UNIVERSIDADE DA BEIRA INTERIOR
Covilhã | Portugal

Construção Automática de Micro-Ontologias para a Personalização na Web

Tiago José Santos Barbosa

Tese submetida à Universidade da Beira Interior para o preenchimento dos requisitos para a concessão do grau de Mestre

Efectuada sobre a supervisão do Doutor Gaël Harry Adélio André Dias

Departamento de Informática
Universidade da Beira Interior
Covilhã, Portugal
http://www.di.ubi.pt

Agradecimentos

Em primeiro lugar, gostava de agradecer ao meu orientador, Professor Gaël Dias, por toda a ajuda que me deu na realização desta tese de mestrado. Queria agradecer igualmente a todos os outros membros do HULTIG por tudo o que me ajudaram e por tudo o que significam para mim. Destes queria destacar o David que foi quem esteve mais presente durante todo este processo e sem ele seria impossível eu entregar esta tese a tempo.

Agradeço também a todas as pessoas que durante este ano trabalharam comigo na Microsoft e que me ajudaram a crescer como profissional mas acima de tudo como pessoa. Foi um ano bastante complicado e sem vocês nada disto seria possível. Por último, e mais importante, gostava de agradecer a todos os meus amigos e família por estarem sempre presentes na minha vida independentemente da situação.

A todos, o meu muito obrigado!

Resumo

O facto da Web estar a crescer a um ritmo imparável e com uma grande falta de estruturação faz com que os sistemas de recolha de informação atravessem graves dificuldades ao tentarem atingir os objectivos para os quais foram criados. Por outro lado, estes problemas podem ter aspectos positivos visto que nos últimos anos existiu um acréscimo de investigadores a tentar arranjar soluções para os mesmos.

Existem várias abordagens para tentar resolver os problemas dos sistemas de recolha de informação sendo que um deles é a personalização e é nesta abordagem que nos vamos focar ao longo desta tese.

Os motores de pesquisa existentes nos nossos dias devolvem ao utilizador resultados gerais, ou seja, orientados para a globalidade dos utilizadores. O objectivo desta tese é melhorar a experiência do utilizador no motor de pesquisa, criando um perfil de utilizador e devolvendo-lhe os resultados mais aproximados aos seus gostos. Para isso será recolhida informação do utilizador, como por exemplo, as páginas visitadas bem como as categorias onde estas se integram, a quantidade de tempo que o utilizador passa numa página, a complexidade do texto lido, entre outras.

Assim sendo, vamos criar dois perfis de utilizador diferentes, o perfil do utilizador e o perfil de nível de conhecimento do utilizador. Tanto um como o outro serão criados *"offline"* pois necessitam de alguma quantidade de informação por isso têm que ser criados ao longo do tempo. O primeiro representa os gostos do utilizador tendo em conta o histórico de utilização e o segundo será uma representação do nível de conhecimento do mesmo, tendo como base, como é óbvio, as páginas que este visitou.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ARPA Advanced Research Projects Agency

ARPANET Advanced Research Projects Agency Network

IR Information Retrieval

Web World Wide Web

WebIR World Wide Web Information Retrieval

TF-IDF Term Frequency - Inverse Document Frequency

SVM Support Vector Machines

HULTIG Center for Human Language Technology and BioInformatics

Capítulo 1

Introdução

Em 1967, quando Dwight D. Eisenhower deu ordens para se iniciar o projecto ARPA e consequentemente a rede ARPANET com certeza que não imaginou o que estava a criar. No dia 1 de Dezembro de 1969 "nascia"finalmente a ARPANET estabelecida entre 4 Universidades nos Estados Unidos da América. Inicialmente, esta rede mapeava apenas um directório mas evoluiu de tal forma que a ARPANET é hoje em dia a Internet como a conhecemos. Um sistema mundial público que interliga todos os computadores.

