org), organizações da rede (.net), educacional (.edu).

Os utilizadores especificam os seus interesses sob a forma de um vector binário correspondente ao interesse ou não numa determinada categoria. Dado um vector de entrada, o sistema computa o valor PageRank para cada página baseado na comparação do domínio do url desta com os do vector representativo do perfil.

Visto existirem nove categorias, existe a possibilidade de existirem 2^9 perfis diferentes que são computados off-line. Quando o utilizador utiliza o sistema, indica em que categorias está interessado e o perfil daí resultante é utilizado para a escolha do PageRank personalizado já calculado que depois irá proporcionar a selecção dos documentos mais relevantes para a querie do utilizador.

Tamine, Boughanem, Zemirli [23], inferem os interesses do utilizador a partir do histórico das suas procuras. No seu ponto de vista, um perfil de utilizador expressa os seus interesses de longo tempo e estes estão presentes no histórico das suas pesquisas sobre a forma das palavras mais utilizadas por entre os documentos considerados relevantes para cada pesquisa feita. A construção do perfil é um processo de duas etapas: Primeiro existe um espaço de tempo em que é armazenada alguma informação. Após este espaço de tempo, é possível inferir os interesses a partir de uma análise estatística dos termos mais frequentes entre os documentos e as queries. A segunda

etapa, consiste na monitorização de uma possível actualização do perfil através de uma comparação entre os termos das queries que vão sendo introduzidas e dos termos que representam o perfil. Quando existe um afastamento considerável, o perfil é actualizado.

Masayasu Atsumi [24] recorre a uma metodologia completamente diferente para extrair os interesses do utilizador a partir das páginas Web que este consulte. Faz o uso de Algoritmos Genéticos. Representa um conjunto de documentos como um vector de palavras representando no vector space model [29] onde a cada termo é atribuída a frequência no documento. O interesse é representado como um cromossoma de tamanho variável cujos genes são pares dos termos e dos seus valores. A extracção dos interesses do utilizador pode ser vista como uma procura genética no espaço de documentos por um cromossoma óptimo, capaz de aproximar todos os documentos ao julgamento de uma porção destes por parte do utilizador.

Gulli e Ferragina [10], autores já referidos no capitulo anterior, pela sua proposta de um algoritmo de Clustering de Web snippets, propõe um tipo de personalização que não faz uso de registo de documentos relevantes para o utilizador e consequentemente, que se afasta do actual problema de privacidade. A sua forma de personalização passa por permitir ao utilizador uma selecção dos tópicos existentes na árvore de conceitos gerada no processo de Clustering por forma, a gerar uma nova lista (ordenada por relevância) de resultados cujos documentos estão mais associados a esses tópicos. Desta forma, é permitido ao utilizador adaptar a escolha dos tópicos de interesse de acordo com a sua subjectividade e interesses temporais dependentes.

Sieg, Mobasher e Burke [25] propõem uma procura na Web personalizada através do uso de ontologias como representação dos interesses do utilizador. Cada ontologia do perfil de utilizador é inicialmente uma instância de uma ontologia de referência constituída por múltiplos conceitos. A cada conceito no perfil do utilizador é atribuído um valor de interesse que inicialmente é 1. À medida que o utilizador vai interagindo com o sistema através da selecção ou visualização de documentos, a ontologia deste é actualizada e os valores para o grau de interesse de cada conceito são

modificados através do desencadeamento de uma acção de propagação. Desta forma, o contexto com o utilizador é garantido e actualizado de acordo com o comportamento deste. A informação sobre os verdadeiros interesses do utilizador é colectada com a intervenção mínima nos padrões de navegação deste. Recorrem para isso a uma observação passiva do seu comportamento: a frequência de visita a uma determinada página, o tempo dispendido na sua leitura, acções de impressão ou adição aos favoritos.

A escolha dos conceitos para a criação da ontologia de referência ficou-se no uso do Open Directory Project, que é organizado numa hierarquia de tópicos e paginas relacionadas com esses mesmos tópicos. A ligação de termos a conceitos na ontologia, foi feita a partir do cálculo dos termos mais relevantes existentes nas páginas de cada tópico tendo por base a medida TFIDF. Os resultados provenientes da procura, são devolvidos ao utilizador por ordem crescente de um valor de importância dado a cada documento. Este valor é calculado com base da multiplicação de outros 3 valores: a distância do documento à querie (Cosine), o valor de interesse para o melhor conceito encontrado para o documento e o valor da distância entre o conceito e a querie.

Singh, Murthy e Gonsalves [26] propõe não um sistema de procura personalizado, mas sim, uma forma de encontrar o interesse do utilizador em tempo real. O objectivo passa por descobrir para uma determinada sessão de pesquisa os termos (palavras) de interesse para o utilizador, baseando-se numa análise aos snippets devolvidos e na supervisão dos documentos consultados pelo utilizador. Os autores referem que, se um utilizador introduz a querie "Jaguar" e visualizar os documentos referentes a "car" então é porque muito provavelmente está interessado em "Jaguar car". Ao descobrir este tipo de informação é possível utiliza-la de várias formas como no auxílio à criação de um perfil de utilizador, ou em reordenar os resultados se a pessoa avançar para os próximos resultados da querie.

A metodologia proposta para encontrar os referidos termos baseia-se em dividir o conjunto de resultados (T) em dois subconjuntos T_C e $T_{\bar{C}}$, que representam respectivamente os documentos visualizados e os não visualizados. O próximo passo, é encontrar o termo w (w $\in$ W que representa todos os termos existentes em todos os

documentos) que ocorre mais frequentemente nos resultados clicados (Pc (w)) e menos frequentemente nos resultados não visualizados ($P_{\bar{C}}$ (w)).

Calculando d(w) = $|P_C(w) - P_{\bar{C}}(w)| \times \text{Log} \times \frac{(2-P_{\bar{C}}(w))}{(2-Pc(w))}$, obtêm-se um valor entre [-1 e 1], sendo que quanto mais próximo de 1 estiver este valor, mais força tem o termo no conjunto de documentos que o utilizador consultou e mais representativo do interesse do utilizador.

Tanner [3], aponta a criação de uma hierarquia de interesses do utilizador por forma, a melhorar a personalização dos motores de pesquisa.

A sua abordagem para a reunião dos documentos que mostram os interesses do utilizador, passa simplesmente pela consulta do histórico das páginas favoritas deste. Após reunir o conteúdo de cada página, efectua uma limpeza do seu conteúdo através da remoção de palavras sem significado e posteriormente, efectua um processo de stemming (transformar as palavras na sua raiz). Cada documento passa a ser representado por frases cujos termos são todos significativos. Destas frases são retirados todos os possíveis pares de palavras com um algoritmo próprio denominado "Divisive Hierarchical Clustering" (DHC), cuja entrada é um grafo constituído por todas as palavras extraídas dos documentos e as relações entre elas (obtidas dos pares de palavras). A partir deste grafo é feita a criação da árvore de interesses.

A raiz da árvore é um cluster com todas as palavras. Os elementos de cada modo vão também eles ser Clusters (filhos), os quais contem subconjuntos de termos pertencentes ao seu pai. De modo a determinar os clusters filhos, inicia-se a parte divisiva do algoritmo que é a eliminação de algumas ligações entre os elementos com base no cálculo da pertença a um intervalo definido. O valor (para reordenar os resultados de uma pesquisa) para cada documento é calculado através do somatório do valor dado a cada termo existente na árvore e no documento.

Liu, Yu e Meng [28] efectuam a personalização das pesquisas na Web através do mapeamento das queries do utilizador em categorias. O processo desenvolve-se em três fases: primeiro, modelar e reunir a informação proveniente de uma secção de pesquisa;

segundo, construir o perfil de utilizador baseado no histórico das pesquisas e num perfil genérico construído a partir da hierarquia de categorias existentes no Open Directory Project (ODP); terceiro, deduzir as categorias que estarão associadas e uma nova querie introduzida pelo utilizador.

A primeira fase implica reconhecer o que é realmente relevante para o utilizador, a nível de documentos. São utilizadas as técnicas mais comuns como o tempo passado a ver o texto, o scroll e a movimentação do rato. Para cada querie introduzida são guardadas duas categorias relacionadas bem como os documentos lidos e que pertencem a estas categorias. A informação sobre estas categorias pode ser obtida em alguns motores de pesquisa que relacionam certos documentos a categorias.

A segunda fase, transforma esta informação numa árvore de interesses, que implica organizar as relações entre queries e documentos, bem como, categorias e documentos.

Para tal, é feito o uso de duas matrizes (DT e DC), onde as linhas são os documentos e as colunas as palavras da querie em DT, ou as categorias em DC. A cada posição da matriz DT, o valor atribuído tem por base o cálculo do TFIDF da palavra.

A partir de DT e DC computa-se a matriz M, cujos valores das linhas representam a relação entre os termos e as categorias. O mesmo tipo de processo é realizado para a construção da matriz Mg, representativa do perfil genérico obtida a partir das relações entre documentos e categorias existentes no ODP.

A forma como são criadas estas matrizes, M e Mg, passa pela utilização do algoritmo Linear Least Squares Fit (LLSF), sobre as matrizes DT e DC de modo a aproximar $DT*M^T$ a DC.

Para uma aprendizagem adaptativa, é utilizado como alternativa o método de Rocchio.

Por fim, de forma a calcular as categorias mais próximas de uma nova querie introduzida pelo utilizador, é calculada a distancia com base na função de Cosine entre a querie e cada categoria de M.

Capítulo 5

5 - Categorização de Web snippets e criação de modelos de utilizador

5.1 - Proposta de trabalho

Como já referido no capítulo 2, o processo de categorização permite agrupar em conjuntos elementos com propriedades semelhantes. Esta metodologia é aplicada em diversas disciplinas e a pesquisa na Web é uma das quais o futuro do clustering parece mais promissor. O estado da arte propõe a categorização automática como solução para resolver os problemas relativos à ambiguidade das queries e ao crescente número de resultados relevantes devolvidos por um sistema IR. O motor de pesquisa recebe a querie, encontra os resultados relevantes e numa fase final, processa estes de modo a encontrar similaridades entre eles de tal forma que possam ser colocados em vários grupos, devidamente identificados. Desta forma o utilizador recebe não só a lista de resultados como uma estrutura com a qual pode interagir e centralizar a sua atenção para um conjunto de documentos, já mais restrito, pertencente a uma categoria.