Com a evolução da Internet para um sistema de informação público houve a necessidade de mapear o seu conteúdo para tornar a sua utilização mais simples. Foi nesta altura que começaram a aparecer os motores de pesquisa. O primeiro motor de pesquisa Web foi o Wandex, actualmente extinto, feito pela World Wide Web Wanderer. Um web crawler (programa automatizado que acede e percorre os sites seguindo os links presentes nas páginas) desenvolvido por Matthew Gray no MIT, em 1993. O primeiro sistema de pesquisa "full text"foi o WebCrawler, que saiu em 1994. Ao contrário de seus antecessores, ele permite aos utilizadores pesquisar por qualquer palavra em qualquer página, o que tornou-se padrão para todos os motores de pesquisa desde então. Ainda em 1994, o Lycos (que começou na Carnegie Mellon University) foi lançado e tornou-se um grande sucesso comercial.

São vários os métodos utilizados pelos diferentes motores de pesquisa, tais como os sistemas booleanos, probabílisticos, entre outros. Mas o mais conhecido apareceu em 1998 com o surgimento da Google. Nesta altura a Google apresenta um novo sistema que considerava a estrutura das hiperligações dentro dos documentos e não apenas o seu conteúdo. Para isso utilizavam um novo algoritmo denominado de

PageRank [1] que veio introduzir o conceito de citação na Web. Este conceito diz que quanto mais citações um documento tenha, maior importância lhe é dado.

Nos dias de hoje, a utilização dos motores de pesquisa é imprescindível. Estudos de mercado nos Estados Unidos da América revelam terem sido efectuadas 9.4 biliões de pesquisas nos principais motores de pesquisa apenas no mês de Maio de 2009 [2] e que tem vindo a existir um aumento considerável na utilização por parte dos utilizadores dos mesmos a cada ano que passa.

1.1 Problemática dos motores de pesquisa na Internet

A dimensão cada vez mais colossal e a falta de estruturação da informação na Internet faz com que os sistemas de recolha de informação enfrentem graves dificuldades no cumprimento das tarefas para as quais foram desenhados. Este facto originou um aumento na comunidade de investigadores que se debatem diariamente na resolução deste problema.

O principal objectivo de um motor de pesquisa é devolver os documentos considerados relevantes para uma sequência de palavras (querie) introduzidas por um utilizador. Uma querie na sua essência é um conjunto de palavras pertencentes a uma língua e com uma forte relação com a informação pretendida. Os resultados devolvidos são globais e podem mudar de um motor de pesquisa para outro devido aos algoritmos utilizados no processo de selecção assim como a quantidade de páginas indexadas.

Se pararmos um pouco para pensar na quantidade de páginas e o número de línguas diferentes na Internet chegamos rapidamente à conclusão que os problemas dos sistemas de WebIR não são poucos nem simples de resolver. Algumas das soluções normalmente utilizadas por estes sistemas são algoritmos de crawling eficientes, capacidade de processamento distribuído, capacidade de indexação distribuída, algoritmos de filtragem linguísticos, algoritmos de reconhecimento de spam, entre outras. Estas soluções servem para resolver problemas funcionais mas os 5 problemas principais dos sistemas de WebIR foram identificados por Ferragina e Gulli [3] e vão ser demonstrados nos próximos parágrafos.

O primeiro problema está na própria definição da palavra "relevante". Há já muito tempo que se trabalha em algoritmos capazes de devolver os resultados mais relevantes. Mas o que é a relevância de um documento? A relevância de um documento no contexto

dos motores de pesquisa é a importância que esse documento tem dentro do âmbito da querie introduzida pelo utilizador. O método com melhores resultados é o *PageRank* utilizado pela Google. Este método verifica as referências ao documento em foco e adiciona ou retira importância a este, de modo a torná-lo mais ou menos relevante. Contudo, nem sempre é a melhor solução pois não consegue ultrapassar certos problemas inerentes à complexidade da estrutura da Web. Para queries específicas, onde o número de documentos relacionados é significativamente reduzido, torna-se bastante difícil encontrá-los. Da mesma forma que para queries mais gerais, acabam por ser devolvidos milhões de documentos considerados relevantes mas que nem sempre o são. Desta forma e como este tipo de problemas é do conhecimento público existem algumas empresas que se publicitam na Internet utilizando um mecanismo denominado de optimização para motores de pesquisa.

O segundo desafio, e um dos maiores, é o já referido facto da enorme quantidade de informação disponível na Internet. É impossível nos dias de hoje um motor de pesquisa indexar todas as páginas existentes, já para não falar que para o funcionamento de um motor de pesquisa ser o ideal os seus índices têm que estar sempre actualizados e não pode haver hiperligações quebradas. Uma solução para o problema da indexação da maior parte dos documentos da Web passa por recorrer à utilização de um meta-motores de busca [4]. Estes exploram um conjunto de resultados provenientes de múltiplos motores de pesquisa e como tal aumentam o número de possíveis páginas. Contudo leva a outro problema que é a reordenação por grau de interesse.

O terceiro problema prende-se com a particularidade da relevância de um documento ser subjectiva dependendo do contexto em que está inserido. A mesma querie pode ter objectivos de pesquisa diferentes e por consequência atribuir relevâncias diferentes aos documentos que a contêm. Por exemplo, imaginando o caso em que dois utilizadores introduzem a mesma querie. Um utilizador pode estar a começar a aprender coisas sobre este tema e pretende documentos que o ajudem na iniciação, enquanto que o segundo já é um conhecedor do tema e quer ver documentos mais avançados. O que acontece num motor de pesquisa normal é que os resultados devolvidos aos dois utilizadores são exactamente os mesmos pois não existe qualquer tipo de personalização.

O quarto desafio reside no interesse por parte do utilizador em obter informação mais actualizada possível. Por exemplo, basta dar-se um acontecimento invulgar, que suscite o interesse das pessoas e a primeira reacção que as pessoas têm é ir procurar

a um motor de pesquisa. Para este tipo de acontecimentos não podem ser aplicadas as mesmas técnicas para identificar os documentos mais relevantes. Já para não falar que é extremamente difícil identificar este tipo de eventos.

Por último, outro problema existente nos sistemas de WebIR está relacionado com os sinónimos (diferentes palavras com o mesmo significado) e palavras polissémicas (uma palavra pode ter significados diferentes) presentes nas mais variadas línguas. Isto faz com que o significado de uma querie nem sempre seja explícito. Por exemplo, se um utilizador procurar num motor de pesquisa pela palavra "sol"pode pretender resultados relativos à principal estrela do nosso Sistema Solar, ao jornal "O SOL"ou pode estar a referir-se a uma nota musical. Este problema vai fazer com que sejam devolvidas ao utilizador inúmeras páginas sobre vários temas o que se torna pouco interessante do ponto de vista da experiência do utilizador.

1.2 Nova geração de motores de pesquisa

Para resolver os problemas referidos anteriormente torna-se indispensável encontrar novos métodos capazes de responder e satisfazer os objectivos básicos de qualquer motor de pesquisa: devolver os resultados mais relevantes para a querie realizada pelo utilizador.

Após uma análise exaustiva à literatura existente somos capazes de constatar uma grande contribuição a este nível por parte de muitos autores, onde múltiplas teorias e experiências fazem crer que os sistemas de WebIR poderão voltar a atingir com sucesso as funções para os quais foram criados. Existem várias abordagens, mas as três mais importantes são: ordenação [5] [6] [7], categorização [3] [6] [7] [8] [9] e personalização [3] [10] [11] [12] [13] [14] [15]. Existem algumas abordagens muito interessantes em que são utilizadas as referidas em cima aos pares, tais como, ordenação e categorização [6] [7], ordenação e personalização [10] [5] [11] [13] [14] [15] [16] [17], categorização e personalização [3] mas não há muitas que façam o uso colaborativo de todas.

A categorização, ou clustering, pode ser definida como o processo de agrupamento de documentos existentes num conjunto inicial em subcategorias que partilham propriedades semelhantes. No caso de um meta-motor de busca, como acontece no âmbito desta tese, a categorização é aplicada depois dos resultados dos vários motores de pesquisa terem sido devolvidos. É então aplicado o algoritmo de categorização de

modo a formar subcategorias do conjunto inicial dando assim hipótese ao utilizador de fazer filtragem dos resultados por categoria [8]. A categorização é um processo essencial quando se trata de estruturar conjuntos de informação muitos vastos e onde existem semelhanças. É uma forma mais simples, rápida e clara do utilizador aceder à informação sem ter que passar por documentos que não lhe interessam.