Com o mesmo objectivo de melhorar a performance dos sistemas WebIR, o estado da arte refere também o uso do clustering como um dos auxiliares na criação de modelos de utilizador. Estes modelos, são depois adaptados aos sistemas WebIR e permitem fazer com que possa existir uma personalização dos resultados devolvidos. Nem todas as metodologias que geram modelos de utilizador necessitam de recorrer ao clustering mas as que o fazem, normalmente aplicam os algoritmos de categorização a uma lista de documentos relativos ao histórico de navegação do utilizador. Dado ao fim de algum tempo o utilizador ter visitado uma imensa quantidade de páginas Web, há a necessidade de filtrar um pouco este conjunto onde para isso são aplicadas diversas abordagens que foram referidas no capítulo 3. Um modelo de utilizador tem como objectivo caracterizar este e a forma automática de efectuar isso está na extracção do conhecimento escondido atrás dos documentos consultados pela pessoa ao qual o modelo se destina. Ao submeter estes documentos a um algoritmo de categorização, as

particularidades destes permitem encontrar muita informação oculta e comum, muitas vezes representativa dos interesses do utilizador. As categorias geradas representam normalmente estas áreas de interesse, e a transformação da hierarquia em um modelo de utilizador quantificado é feita de forma simples.

O processo referido anteriormente, necessita do armazenamento dos documentos que são relevantes para o utilizador, fazendo com que passado algum tempo já seja necessária uma base de dados gigantesca para se poder efectuar a criação ou a actualização dos modelos de utilizador. Este tipo de metodologia talvez não seja a mais indicada para a personalização dos resultados para os milhões de utilizadores que efectuam pesquisas na Web. Deste modo e após ter estudado profundamente o assunto decidimos seguir um caminho diferente para a criação dos modelos de utilizador que é descrito de seguida.

Sabendo que os documentos guardados apenas servem para encontrar informação que seja comum entre os documentos, normalmente representativa dos interesses do utilizador, porque não efectuar o mesmo mas a partir de uma fonte diferente. Ao efectuar a categorização dos resultados para as pesquisas, é possível encontrar conhecimento que não era visível de forma explícita por entre os resultados. Estas categorias são úteis para o utilizador porque lhe podem ser chamativas e quando o são normalmente deve-se ao facto de irem de encontro aos seus interesses. Podemos então dizer que as categorias resultantes de uma pesquisa podem ser consideradas como áreas de interesse do utilizador. Resta agora encontrar quais as categorias que são realmente interessantes para cada pessoa. Num sistema trivial, poderíamos simplesmente gerar estas categorias, propor ao utilizador e as que este consulta-se seriam os seus interesses. Contudo, no mundo real as coisas não funcionam assim e muitas vezes o utilizador não faz uso destas categorias, pois ele próprio não sabe muito bem a categoria onde está aquilo que pretende. Cabe ao sistema encontrar estes interesses automaticamente.

Ao analisar o histórico de pesquisas do utilizador constituído apenas pelas queries que este introduziu e que lhe providenciaram documentos realmente de interesse, é possível construir uma árvore relacional de pesquisas. Esta árvore só tomará em conta pesquisas cujos termos ocorrem com alguma frequência em termos de introdução e de tempo, visto serem estes os mais representativos dos interesses reais do

utilizador. Uma pesquisa feita por impulso pode ser feita de longe a longe, mas as pesquisas que são feitas por modo a encontrar informação relativa a interesses deste são muito mais frequentes. Quem está interessado em linguagens de programação, provavelmente irá pesquisar muito sobre vários tipos de linguagem ao longo do tempo, mas se necessitar de saber onde fica um determinado local, provavelmente só fará pesquisas relativas a isso num único dia. O tipo de estrutura daqui resultante irá certamente gerar um vasto número de ramificações, pois muitas pesquisas podem ser feitas com recorrência a palavras diferentes apesar de recaírem sobre uma mesma categoria. É nesta fase que pensamos ser possível dar um outro uso á categorização dos resultados das pesquisas. Ao termos ramificações de termos na árvore cujos representam pesquisas do utilizador, guardamos também para cada uma dessas pesquisas as categorias que lhe estão relacionadas. Deste modo, podemos encontrar as categorias mais comuns entre as pesquisas que pertencem a cada ramo. Ao fazer isto para cada ramo, obteremos as categorias que melhor representam cada um (as mais frequentes e mais comuns a todos os elementos) e que muito provavelmente representarão os reais interesses do utilizador.

Também muito provavelmente, existirão repetições de categorias devido ao facto já referido de que pesquisas que têm termos diferentes muitas vezes recaem sobre a mesma categoria. Ao acontecer isto, estas categorias terão ainda mais importância no que toca a quantificar os interesses do utilizador.

Digamos por exemplo, (caso muito simples) um utilizador pesquisa bastante por shimano xt, shimano brakes, por mavic, mavic wheels e por truvativ, truvativ firex, truvativ brakes. A estrutura resultante irá ser do género apresentado na figura 5.1.

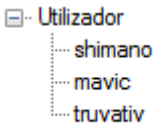

Figura 5.1: Perfil do utilizador sem categorização.

Ao ser incluída a informação sobre as categorias, a estrutura resultante será algo que já contem um dado mais específico que demonstra que todas estas pesquisas recaem sobre a categoria de bicicletas de montanha, ou seja, o utilizador está interessado neste tipo de modalidade. Ver figura 5.2.

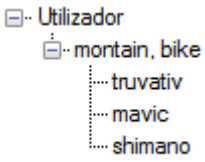

Figura 5.2: Perfil do utilizador com categorização.

A criação deste tipo de modelos de utilizador, leva a que o trabalho tenha de ser dividido em duas partes: a categorização dos resultados das pesquisas e a criação dos respectivos modelos de utilizador.

5.2 - Categorização dos resultados de um meta-motor de pesquisa

A literatura mostra que a categorização de Web snippets resultantes de uma pesquisa é um processo que acresce dificuldades às correntes metodologias de categorização. Tal deve-se ao facto do conjunto de informação ser muito mais reduzido e apresentar particularidades muito interessantes como o conteúdo dos snippets ser muito dependente da querie ou as frases existentes serem alvo de grande personalização como a repetição de palavras para dar ênfase ao conteúdo que muitas vezes nem é assim tão apelativo. É por isso, difícil classificar as palavras com recorrência às simples medidas de tfidf e sem recorrer a outros serviços que providenciem uma avaliação da categoria morfológica das palavras.

Em [10] Gulli e Ferragina, propõe um sistema capaz de efectuar a categorização dos resultados de uma pesquisa com o recurso a algumas medidas que a nosso ver são realmente interessantes. Dentro do universo académico, esta é também a abordagem que conduz a melhores resultados. Dizemos dentro do universo académico porque se olharmos para o mundo comercial, existe um sistema cujo está na vanguarda e os seus resultados são irrepreensíveis. Falamos do motor de pesquisa Vivissimo (também referido como Clusty), um sistema já existente há alguns anos mas que ao longo do

tempo tem vindo a evoluir muito positivamente. Infelizmente não existe literatura que faça uma análise mais profunda à metodologia utilizada por este sistema. Parte da metodologia proposta por Gulli e Ferragina serviu de inspiração para a criação do nosso algoritmo pois a sua proposta tem em conta para a criação das categorias, não o cálculo da similaridade entre documentos, mas sim a procura das melhores e mais frequentes expressões que existem entre os documentos de forma a criarem os grupos em torno destas mesmas expressões. O nosso algoritmo de certa forma também ambiciona efectuar o mesmo. Encontrar os termos mais fortes por entre o conteúdo dos documentos para depois agrupar os documentos que contemplam estas expressões. A diferença principal é que não introduzimos conhecimento no nosso algoritmo como o uso de bases de dados classificativas (o Snaket usa o DMOZ) ou remoção de stop-words, nem adicionamos informação ao snippet de cada url. Dada a natureza do nosso algoritmo, que se baseia na selecção de termos fortes para a construção dos clusters e atribuição dos nomes ou "labels" dos clusters também com base nestes termos, este foi denominado por CBL, Clustering by Labels.

5.2.1 - Obtenção dos documentos relevantes para uma querie a partir de um meta-motor de pesquisa

Não sendo o objectivo deste trabalho melhorar o os motores de pesquisa relativamente á recolha dos documentos que mais estão relacionados com uma querie, o processo de obtenção destes documentos passou pela criação de um meta-motor de pesquisa [9]. Um meta-motor de pesquisa é um sistema que faz uso de vários motores de pesquisa existentes para obter um conjunto de documentos mais forte no que toca á relação com a querie. Isto acontece porque ao recolher os resultados de cada motor, o conjunto resultante irá conter uma maior variedade de assuntos visto que cada motor de pesquisa cobre uma área de páginas Web que não é exactamente a mesma dos outros motores. Estes diferentes motores de pesquisa também aplicam diferentes algoritmos para a classificação dos documentos, logo os resultados também são muitas vezes diferentes. É também importante referir que a representação dos documentos deste meta-motor de pesquisa é feita através de snippets. Ou seja, cada documento é representado pelo seu respectivo url, o título e algumas frases mais importantes do documento.

Figura 5.3: Exemplo do snippet de um documento pertencente ao conjunto de resultados no motor de pesquisa Google para a querie "*Apple*".

O nosso meta-motor de pesquisa recorre ao uso de três dos mais conhecidos motores de pesquisa: o Google, o Yahoo e o MSN Live cuja ultima remodelação levou á alteração do seu nome para Bing. Para cada motor de pesquisa são obtidos os primeiros 50 a 100 resultados, conforme o valor escolhido pelo utilizador. Foi estabelecido este intervalo pois achamos que os resultados a partir da posição 100 já não são úteis para o que se pretende visto que é muito raro um utilizador navegar até ao resultado numero 100 para verificar se existe algum conteúdo de interesse. O mínimo imposto deve-se ao facto de ser necessário uma quantidade relativa de documentos para que possa existir uma boa geração de categorias.

Visto ser possível que os diferentes motores de pesquisa possam devolver alguns documentos iguais, é necessário calcular o valor de importância que estes vão ter relativamente aos outros documentos. Essa importância é calculada a partir do ranking dado ao documento por cada motor. O valor de ranking atribuído pelo nosso meta-motor de pesquisa para cada documento dá mais importância aos resultados do Google.

$$R(d) = 0.4 * Rg(d) + 0.3 * Rm(d) + 0.3 * Ry(d)$$

Onde $R(d)$ representa o ranking dado ao documento d, $Rg(d)$ é o ranking dado pelo Google ao documento, $Rm(d)$ é o ranking dado pelo MSN ao documento e $Ry(d)$ é o ranking dado pelo Yahoo ao documento.