A personalização é o processo de adaptação de qualquer conteúdo a uma entidade específica, neste caso o utilizador do motor de pesquisa. No contexto dos sistemas de WebIR significa realçar os documentos que o utilizador prefere ou num caso perfeito devolver apenas os resultados que o utilizador necessita. Obviamente este processo necessita que o sistema recolha informação sobre o utilizador. Imagine-se por exemplo o caso em que existem dois utilizadores, ambos fazem uma pesquisa por "música"mas um prefere música Rock e outro, música POP. Num sistema onde exista personalização os resultados seriam diferentes para cada um deles sendo que para o primeiro os resultados seriam sobre música Rock e para o segundo sobre música POP.

Como seria de prever este método tem alguns problemas ao nível de privacidade do utilizador pois a recolha de informação de um utilizador levanta sempre questões éticas [18]. Daí não ser muito utilizado nos motores de pesquisa mais visitados. No entanto começam a surgir motores de pesquisa que utilizam este tipo de mecanismos mas ainda em fase de protótipo.

A ordenação é um processo que normalmente é aplicado em conjunto com um dos outros métodos referidos anteriormente. É mais frequentemente aplicada juntamente com a personalização de modo que os resultados que estão mais de acordo com o utilizador sejam reordenados de acordo com o seu perfil.

Actualmente, os principais motores de pesquisa devolvem o mesmo resultado independentemente da pessoa que efectua a pesquisa. É aqui que pretendemos inovar e propor um sistema que combina a categorização, a personalização e a ordenação de forma a chegar ao resultado final da pesquisa de informação personalizada.

1.3 Objectivos

O objectivo desta tese é o de melhorar os resultados devolvidos por um sistema de recolha de informação, o VIPAccess [19], que já utiliza categorização nos seus processos, introduzindo a personalização dos resultados devolvidos. Desta forma esperamos obter

resultados mais de acordo com os interesses do utilizador evitando que este perca tanto tempo à procura dos documentos que lhe interessam.

O funcionamento de um motor de pesquisa implica os seguintes processos: crawling, indexação de documentos, procura e ordenação dos resultados. Os processos de personalização e de categorização poderiam ser efectuados em qualquer um dos passos referidos anteriormente mas porque estes métodos irão ser integrados no meta-motor de busca VIPAccess, iremos aplicá-los depois de recebermos os resultados dos motores de busca, isto é, em tempo real. Ou seja, recebemos os resultados dos motores de busca, aplicamos o algoritmo de personalização e em seguida os resultados são reordenados de maneira a termos nos primeiros lugares os resultados que estão mais de acordo com os interesses do utilizador.

De modo a podermos criar um perfil de utilizador iremos guardar o histórico de utilização do meta-motor de pesquisa pelo utilizador, ou seja, serão armazenadas as queries efectuadas, as categorias mais vistas, os documentos mais lidos entre outras informações. Este processo será explicado mais detalhadamente no capítulo 3. Desta forma, com o uso deste modelo de utilizador, o sistema pode reordenar os resultados e as categorias de acordo com o perfil do utilizador em questão.

1.4 Organização da tese

No capítulo 2 será efectuada uma análise do estado da arte dos sistemas de recolha de informação. Numa primeira fase será estudado o seu funcionamento geral, avançando posteriormente para uma fase de análise mais profunda dos processos utilizados para a criação de perfis. No final deste capítulo será também apresentado o trabalho proposto no âmbito desta tese.

O capítulo 3 apresenta todo o trabalho desenvolvido ao longo desta tese, mais propriamente a criação de um perfil do nível de interesse do utilizador e um perfil do nível de conhecimento do utilizador.

No capítulo 4 será abordado o processo de reordenação dos resultados de acordo com os perfis de utilizador criados.

No capítulo 5 da tese serão apresentados e discutidos os resultados do trabalho.

Finalmente, o capítulo 6 apresenta as conclusões e o trabalho a ser efectuado no futuro.

1.5 Contribuições

As maiores contribuições desta tese em relação ao estado da arte são:

1. Combinação da categorização, da ordenação e da personalização
2. Ordenação de categorias
3. Construção automática do modelo de interesse do utilizador utilizando o formalismo da Pré-topologia
4. Construção automática do modelo de conhecimento do utilizador
5. Ordenação dos documentos por perfis de interesse e perfis de conhecimento

Capítulo 2

Estado da Arte

2.1 Sistemas de recolha de informação

Desde sempre que as pessoas tiveram a consciência da importância de encontrar e armazenar informação de modo a passar o conhecimento de geração em geração, mas esta não era de modo algum uma tarefa fácil. Daí o facto de ao longo da história se terem perdido documentos muito importantes sobre o nosso passado. Com o surgimento da era dos computadores, tornou-se possível guardar grandes quantidades de informação e por conseguinte também se tornou mais simples procurar e encontrar informação útil dentro destas bases de conhecimento. Foi assim que nasceu o campo dos sistemas de recolha de informação em 1945 [17]. Em 1975, Gerard Salton desenvolveu o primeiro sistema de recolha de informação relevante na área, o SMART [20]. Este era um protótipo de um sistema de recolha de informação que serviu para testar diferentes algoritmos que automaticamente indexavam documentos e devolviam uma lista de documentos relevantes para uma dada query. Eram utilizados métodos que ainda são utilizados hoje em dia como por exemplo, a análise estatística de texto a partir da métrica TF-IDF, extracção da raiz das palavras ou remoção de palavras que não representam conhecimento. A diferença é que estes algoritmos eram aplicados em ambientes controlados.

Com o surgimento da Web este paradigma alterou por completo. Os conteúdos passaram a ser dinâmicos, subjectivos, heterogéneos, distribuídos, já para não falar do crescimento exponencial das fontes que disponibilizam estes documentos. De modo a reagir a este nova realidade apareceram os sistemas de recolha de informação na Web

(WebIR).

2.1.1 Os sistemas WebIR

Os sistemas de recuperação de informação tradicionais propuseram diversos modelos para representar o conteúdo dos documentos [1], mas os mais conhecidos são os booleanos, probabilísticos, de inferência e de espaço vectorial. Os três últimos são os mais utilizados pois permitem calcular e atribuir um valor de interesse para cada documento tendo em conta uma querie, o que possibilita uma posterior ordenação destes mesmos. É contudo, o modelo de espaço vectorial [21] aquele que merece mais destaque devido à sua maior aceitação por parte dos sistemas de recolha de informação.

Como é do conhecimento geral, a Web é hoje a maior fonte de informação do mundo. É lá que encontramos todo o tipo de informação, desde notícias, vídeos, música, jogos, entre outros. A dificuldade numa rede tão vasta e em constante actualização é mesmo encontrar aquilo que pretendemos. Os sistemas de recolha de informação modernos permitiram através da exploração de algumas metodologias clássicas de IR, desenvolver métodos inovadores capazes de encontrar mais rapidamente informação relevante neste mundo "infinito"que é a Web.

Uma evolução deste tipo de sistemas é o já referido anteriormente, *PageRank*. Este sistema permite atribuir pesos a documentos tendo em conta a relação entre estes. Foi desenvolvido por Larry Page e Sergei Brin, fundadores do Google, em 1998 quando ainda estudavam na Universidade de Stanford. O Google tem indexados vários biliões de páginas ordenadas pelo resultado do *PageRank*. Este resultado é calculado através do número de ligações existentes noutras páginas para a página em questão. O *PageRank* é um critério de ordenação de páginas muito democrático pois reflecte o verdadeiro comportamento da Internet. Obviamente, tem alguns problemas porque pode ser manipulado através da colocação de links descontextualizados em várias páginas, fazendo com que páginas pouco importantes ganhem mais importância.

Um estudo de 2002 [22] mostra que 80% dos utilizadores que navegam na Web encontram os sites que procuram. Mais tarde voltam a visitar os mesmos sites por intermédio de motores de pesquisa, tais como o Google [1], Yahoo [2], Bing [3], entre outros.

[1] http://www.google.com
[2] http://www.yahoo.com
[3] http://www.bing.com

É um facto que hoje em dia as pessoas utilizam os motores de busca para tudo e é isto que faz com que a recolha de informação seja nos nossos dias uma das áreas de pesquisa mais importantes na comunidade científica dos sistemas de informação.