Quando um ou mais motores não devolvem o mesmo documento, o seu peso é dividido para o peso de cada motor que possa ter devolvido o resultado de forma equitativa.

$$R(d) = 0.5 * Rm(d) + 0.5 * Ry(d)$$

A informação relativa ao título e às frases mais importantes de um documento que é devolvido por vários motores de pesquisa é unida numa estrutura que representa o conteúdo do documento. Ou seja os vários títulos são unidos numa única representação

de texto bem como numa outra variável são unidas todas as frases mais importantes desse documento. Mais para a frente iremos explicar porque não faz diferença nesta fase verificar se a informação relativa ao mesmo documento e proveniente de diferentes motores, não é igual.

5.2.2 - Aglomeração dos snippets por domínio e pré-processamento dos dados

Esta é uma das fases mais importantes para que o processo de categorização possa ter como entrada boas palavras. Para tal é necessário primeiramente agrupar todos os resultados de acordo com o domínio do url. Isto vai ser útil numa posterior fase para poder analisar as várias frases que possam existir em documentos que pertençam ao mesmo domínio.

O pré-processamento dos dados implica a remoção de tudo o que é lixo numa frase. Por lixo entende-se as tags utilizadas por certos motores de pesquisa para realçar algumas palavras, o uso de caracteres especiais para representar conteúdo html, simbologias que muitos utilizadores usam para personalizar as suas frases mas que não introduzem nenhum conhecimento e alguns caracteres especiais cuja remoção não altera a estrutura das frases.

Após esta primeira filtragem efectua-se uma análise á estrutura do texto onde, com recurso a uma etiquetagem definida por nós, são marcadas as secções onde uma frase acaba, onde existe uma vírgula ou onde existem cortes. Cada sequência de caracteres é analisada e conforme a sua constituição é-lhe também atribuído um marcador.

%p	Final de frase
%d	Delimitador na frase
<o>	Conteúdo da palavra analisável
<a>	Acrónimo
<n>	Nome
<t>	Conteúdo descartável
<d>	Data
<m>m	Interrupção da frase

Tabela 5.1: Descrição dos marcadores utilizados no pré-processamento dos snippets.

Estes marcadores servem para que numa fase seguinte, certas propriedades que decidimos avaliar nas palavras possam estar marcadas nos snippets. Algumas destas propriedades são descritas de seguida:

Classificação de uma palavra como nome: Uma palavra é classificada com sendo representativa de um nome quando esta começa com uma maiúscula, sendo todos os restantes caracteres minúsculos, quando não esta no inicio de uma frase, e quando numa janela de ***p*** palavras em torno da que está em avaliação, não existem mais que ***n*** % de palavras também elas podendo ser classificadas de nomes. Tal restrição deve-se ao facto do conteúdo dos snippets ser muitas vezes alvo de uma má escrita onde por exemplo uma frase é composta toda ela por palavras que começam em maiúsculas.

Classificação de uma palavra como acrónimo: O processo é mesmo que para a classificação de nomes e com as mesmas restrições. A única diferença é que a palavra tem de ser composta na sua totalidade por letras maiúsculas.

Classificação de uma palavra ou expressão como não útil: Todas as sequencias de caracteres que tenham mais de um hífen, que representem hiperligações ou cujo seu conteúdo esteja representado por caracteres não alfanuméricos com a excepção dos hífenes.

```
Snippet sem pré-processamento:

The <b>apple</b> is the pomaceous fruit of the <b>apple</b> tree, species
Malus domestica in <br>  the rose family Rosaceae. It is one of the most
widely cultivated tree fruits <br>  <b>...</b>


Snippet com pré-processamento:

<o>the <o>apple <o>is <o>the <o>pomaceous <o>fruit <o>of <o>the <o>apple
<o>tree %d <o>species <n>malus <o>domestica <o>in <o>the <o>rose <o>family
<n>rosaceae %p <o>it <o>is <o>one <o>of <o>the <o>most <o>widely
<o>cultivated <o>tree <o>fruits <m>m
```

Figura 5.4: Comparação entre um snippet com e sem pré-processamento.

5.2.3 – Extracção de características das palavras

Com o conteúdo dos snippets já devidamente etiquetado e agrupado por domínios passa a ser possível processar todos estes textos e reunir a informação relativa a cada palavra existente. Esta informação vai ser depois utilizada para calcular o nível de importância que o algoritmo vai dar a cada palavra. Alem de ser guardada a frequência com que cada palavra ocorre, o número de vezes que é considerada um nome, um acrónimo, o número de urls em que ocorre, são efectuadas contagens mais relevantes para o cálculo da sua importância. Estas têm a ver com a co-ocorrência da palavra com outras palavras na sua frente e retaguarda (caso existam claro). Após muito tempo a analisar as características dos snippets chegamos a uma teoria de que poderíamos obter a qualidade de uma palavra a partir do número de palavras diferentes que ocorrem com a palavra em questão. As palavras que não introduzem conhecimento relacionam-se com um número muito variado de palavras (por ex: "o carro", "o gato", "ganhou o", "o sol"). Pelo contrário palavras que induzam conhecimento na generalidade não se associam a um número tão grande de outras palavras. Ao sabermos o número total de palavras que ocorreram nas laterais da palavra em análise e o número de palavras diferentes deste conjunto, é possível quantificar o nível de interesse que essa palavra terá.

Dado que muitas vezes os motores de pesquisa utilizados no nosso meta motor devolvem resultados pertencentes ao mesmo domínio e cujo seu conteúdo é integralmente (ou mesmo exactamente) igual apenas variando o url, existiu a necessidade de adaptar o nosso algoritmo a esta realidade. A ignorância deste facto iria viciar os resultados no sentido de que ao serem repetidas muitas frases iria acontecer que as palavras iriam ocorrer com um conjunto muito pouco variado de palavras e com uma frequência elevada, isto resultaria numa boa classificação da palavra aos olhos do nosso algoritmo coisa que não seria verídica. O excerto que se segue de uma lista de resultados para a querie "***cars***" demonstra bem este caso:

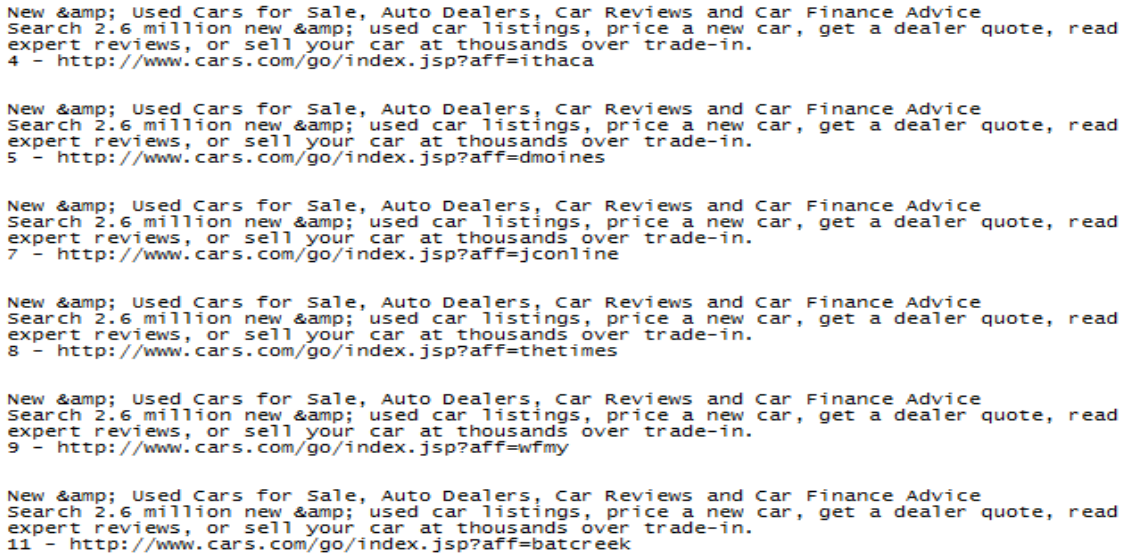

Figura 5.5: Conjunto de snippets para a querie “cars” cujo conteúdo é exactamente igual.

É este o motivo pelo qual os resultados foram agrupados por domínios. Porque o algoritmo efectua a extracção da informação através da análise das sequências de palavras que são distintas entre os urls de cada domínio. Cada uma destas sequências é também adicionada a uma lista que depois servirá para a extracção de expressões compostas. Na tabela seguinte (5.2) são apresentadas todas as características que são guardadas para cada palavra. Na figura 5.5 está o pseudo-código do algoritmo que permite a extracção desta informação por entre todos os resultados do meta-motor de pesquisa.

W	Representação em caracteres da palavra
Querie_Word	Indicação se a palavra existe na querie
F_Name	Número de ocorrências em que a palavra é considerada um nome
F_Acron	Número de ocorrências em que a palavra é considerada um acrónimo
S	Número de ocorrências em que a palavra está isolada
F	Frequência da palavra
U	Número de Urls em que a palavra ocorre
WIF	Número de palavras contadas na esquerda da palavra
WIR	Número de palavras contadas na direita da palavra
WDF	Número de palavras diferentes contadas na esquerda da palavra
WDR	Número de palavras diferentes contadas na direita da palavra
W_Class	Raiz da palavra (quando é utilizado stemming)

Tabela 5.2: Características analisadas numa palavra.

```
Criar lista de expressoes (E);

Por cada dominio
{
    Criar lista de expressoes_temp (Et);

    Por cada url de um domínio
    {
        Por cada expressao (exp) do url
        {
            se(singular) -> incrementar F e S;

            se(composta && !existe em Et) -> processar cada palavra (w) e analisar as associadas;
                                             incrementar F(w), WIL(w) ou WIR(w), WDL(w) ou WDR(w);
                                             se(w) = <n> -> incrementar F_Name(w);
                                             se(w) = <a> -> incrementar A_Name(w)
                                             adicionar exp a Et;
        }

    }

    Adicionar Et a E;

}
```

Figura 5.6: Pseudo-código para a extracção de informação das palavras

Visto que durante a execução do algoritmo é necessário estar constantemente a aceder ao objecto que contem a informação relativa a cada palavra, foi criada uma estrutura que permite encontrar muito rapidamente esse objecto. Esta estrutura consiste numa árvore de nível 3 onde cada ramo e sub-ramos contêm os caracteres do alfabeto. No final de cada ramo da árvore, existe um apontador para uma lista que contem as referências dos objectos que contem a informação das palavras cujo nome começa pelas letras desta ramificação. É assim possível saber muito mais rapidamente para um palavra onde está a estrutura que a representa.

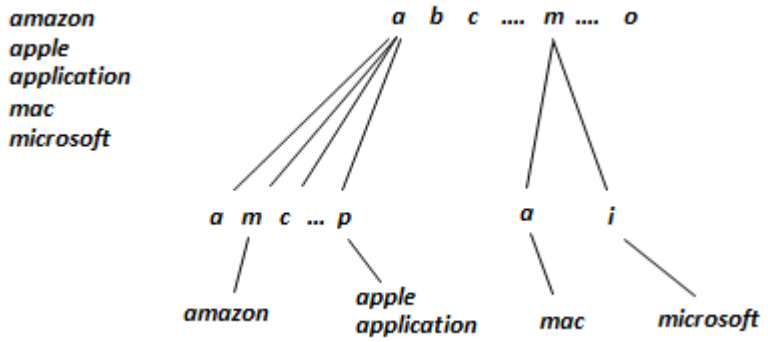

Figura 5.7: Estrutura de indexação das palavras em árvore.

5.2.4 – Cálculo do valor de importância de uma palavra e filtragem das palavras compostas

Após nas fases anteriores terem sido reunidos todos os dados relativos a cada palavra, o valor de importância final dado a cada uma destas é calculado com base no valor das seguintes propriedades:

Propriedade W1: Se um termo aparece sozinho num conjunto de texto, quer seja separado dos restantes termos por uma vírgula, um ponto ou outro separador, então é muito provável que esse termo tenha significado.

$$W_1(\mathrm{w}) = \frac{A(w)}{\ln\big(F(w)\big)}$$

Onde *w* é qualquer termo, *A(w)* é o numero de ocorrências em que *w* aparece sozinha e *F(w)* é a frequência do termo w.

Propriedade W2: Quanto maior for o numero de termos que co-ocorrer com qualquer termo w tanto no contexto do lado esquerdo ou do lado direito, então menos importante esse termo será.

$$W_2(\mathrm{w}) = \frac{WIL(w) + WIR(w)}{2 \times F(w)}$$

Onde *w* é o termo, *WIL(w)* e *WIR(w)* são o numero de termos que co-ocorrem imediatamente nos lados esquerdo e direito do termo w respectivamente e *F(w)* é a frequência do termo *w*.

Propriedade W3: Quanto maior for o número de termos diferentes que co-ocorrem com o termo w em ambos os seus lados esquerdo e direito comparativamente ao numero total de termos existente nos seus lados esquerdo e direito respectivamente, então, provavelmente menos importância terá essa palavra.

$$W_3(w) = \left[\left(\frac{\mathrm{WDL(w)}}{\mathrm{WIL(w)}} + \frac{\mathrm{WDL(w)}}{\mathrm{FH(w)}}\right) \times \frac{\mathrm{WIL(w)}}{\mathrm{F(w)}}\right] + \left[\left(\frac{\mathrm{WDR(w)}}{\mathrm{WIR(w)}} + \frac{\mathrm{WDR(w)}}{\mathrm{FH(w)}}\right) \times \frac{\mathrm{WIR(w)}}{\mathrm{F(w)}}\right]$$

Onde *w* é qualquer termo, *WDL(w)* e *WDR(w)* são respectivamente o número de termos diferentes imediatamente ao lado do termo w e *FH(w)* é igual ao *Max[F(w)],* para todos os termos *w*.

Propriedade W4: Se um termo aparece designado pelo processo de pré-filtragem como sendo um nome ou um acrónimo com uma certa frequência num conjunto de texto, então é muito provável que esse termo tenha significado.

$$W_4(\mathrm{w}) = \frac{I(w)}{\ln\big(F(w)\big)}$$

Onde w é qualquer termo, *I(w)* é o valor da melhor representação do termo como sendo acrónimo ou nome e *F(w)* é a frequência do termo *w*.

Baseado nestas quatro propriedades é possível atribuir um valor de importância a w.

$$W(w) = \begin{cases} W_2(w) \times W_3(w), W_1(w) < 0.5 \\ \dfrac{W_2(w) \times W_3(w)}{1 + W_1(w)}, W_1(w) \geq 0.5 \ \text{^} \ W_4 < 0.5 \\ \dfrac{W_2(w) \times W_3(w)}{1 + W_1(w) + W_{4(w)}}, W_1(w) \geq 0.5 \ \text{^} \ W_4 \geq 0.5 \end{cases}$$

Onde *W(w)* é o valor da importância do termo que quanto mais baixo for, mais importante é o termo.

5.2.5 – Geração das categorias com base nas palavras mais fortes

Após todas as importantes palavras terem sido identificadas, elas vão, como já havia sido referido, representar um papel crucial no processo de categorização dos resultados visto que este tem por base a escolha destas melhores palavras. O algoritmo é executado nos três seguintes passos: Criação dos pólos, unificação e absorção, escolha de um nome identificador para o conteúdo do cluster.

Criação dos pólos: É necessário inicializar o algoritmo para que sejam escolhidos os termos mais representativos. Com esse propósito, todas as palavras que pertençam a um intervalo α definido e que existam em mais de dois urls, são propostas para centros iniciais de clusters (pólos). A cada pólo, é associado uma lista de urls. Um url é adicionado à lista se contem a palavra que está no pólo entre as β primeiras posições de uma lista ordenada de palavras por grau de importância para esse url. Particularmente, o valor de β permite controlar o número de urls que é adicionado a cada pólo visto que um

β baixo vai produzir clusters mais pequenos e um β maior irá levar á existência de mais resultados dentro dos clusters.

União e Absorção: O próximo passo pretende iterativamente unir clusters cujos contenham urls similares. Para este propósito definimos dois tipos de aglomeração: união, quando dois clusters contem um numero significante de urls comuns e partilham de um tamanho semelhante em termos do número de urls. Absorção, quando partilham muitos urls mas o tamanho dos clusters é bastante diferente, sendo um bem mais numeroso que o outro. Como consequência definimos duas proporções: P1, o número de urls comuns aos clusters dividido pelo número de urls do cluster mais pequeno e P2, o número de urls do cluster mais pequeno dividido pelo número de urls do cluster maior. O algoritmo que se segue é iterado.

Por cada cluster, P1 é calculado relativamente a todos os outros clusters. Depois por cada par de clusters, se P1 é mais elevado que o valor de uma constante μ, então avalia-se o valor de P2 entre esses dois clusters. Se P2 é maior que uma constante ð, então o par de clusters é adicionado a uma lista de união, caso contrário é integrado numa lista de absorção. Assim que todos os clusters tiverem sido mapeados, ambas as listas de união e absorção são tratadas. A lista de uniões é tratada primeiro como se descreve de seguida.

Por cada par de clusters na lista de uniões, os dois clusters pertencentes as estes pares são juntos no cluster com o maior valor de importância da palavra *W(w)* que é representativa do seu centro. A cada passo deste processo, os índices dos clusters são substituídos e os clusters unidos são removidos da lista de uniões de forma a manter uma lista actualizada de clusters. Após todos os elementos desta lista terem sido processados, dá-se lugar ao tratamento da lista de absorções.

Iterativamente seleccionam-se o par de clusters que contem um qualquer cluster que não possa ser absorvido por nenhum outro cluster existente na lista. Quando encontrado, este cluster absorve o cluster que efectua par com ele. Os índices dos clusters são novamente actualizados e os clusters inutilizáveis são removidos. Ambas as listas são actualizadas e o processo repete-se, dando a possibilidade da geração de uma categorização do tipo plana (primeiro passo do algoritmo), ou a uma categorização hierárquica (todos os passos do algoritmo). De realçar que o algoritmo permite a um url

existir em diversos clusters, podendo assim o algoritmo ser classificado como Soft Clustering.

Escolha de um nome identificador para o conteúdo do cluster: através da união e absorção, cada cluster pode conter mais que uma potencial palavra para descrever o seu conteúdo. Contudo também pode acontecer que os urls do cluster contenham um outro tipo de palavras (expressões compostas) cujas providenciem um nome mais interessante para o cluster. Como consequência, estas expressões, extraídas após a fase de cálculo do grau de importância das palavras com o uso do algoritmo proposto em [30], são comparadas às frequências das palavras simples que pretendem representar o cluster. No caso da frequência de uma palavra composta ser mais elevada que um valor ℓ imposto de proporcionalidade com a frequência da melhor palavra simples candidata a nome do cluster, então será a expressão composta a identificar o conteúdo do cluster. Caso contrário será a palavra simples candidata mais forte.

5.3 - Criação de modelos de utilizador

5.3.1 – Reunião das queries que providenciam resultados de interesse para o utilizador

Para que possa ser criado um modelo de utilizador é necessário reunir informação que seja realmente relevante para este. Existem duas formas de reunir esta informação, implicitamente ou explicitamente. O problema de reunir dados que sejam de interesse para uma pessoa sob forma automática, ou seja implicitamente, reside na escolha dos indicadores que permitem saber que um documento é realmente relevante para este. Por outro lado o problema de reunir informação que o utilizador indica como sendo relevante para ele, é que este usualmente é pouco participativo.

A nossa escolha passou pelo uso das duas mas, com muito mais uso da informação recolhida sem percepção por parte do utilizador de modo a não o afastar do sistema. Como já referido na proposta de trabalho a metodologia escolhida para o nosso algoritmo de criação de modelos de utilizador passa, numa primeira fase, pela criação de uma árvore relacional das queries relevantes introduzidas pelo utilizador. Dizemos relevantes, porque é muito frequente fazermos uma pesquisa para a qual nenhum dos

resultados nos é chamativo, dando origem a uma das duas situações: a reformulação da querie, ou a desistência. Devido a isto existe a necessidade de saber que queries resultaram numa experiencia positiva para o utilizador. Muitos estudos [17] [18] [19] [20] [21] revelam que os factores implícitos indicadores do interesse de um utilizador num documento são o tempo que este passa a ver o documento e as acções de movimentação do cursor. A nossa opção para saber se uma querie foi ou não produtiva para um utilizador passou pelos seguintes indícios:

- Abertura de um ou mais documentos

- Em pelo menos um dos documentos abertos o utilizador passou mais de ***s*** segundos na sua leitura.