2.1.2 Estrutura de um motor de pesquisa Web

O processo de execução de um sistema de recolha de informação é composto na sua essência por 4 passos: extracção de documentos, indexação, procura e ordenação. Estes passos são posteriormente divididos em passos mais pequenos, tal como podemos ver na figura 2.1.

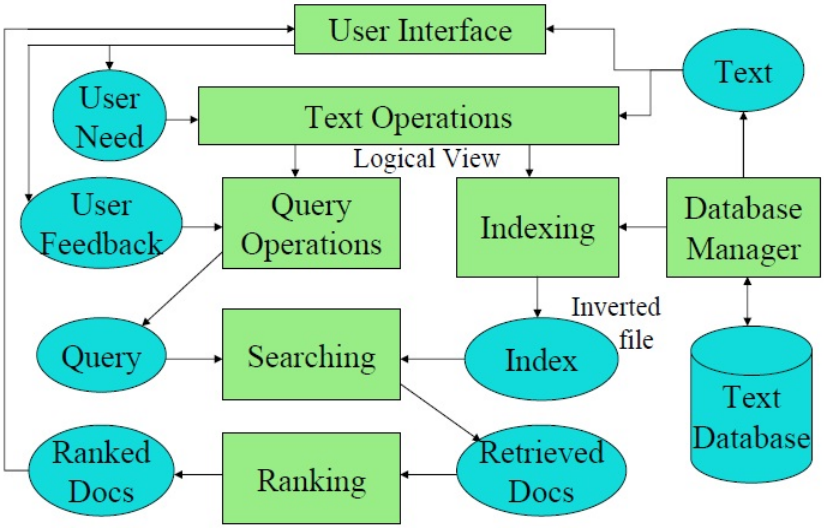

Figura 2.1: Funcionamento de um sistema de recolha de informação

A primeira fase é mais conhecida na comunidade científica por "crawling"e consiste em percorrer a Web ou parte dela e reunir o conteúdo dos documentos guardando-os num repositório algures. Este trabalho é realizado por agentes controlados que operam de diversas formas conforme as regras impostas pela entidade que os controla. Este tipo de trabalho só é possível graças à computação distribuída e por agentes. Após os documentos terem sido armazenados no repositório vão ser processados e indexados numa estrutura especial (o índice invertido para o modelo de espaço vectorial, por exemplo) de forma a permitir um acesso mais rápido à estrutura e conteúdo dos mesmos. No caso do *PageRank* [23], os documentos formam um grafo com as suas hiperligações. Em particular, os documentos são representados por vértices e as hiperligações por arestas. Esta estrutura é depois analisada tanto do ponto de vista relacional como

contextual, permitindo assim fazer o ranking dos vários documentos. É na parte da análise relacional que estão presentes muitos dos processos inovadores [1]. Além de considerar os termos contidos no documento e a sua importância relativamente ao conteúdo do documento, são considerados também os termos que representam os textos correspondentes às hiperligações que apontam para este. Isto deve-se ao facto de muitas das vezes as descrições associadas às hiperligações serem boas representações do conteúdo do documento para onde apontam.

2.2 Categorização de documentos

O facto da informação disponível na Web ser tanta e tão desorganizada impõe requisitos muito altos aos sistemas de recolha de informação. Em capítulos anteriores foram abordados os motores de pesquisa que permitem, baseados em variados mecanismos, devolver ao utilizador um conjunto de documentos relacionados com uma determinada querie. Estes sistemas conseguem, de facto, devolver documentos cuja relação com a querie não se questiona. Porém, também é um facto que as queries são muitas vezes ambíguas o que faz com que sejam devolvidos muitos documentos sobre vários temas distintos. Isso implica inevitavelmente que o utilizador irá necessariamente passar necessariamente vários minutos à procura da informação que pretende.