Se uma querie introduzida por um utilizador satisfaz estes requisitos, então é adicionada ao histórico da sessão de pesquisa do utilizador.

O termo sessão que ainda não havia sido referido anteriormente, pretende denominar todo o conjunto de pesquisas efectuado desde a abertura até ao encerramento da aplicação criada para servir de plataforma ao desenvolvimento deste trabalho. Esta aplicação tem as funcionalidades básicas de um browser visto que permite ao utilizador navegar por entre o conteúdo dos sites mas, também providencia o acesso ao meta-motor de pesquisa desenvolvido que permite obter os documentos mais relevantes para uma querie bem como a listagem das categorias que estão associadas a estes resultados. Em cada sessão são guardadas as queries relevantes do utilizador bem como as categorias associadas a estas queries e que foram obtidas a partir do algoritmo de categorização descrito na primeira secção deste capítulo. São guardadas também as categorias cujo utilizador escolheu explicitamente para consultar documentos que revelaram terem sido de interesse para ele. No final de uma sessão toda a informação é enviada e guardada num servidor criado para este efeito.

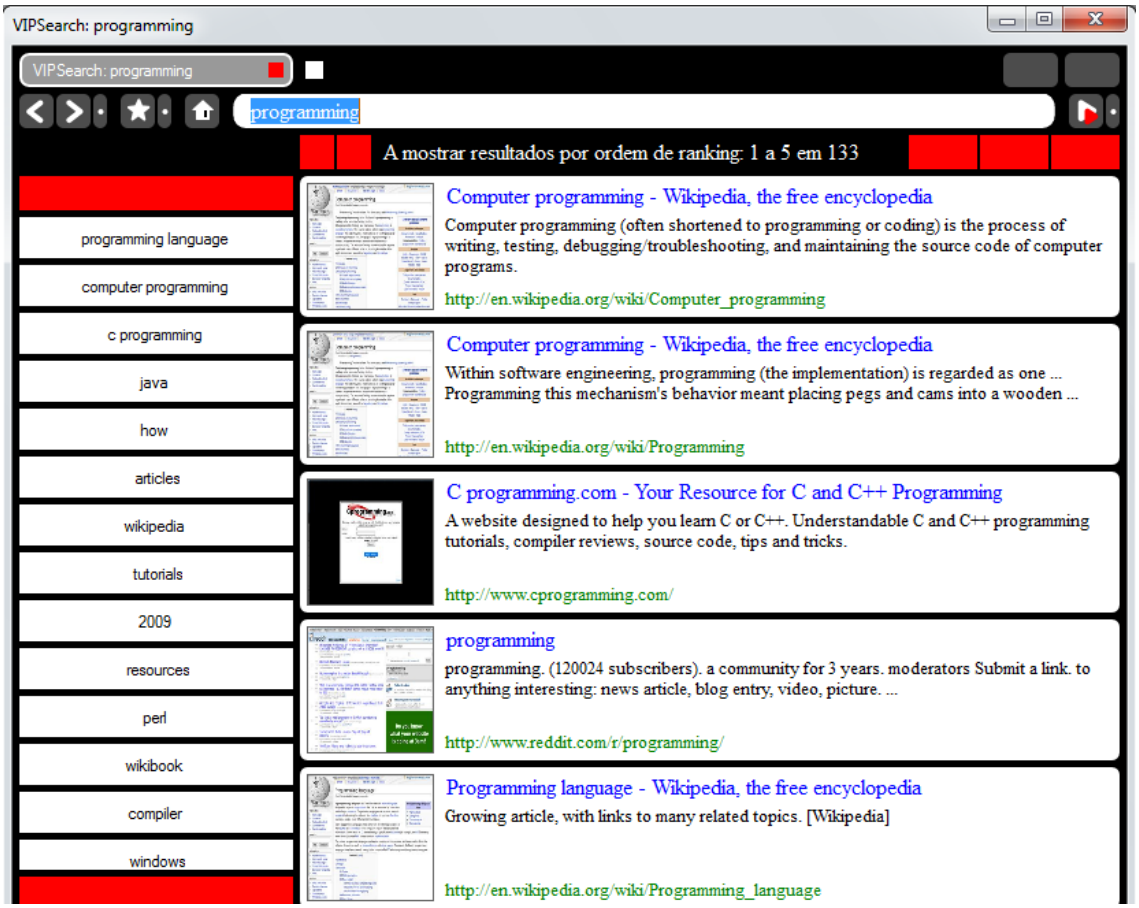

Figura 5.8: Aplicação VIPWeb.

5.3.2 - Criação da árvore relacional

Como já havia sido referido, a metodologia escolhida para a criação dos modelos de utilizador passa pela extracção do conhecimento existente nas queries. Na fase anterior foi reunida toda a informação possível sobre estas queries mas por modo a encontrar os interesses de longo prazo do utilizador, é necessário filtrar este conjunto. Ou seja só são consideradas para a criação da árvore as queries que satisfaçam as seguintes propriedades:

- A frequência de pelo menos um dos seus termos seja maior que ***fTm (frequência do termo mínima).fTm = 10***

- O intervalo de tempo decorrido entre o uso pela primeira vez de um termo e o uso pela ultima vez dele seja maior que ***iTm (intervalo de uso do termo mínimo).*** iTm = 15.

- O número de termos da querie não exceda o ***nTm (numero de termos máximo)***, foi necessária esta restrição pois uma querie com mais de um certo numero de termos é muito específica e não ajudaria em muito na criação de um perfil que pretende ser algo abrangente. Este número é o limitador do nível de profundidade da árvore. nTm = 4.

Por modo a tornar mais eficiente a selecção das queries que servirão para a construção da árvore de interesses, é criada uma tabela para cada utilizador onde para cada termo utilizado nas suas queries, são guardadas a frequência do uso desse termo bem como as datas de inicio e última utilização.

Com o uso desta tabela é possível seleccionar as queries que servirão de entrada ao algoritmo que as ramificará. Contudo, existem ainda dois problemas. A árvore que se pretende criar para servir de modelo de utilizador visa nos seus primeiros níveis de profundidade conter os termos mais genéricos e só depois apresentar os mais específicos. É por isso, necessário que os termos sejam classificados com um valor representativo do grau de generalidade. Este é um dos problemas. O outro, prende-se com a própria definição da palavra generalidade. Em certos casos, é fácil saber que um termo é mais genérico que outro. Por exemplo consideremos os termos: "automóvel" e "audi". É fácil saber que o termo "automóvel" é bastante mais genérico que "audi". Agora adicionemos a lista destes termos a palavra "análise". Sabemos que "análise" pode estar associado a todos os termos que se relacionam com "automóvel" e ainda a outras categorias sendo por isso mais genérico. Contudo na definição de um perfil de utilizador, será uma palavra que deva estar nos primeiros níveis a representar os seus interesses mais genéricos, ou será uma palavra representativa de uma particularidade do utilizador? Na implementação deste trabalho, ao calcular a generalidade das palavras, esta obterá um grau indicativo de uma forte generalidade. Contudo ao refinar a árvore gerada esta palavra passará para os níveis mais internos da árvore visto que o conhecimento que introduz sobre o utilizador só é relevante se estiver associada a outras palavras.

Com a atribuição de um grau de generalidade a todas as palavras, passa a ser possível efectuar-se a construção da árvore de interesses do utilizador. Esta é construída com base ordenação por grau de generalidade os termos que constituem cada querie.

Depois todas as queries são ordenadas por ordem alfabética numa lista, procedendo-se a uma execução iterativa dos seguintes passos:

Retirar o conjunto de queries que esteja no topo da lista e cujo primeiro termo de todas as queries seja o mesmo (D). Se o tamanho desse conjunto for maior que um valor ℓ imposto, então é criado um primeiro nível na arvore com esse termo. Do conjunto inicial D, obter o sub-conjunto SD que contenha termos de segunda ordem iguais. A esse conjunto reaplica-se novamente o conjunto de passos descrito até agora. E assim sucessivamente.

O algoritmo termina quando a lista de queries ordenadas estiver vazia, obtendo-se como resultado uma árvore da estrutura relacional dos termos das queries.

5.3.3 – Refinamento da árvore representativa dos interesses do utilizador com o uso da categorização

A árvore de interesses do utilizador obtida a partir da ramificação dos termos das queries já permite obter algum conhecimento sobre este. É contudo incompleta visto que somente a partir das queries não é possível extrair informação sobre todas as categorias de interesse para o utilizador. Um utilizador menos habituado ao uso de motores de pesquisa, se pretender encontrar informação sobre bicicletas provavelmente introduzirá uma querie com este termo. Ao ver o conjunto de resultados e as inúmeras categorias apresentadas irá com certeza numa próxima pesquisa efectuar uma querie mais forte como “bicicletas montanha”. Neste caso o sistema será capaz de pela ramificação das queries obter as categorias de interesse do utilizador. Contudo utilizadores mais avançados, cujas queries utilizadas são mais específicas, não permitem ao sistema saber que pesquisas por “ktm”, “trek”, “shimano”, “berg”, pretendem obter informações sobre o mesmo tipo de interesse do utilizador menos experiente (bicicletas).

O uso das categorias associadas a cada querie permite descobrir esta informação. Se um utilizador efectua muitas pesquisas sobre um determinado tema, é provável que esse tema seja uma das categorias encontradas pelo algoritmo de categorização dos resultados. Ou seja, um utilizador vai introduzindo as suas queries. A estas queries estão

associados um conjunto de tópicos. Os tópicos que muito provavelmente são relativos ao utilizador, são aqueles que são mais frequentes de aparecer no conjunto das queries do utilizador.

Para encontrar estas categorias, o algoritmo para cada sub-árvore no nível 1 da árvore de interesses de utilizador, extrai as categorias que são mais frequentes por entre o conjunto de queries pertencente a essa sub-árvore. Isto permite encontrar a categoria que melhor representa o conteúdo desta sub-árvore. Esta nova categoria é introduzida no nível 1 da árvore de interesses do utilizador, sendo a sub-árvore ligada a esta mesma categoria.

Visto que provavelmente muitas sub-árvores têm as mesmas categorias em comum, efectua-se a aglomeração dessas sub-árvores na respectiva categoria.

Por fim, e com o objectivo de retirar do primeiro nível as palavras cujo grau de generalidade é bastante elevado mas não denotam conhecimento, o conteúdo (analisado somente até ao nível 2) de cada categoria existente no nível 1 da árvore de interesses do utilizador é comparado com o restante conteúdo existente no nível 1. Se esse conteúdo (representado sobre a forma de termos) existir em algumas das categorias do nível 1, então a categoria de nível 1 em foco, é apagada e os termos da sua sub-árvore distribuídos pelas restantes categorias.

Capítulo 6

6 – Discussão dos resultados

6.1 – Categorização de Web snippets

O processo de categorização apresenta um grande problema no que diz respeito à análise de resultados. É muito difícil arranjar uma medida que possa quantificar o nível da qualidade dos resultados. Tal facto deve-se a este ser um processo automático e não supervisionado, não existem resultados pré-definidos como sendo bons resultados para que se possa fazer uma comparação. Cabe assim a avaliação dos resultados ser fruto de uma observação subjectiva entre o conteúdo dos resultados e que categorias um utilizador esperaria ter. Outra forma de avaliar os resultados é comparar o sistema criado aos sistemas que são referência neste sector. No nosso caso concreto, vamos comparar os resultados do processo de categorização implementado, aos resultados do motor de categorização referência no sector: o Vivíssimo. Para tal propomos a comparação entre os resultados obtidos para duas queries: "programming" e "apple".

É também apresentada uma listagem dos termos que o algoritmo de categorização classifica como sendo mais importantes para a querie "programming". É a partir destes termos que cada grupo é construído e muitas vezes estes termos, em conjunto com uma lista de palavras compostas, podem representar o label, ou descrição, desse mesmo grupo.

No capítulo 5, ao ser descrito o algoritmo implementado (CBL), foi referido que este é capaz de gerar uma categorização hierárquica. Os resultados apresentados não vão de encontro a esta definição. Tal facto deve-se ao termos seleccionado apenas os grupos do primeiro nível, ocultando os sub-níveis e por isso apresentando uma lista de categorias plana. O motivo para a escolha tomada deve-se ao uso deste algoritmo para auxiliar à construção do modelo de utilizador onde as subcategorias já seriam demasiado específicas para o tipo de perfil de utilizador que ambicionamos construir.

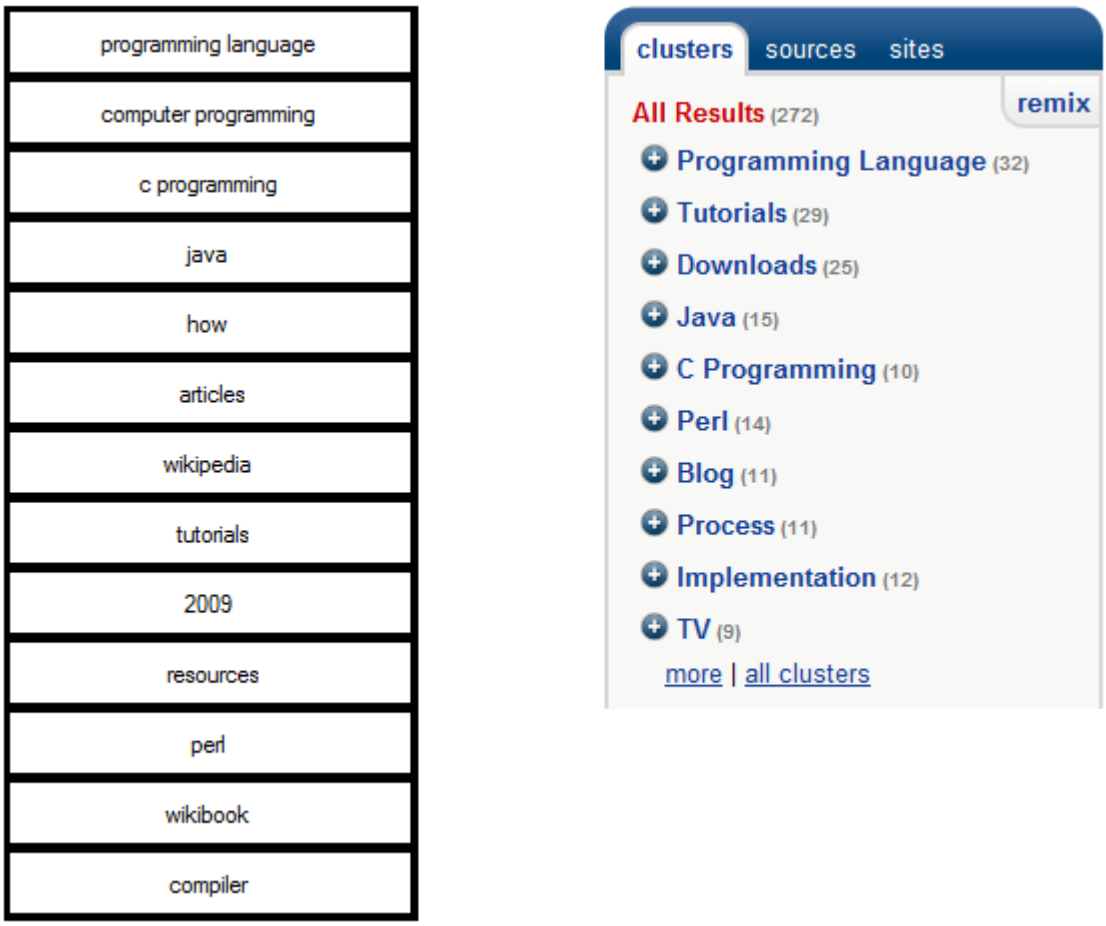

Figura 6.1: Comparação de resultados para a querie “programming” entre o algoritmo CBL (à esq) e Vivíssimo (à dir).

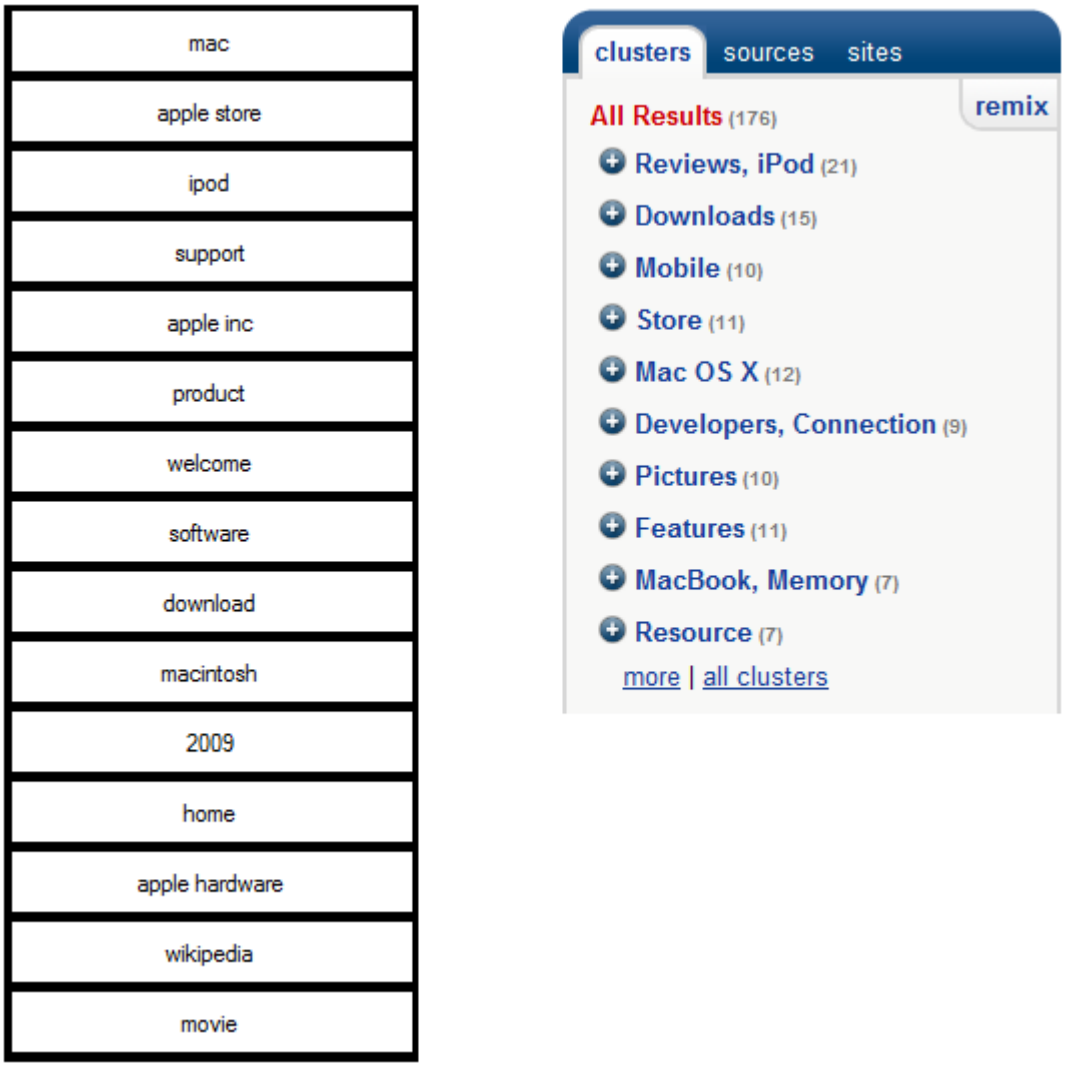

Figura 6.2: Comparação de resultados para a querie “apple” entre o algoritmo CBL (à esq) e Vivíssimo (à dir).

```
sql - H: 0
php - H: 0
testing - H: 0
wikipedia - H: 0,000868445022923877
programming - H: 0,0101186718525534
perl - H: 0,0126150786185803
wikibook - H: 0,0183589086907047
javascript - H: 0,0199249779479264
html - H: 0,0291500944615555
java - H: 0,0301692398748821
cgi - H: 0,0305867941961141
http - H: 0,0397369795740503
forum - H: 0,0397369795740503
subject - H: 0,0398499558958529
faq - H: 0,0426280557787185
window - H: 0,0503832084705061
compiler - H: 0,0656312225221776
edition - H: 0,0698403727896787
2009 - H: 0,0760064412238325
scheduling - H: 0,0794739591481007
os - H: 0,0853490028580011
search - H: 0,0958823955428256
channel - H: 0,0982639262828658
pres - H: 0,0982639262828658
c - H: 0,102853046061187
tool - H: 0,113674815409916
tv - H: 0,121113591448273
article - H: 0,138309034184131
work - H: 0,156810191368242
computer - H: 0,166521477033787
language - H: 0,168851964712066
resource - H: 0,179018740446899
how - H: 0,182298774197013
sample - H: 0,224154589371981
design - H: 0,225465838509317
3rd - H: 0,228019323671498
content - H: 0,243118602494382
technology - H: 0,248344575392463
tip - H: 0,254347826086957
online - H: 0,287653962476906
```

Figura 6.3: Lista ordenada por grau de interesse das palavras analisadas pelo CBL para a querie “programming”.

```
computer programming - 20
programming language - 34
language programming - 3
object-oriented programming language - 3
object-oriented programming - 6
c programming language - 5
c programming - 13
java programming language - 3
java programming - 9
programming perl - 4
programming art - 3
free programming - 3
free online - 5
online encyclopedia article - 3
learn how - 3
web design - 3
extreme programming - 3
```

Figura 6.4: Lista ordenada por grau de interesse das palavras compostas extraídas pelo CBL para a querie “programming”.

Como se pode observar pelos resultados, o algoritmo CBL é capaz de encontrar muitas das categorias que o Vivíssimo encontra, sendo isto um bom indicador de qualidade. Perde na não remoção de palavras que não introduzem conhecimento, ao propor a categoria "apple inc". Contudo, ao não remover este tipo de palavras, podem-se obter expressões mais fortes como "learn how", indicadora de que os documentos mostram como aprender a programar. Existe por isso aqui uma moeda de troca entre o guardar e não guardar as stop-words. Se por um lado muitas vezes prejudicam os resultados, por outro podem ajudar a realçar ou encontrar um assunto específico.

Na figura 6.3 é apresentada uma lista com os primeiros 40 termos qualificados pelo algoritmo com sendo os mais importantes. O algoritmo neste momento trabalha com palavras cujo nível de importância esteja abaixo de 1, um número talvez algo elevado e que muitas vezes possibilita que exista a passagem de termos não desejados. Mais uma vez é apenas uma questão de equilíbrio visto que, com um número próximo de 1, podem existir termos mais fracos, mas também são mantidos praticamente todos os bons termos. Se a constante for definida com um valor mais aproximo de 0, o conjunto de termos vai sendo reduzido, eliminando todas as palavras não desejadas, mas também removendo muitas que poderiam ser interessantes.

A lista de palavras compostas (figura 6.4) é completamente dependente da qualidade da lista de termos que são aceites. Isto porque se a lista de termos mais importantes contiver palavras sem interesse e muito comuns, é provável que estas venham a fazer parte das expressões compostas, reduzindo a sua qualidade.

Salienta-se ainda que a qualidade do processo de categorização vai-se deteriorando com a especificidade da querie. Isto vai de encontro á utilidade do processo de categorização que é realmente importante no caso de as queries serem pouco específicas e serem ambíguas. No caso de serem muito especificas, significa que o utilizador sabe muito bem aquilo que procura, reduzindo o numero de categorias passíveis de serem encontradas. Neste caso, podem surgir muitas palavras que não representam conhecimento. Tal facto deve-se á existência de muitas frases repetidas por entre os diversos resultados, fazendo com que as palavras co-ocorram com poucas palavras diferentes mas com uma frequência elevada, sendo classificadas pelo algoritmo como sendo relevantes.

6.2 – Perfis de Utilizador

Também um pouco como na avaliação dos resultados do processo de categorização, o perfil de utilizador criado é difícil de ser avaliado. Isto porque a única maneira de saber se os resultados são representativos ou não do utilizador passa pela sua avaliação subjectiva do modelo de utilizador criado para especificamente para ele. Dado a implementação dos modelos de utilizador estar ainda numa fase experimental no decorrer do trabalho, não foi possível reunir um número significativo de testes que pudessem comprovar a eficiência da metodologia proposta.

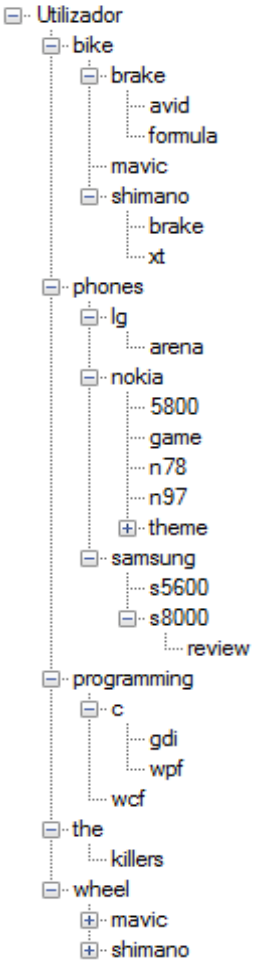

Figura 6.5: Perfil de um utilizador.

Os resultados apresentados neste perfil de interesses, vão de encontro á auto-avaliação feita pelo utilizador alvo da experiencia onde indicou que os seus interesses recaiam sobre as categorias: música, ciclismo, programação, tecnologia. Num futuro este perfil poderá ser utilizado para adaptar os resultados a uma pesquisa efectuada pelo utilizador, sendo escolhidos como resultados mais relevantes, aqueles cujos termos se encontrarem mais nas categorias que definem o utilizador.

Capítulo 7

7 - Conclusão e trabalho futuro

7.1 – Conclusão

Quando iniciamos o trabalho tínhamos como objectivo melhorar a experiencia de pesquisa para o utilizador. Para tal foi aprofundado o conhecimento relativo ao modo de operar dos motores de pesquisa bem como sobre todo o conjunto de metodologias existentes ou em fase de implementação, que visam melhorar os resultados destes sistemas. Deparamo-nos com dois caminhos algo distintos: a categorização da informação e a personalização. Se de um modo a categorização e a personalização estão associados, este é porque simplesmente na personalização a categorização é um dos meios utilizados para construir os modelos de utilizador como foi referido no capítulo 4. Se não for por este motivo, normalmente os sistemas ou fazem a categorização dos resultados e adicionam esta informação aos resultados ou então devolvem uma lista de resultados personalizada mas sem mostrarem nenhum tipo de categorias associadas a essa informação. Ao descobrirmos isto, iniciamos uma investigação com o intuito de juntar o que há de melhor nos dois mundos. Efectuar uma categorização dos resultados e construir modelos de utilizador que depois pudessem ser utilizados para a personalização dos resultados da pesquisa e da estrutura resultante da categorização.

Dada a extensão da ideia, optamos por impor como objectivo a categorização dos resultados e a criação dos modelos de utilizador que, como referido na introdução e na proposta de trabalho, fazem uso da categorização mas de uma maneira diferente de todos os artigos estudados relativos ao assunto. No nosso caso particular a categorização dos resultados serve para adicionarmos ao histórico das queries do utilizador a informação relativa aos assuntos que essas mesmas queries abordam. Esta foi uma ideia inovadora e que permite a construção de um modelo de utilizador representado sobre a forma de uma árvore, simplesmente a partir de uma estruturação das palavras da querie e da interligação das categorias que são comuns a certos ramos da árvore. Sendo assim o trabalho de investigação dividiu-se em duas fases que no final se interligaram. A primeira, implementar um algoritmo que permitisse uma categorização dos resultados.

A segunda, fazer o registo das queries introduzidas e relaciona-las com as categorias geradas na primeira fase de modo a poder construir a partir dai um modelo de utilizador. Podemos deste modo retirar conclusões para as duas etapas independentemente.

Relativamente à fase de categorização, ao optarmos por também ai inovar no sentido de que pretendíamos estudar até que ponto era possível afastarmo-nos de introdução de conhecimento nos resultados provenientes dos vários motores de pesquisa utilizados, podemos concluir que realmente é possível encontrar grupos cujo conteúdo tem por base documentos que estão associados a palavras fortes encontradas através da nossa metodologia. Ao compararmos os resultados com a referência incontestável do sector, o Vivissimo, é possível ver que para a mesma querie, muitas das categorias deste poderoso motor se encontram entre as categorias devolvidas pelo nosso sistema. Uma das vantagens do nosso algoritmo ao não introduzir conhecimento como o enriquecer dos snippets com recurso a ferramentas externas (caso do Snaket), a não remoção de palavras sem conhecimento e a não utilização de algoritmos de analise morfológica é que faz com que seja possível obter resultados para qualquer tipo de linguagem dos continentes europeu e americano (certas particularidades do nosso algoritmo não se adaptam a linguagem chinesa por exemplo). Ao ser independente de informação externa, o desempenho do algoritmo também é bastante bom, gerando categorias em menos de um segundo após obter os resultados de todos os motores de pesquisa utilizados. Grande parte dos sistemas cujos existe documentação sobre as metodologias utilizadas, consegue apenas categorizar um numero limitado de idiomas, sendo que o Vivissimo consegue trabalhar com muitos mais mas não existe literatura que descreva as metodologias utilizadas. A não remoção de palavras sem conhecimento á a principal causa da existência de algum lixo que por vezes faz com que existam nomes de categorias pouco apelativos quando são introduzidas queries mais extensas. Contudo na maior parte dos casos o algoritmo consegue filtrar com sucesso a maior parte das palavras que não inferem conhecimento ou quando as mantém é porque elas estão muito ligadas a certas expressões como o caso de “c programming”.

Sobre a fase de construção de modelos de utilizador, podemos referir que este é o tipo de área para o qual existe menos trabalho relacionado, devido talvez á frescura do assunto no que toca a personalização dos sistemas de pesquisa. A abordagem

implementada foi mais uma vez fruto da tentativa de criar algo de novo. Sabendo que só a informação relativa ao historial de pesquisas (queries) do utilizador não seria suficiente para a criação de um modelo de utilizador forte, decidimos jogar com as categorias geradas para cada querie por modo a encontrar conhecimento oculto no historial de pesquisas. Desta forma foi possível encontrar informação que nunca havia sido introduzida nas queries. De um modo geral os modelos de utilizador criados automaticamente pelo nosso algoritmo representam os reais interesses destes. Contudo a estrutura dos modelos criados não é talvez ainda a mais correcta pois quando o histórico de pesquisas é bastante extenso e muito variado, faz com que exista uma modelo de utilizador com bastantes áreas de interesse. Outro dos problemas é de ordem morfológica e relacionado com o saber que termos são mais genéricos. Excluindo estes problemas, o algoritmo proposto consegue identificar sem recorrer ao conteúdo dos textos lidos pelo utilizador os interesses explícitos e implícitos deste, fazendo com que a teoria fosse comprovada na prática.

Com este trabalho, demonstrou-se que ainda existem muitas áreas a explorar no que toca á melhoria dos sistemas de informação. As metodologias propostas são a prova disso pois ficou demonstrado que é possível criar um sistema de categorização muito pouco dependente de conhecimento com resultados satisfatórios e gerar modelos de utilizador a partir do bom uso da categorização dos resultados e do histórico de pesquisa dos utilizadores. Tal como em todos os trabalhos de investigação, nunca existe o alcance da perfeição. É sempre possível melhorar aquilo que já foi feito e o nosso trabalho não é excepção. No tópico seguinte são descritas as possíveis melhorias ao trabalho bem como as ideias para a continuidade deste.

7.2 - Trabalho futuro

A motivação para este trabalho foi a de melhorar a experiencia de utilização dos sistemas de pesquisa. Um dos objectivos propostos, a categorização, já é um passo nesse sentido pois com a sua aplicação permite ao utilizador centrar-se numa categoria de resultados que lhe seja mais chamativo de entre as diversas categorias apresentadas. O modelo de utilizador, só passa a ser útil quando aplicado ao motor de pesquisa. Desta

forma o sistema passa a possuir conhecimento sobre o utilizador e pode reordenar os resultados e categorias de acordo com aquilo que é mais do seu interesse.

Deste modo, o primeiro passo a dar na continuação do trabalho é adaptar este modelo de utilizador aos resultados iniciais da pesquisa. Isto pode ser feito através de heurísticas simples como em [28] onde é feita apenas uma soma dos termos que aparecem no snippet e num dos ramos da árvore, sendo dado maior relevância aos documentos que mais termos tem em comum com os que descrevem o perfil do utilizador.

O processo de categorização pode ser alvo de muitas modificações pois ao longo do desenvolvimento do trabalho muitas novas ideias vão surgindo e nem todas são passíveis de ser utilizadas e experimentadas. A primeira passa por efectuar um cálculo do tipo PageRank para cada palavra do conjunto de resultados. Poder-se-ia até denominar WordRank, o objectivo seria para cada palavra guardar a lista de palavras com que ela ocorre numa janela para ambos os lados da juntamente com as suas frequências. Para cada uma das palavras calcular o seu peso tendo por base o nível de ocorrências que tem com outras palavras, mas também tendo em conta para cada uma dessas palavras a sua correlação com as outras palavras.

A extracção de palavras compostas efectuada, de momento é feita sem poderem existir saltos entre as palavras. Como foi visto na literatura, alguns algoritmos fazem a extracção de palavras compostas com a possibilidade de salto. Se introduzirmos esta funcionalidade e dermos mais relevância às palavras compostas no nosso algoritmo de categorização, poderemos melhorar os resultados até ao nível dos nomes das categorias.

O próprio processo de agrupamento por palavras mais fortes pode ser revisto e alterado onde a fase de união e absorção de clusters similares pode ter em conta outros parâmetros.

Relativamente á geração dos modelos de utilizador, para uma melhor filtragem dos interesses deste, podem ser utilizados os termos dos snippets dos documentos relevantes, cuja informação pode permitir tomar melhores decisões quanto á geração dos ramos da estrutura. Para ultrapassar os problemas relativos á posição que certas palavras devem ter na arvore, um inquérito pode ser efectuado e consoante isso ser tomada uma decisão mais consensual.

Bibliografia

[1] Hyoung R. Kim and Philip K. Chan: Personalized Ranking of Search Results with Learned User Interest Hierarchies from Bookmarks. In *WEBKDD Workshop, SIGKDD Conf.*, 2005.

[2] Wadee S. Al halabi, Miroslav Kubat, Moiez Tapia: Time spent on a web page is sufficient to infer a user's interest. In *IASTED European Conference on Proceedings of the IASTED European Conference: internet and multimedia systems and applications*, Chamonix, France, pp. 41–46 (2007).

[3] Tanner, Chris: Adaptive Web Personalization**:** Improving Web Personalization via User Interest Hierarchy and Scoring Techniques; Dec 2006.

[4] Nielsen Announces May U.S. Search Share Rankings, With Total Searches Increasing 20 Percent-Over-Year – www.nielsen-online.com

[5] Zeng, H., He, Q., Chen, Z., Ma, W., Ma, J.: Learning to Cluster Web Search Results. Annual ACM Conference on Research and Development in Information Retrieval Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 210 – 217, 2004.

[6] Zamir, O., Etzioni, O.: Web Document Clustering: a Feasibility Demonstration. Annual ACM Conference on Research and Development in Information Retrieval Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 46 – 54, 1998.

[7] Radovanovic, M., Ivanoic, M.: CatS: A Classification-Powered Meta-Search Engine. Advances in Web Intelligence and Data Mining, pp. 191-200, 11, August 2006.

[8] Singhal, Amit.: "Modern Information Retrieval: A Brief Overview". *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 24 (4), pp. 35–43, (2001).

[9] Dreilinger, D. and Howe, A.E.: Experiences with selecting search engines using metasearch. *Journal of ACM Transaction on Information Systems*, 15(3), pp.195–222, 1997.

[10] Ferragina, P., and Gulli. A.: A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. In *Proceedings of WWW05, 14th International World Wide Web Conference*, pp. 801-810, China, Japan, 2005.

[11] Atsumi, M.: Extraction of User's Interests from Web Pages based on Genetic Algorithm. In *IPSJ SIG Notes* (Information Processing Society of Japan, The Special Interest Groups Notes), 1997.

[12] Brin, S., and Page, L.: The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of 7th International World Wide Web Conference*, pp. 107–117, Brisbane, Australia, 1998.

[13] Chakrabarti, S.: Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann, 2003.

[14] Boldi, P., and Vigna, S.: WebGraph framework i: Compression techniques. In *Proceedings of the 13th International World Wide Web Conference*, pp. 595–602, New York, U.S., 2004.

[15] Salton, G., and McGill, M.: Introduction to Modern Information Retrieval. McGraw Hill, 1983.

[16] Frakes, W.B., and Baeza-Yates, R.: Clustering Algorithms, chapter 16. Prentice Hall, Englewood Clifs, NJ, 1992.

[17] Morita, M., and Shinoda, Y.: "Information filtering based on user behaviour analysis and best match text retrieval". In *Proc. 17th ACM Annual International*

Conference on Research and Development in Information Retrieval, Dublin, Ireland, pp. 272–281, July 1994.

[18] Kim, J., Oard, D.W., and Romanik, K.: "Using implicit feedback for user modeling in internet and intranet searching," Tech. Rep., College of Library and Information Services, University of Maryland, College Park, 2000.53

[19] Claypool, M., Le, P., Waseda, M., and Brown, D.: "Implicit interest indicators". In *Proc. International Conference on Intelligent User Interfaces, Santa Fe*, pp. 33–40, 2001,

[20] Goecks, J., and Shavlik, J.: "Learning users interests by unobtrusively observing their normal behavior". In *Proc. 5th international conference on Intelligent User Interfaces*, New Orleans, pp. 129–132, 2000.

[21] Chan, P.: "A non-invasive learning approach to building web user profiles". In *Proc. ACM SIGKDD International Conference,* San Diego, pp. 7–12, 1999.

[22] Aktas, M., Nacar, M., Menczer, F.: Using Hyperlink Features to Personalize Web Search. *Advances in Web Mining and Web Usage Analysis*, pp. 104-115, October 17, 2006.

[23] Tamine, L., Boughanem, M., Zemirli, N.: Inferring the user interests using the search history. *LWA, Vol. 1/2006*, pp. 108-110, (2006).

[24] Atsumi, M.: Extraction of User's Interests from Web Pages based on Genetic Algorithm. *IPSJ SIG Notes* (Information Processing Society of Japan, The Special Interest Groups Notes), 1997.

[25] Sieg, A., Mobasher, B., Burke, R.: Ontological User Profiles for Personalized Web Search. *Conference on Information and Knowledge Management Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 525-534, 2008.

[26] Singh, S., Murthy, H., Gonsalves, T.: Determining User's Interest in Real Time. *International World Wide Web Conference Proceeding of the 17th international conference on World Wide Web,* pp. 1115-1116, 2008.

[27] Schockaert, S., De Cock, M., Cornelis, C., Kerre, E.E.: Clustering Web Search Results Using Fuzzy Ants. *Lecture notes in computer science ISSN 0302-9743*, vol. 3172, pp. 342-349, 2004.

[28] Liu, F., Yu, C., Meng, W.: Personalized web search by mapping user queries to categories. *In Proceedings of the eleventh international conference on Information and knowledge management* (CIKM), pages 558-565. ACM Press, 2002.

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

[30] Frantzi K.T., Ananiadou S.: Retrieving Collocations by Co-occurrences and Word Order Constraint. *16th International Conference on Computational Linguistics*, (1996).

[31] Lipai, A.: World Wide Web Metasearch Clustering Algorithm. *Informatica Economica Journal*, pp. 5-11, 2008.

[32] Lipai, A.: Web Metasearch Result Clustering System. *Informatica Economica Journal*, pp. 113-116, 2008.

[33] Geraci, F., Pellegrini, M., Maggini M., Sebastiani, F.: Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. *String Processing and Information Retrieval, Web Clustering and Text Categorization*, pp. 25-36, 2006.

[34] Machado, D., Barbosa, T., Pais, S., Martins, B & Dias, G.: Universal Mobile Information Retrieval. *13th International Conference on Human Computer Interaction (HCII 2009)*. San Diego, USA. July 19 - 24.

[35] Osinski S., Stefanowsky J., Weiss D.: Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. *Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM´04 Conference*, Zakopane, Poland, 2004, pp. 359—368

[1u] www.google.com

[2u] www.yahoo.com

[3u] www.altavista.com

[4u] Artigo T3, Google: O passado, o presente, o futuro. Agosto 2009.

[5u] http://www.hollyscoop.com/michael-jackson/michael-jacksons-death-crashes-google_20696.aspx

[6u] www.clusty.com

[7u] www.amazon.pt

[8u] http://www.washingtonpost.com/wp-dyn/content/article/2007/07/23/AR2007072301543.html

[9u] http://www.internetnews.com/stats/article.php/1363881

[10u] www.bing.com

[11u] www.ask.com

[12u] http://search.carrot2.org/stable/search

[13u] http://www.mooter.com/

[14u] http://www.iboogie.tv/

[15u] http://www.kartoo.com/en/kartoo.html