De forma a tentar evitar esta lacuna, alguns motores de pesquisa optaram por recorrer à categorização [6] [7] [8] [9], de modo a possibilitar que o utilizador possa filtrar, por categoria, os documentos que pretende consultar. Os documentos são agrupados automaticamente e em tempo real. Exemplos destes sistemas são o Clusty[4], o VIPAccess[5], SnakeT[6] e carrot2[7]. Estes sistemas são inegavelmente uma ajuda mas não são a solução para todos os problemas. Pois, muitas vezes, os tópicos gerados automaticamente a partir do texto não são suficientemente explícitos do seu conteúdo. Ou seja, o utilizador perde tanto tempo a navegar nas categorias para descobrir o conteúdo que pretende como num sistema normal em que não exista categorização. Uma das únicas excepções é certamente o VIPAccess, desenvolvido pelo Centro de Tecnologia da Linguagem Humana e Bioinformática (HULTIG). A consciência deste

[4] http://search.yippy.com
[5] http://hultig.di.ubi.pt
[6] http://snaket.di.unipi.it
[7] http://search.carrot2.org

facto e a necessidade da comunidade científica avançar nesta área levou a que os modelos de utilizador, já utilizados noutras áreas, fossem adaptados para este propósito.

2.3 Modelos de utilizador e WebIR

Um dos problemas associados aos sistemas de WebIR e mais concretamente aos motores de pesquisa é a dificuldade de adaptar os seus resultados a qualquer utilizador, pois cada um tem os seus interesses, gostos e objectivos. Este facto faz com que um motor de pesquisa possa não conseguir devolver ao utilizador os resultados que mais lhe convêm de uma forma rápida e concisa. Cada um de nós, sem excepção, tem preferências, gostos, hábitos que em toda a sua extensão fazem com que tenhamos comportamentos diferentes mesmo quando estamos no mesmo contexto. Um modelo de utilizador pretende mapear estas características únicas de cada um.

Muitas vezes estamos perante situações de criação de modelos de utilizador e nem nos apercebemos. Por exemplo, quando nos dirigimos a uma consulta médica pela primeira vez e o médico nos faz um inquérito sobre o nosso historial de doenças, doenças na família, sintomas actuais, entre outros, estes registos que normalmente ficam guardados numa ficha clínica permitem criar o nosso perfil. Assim numa segunda visita o médico não demora tanto tempo com o seu diagnóstico pois já conhece o perfil do paciente que em questão.

O perfil de utilizador pode ser, ou não, individual. Existem alguns casos concretos em que são utilizados perfis de conjuntos de utilizadores para estudar comportamentos de grupo. Focando-nos mais concretamente no paradigma dos motores de pesquisa colaborativos interessa-nos saber que pessoas (que têm gostos semelhantes) consultaram determinados documentos. Isto permite-nos adicionar ao nosso sistema uma componente de sugestão de documentos bastante interessante. Um exemplo real de um sistema deste género é o site da Amazon[8] onde podemos ver que nos são sugeridas compras de acordo com o histórico de bens que tenhamos adquirido e também de acordo com o histórico de compras de utilizadores que também tenham comprado este objecto.

O objectivo principal da adaptação dos modelos de utilizador aos sistemas de WebIR é introduzir um conhecimento adicional nos resultados devolvidos pelo motor de pesquisa. Sabendo os interesses do utilizador podemos reordenar os resultados

[8] http://www.amazon.com

devolvidos inicialmente de modo a que estes coincidam mais com o perfil deste. Imaginemos, por exemplo, dois utilizadores que inserem no motor de busca a seguinte querie: "Guns and Roses". Um dos utilizadores tem por hábito fazer pesquisas por concertos musicais e o outro costuma mostrar interesse em saber as letras das músicas. Pois bem, com base nos perfis o sistema de WebIR pode devolver resultados diferentes para os dois utilizadores mesmo sendo a mesma querie.

2.3.1 Construção de um modelo de utilizador

Apesar de nos capítulos anteriores termos referido que os métodos de criação de modelos de utilizador ainda não são muito avançados, tem sido realizada muita investigação para reunir e reconhecer os interesses de um utilizador. Os sistemas construídos para desempenhar as tarefas de criação dos modelos podem recorrer a indicadores de carácter implícito ou explícito. Explícito é quando os utilizadores passam informações para o sistema, indicando quais os documentos que são de interesse. Implícito, se de alguma forma é extraído conhecimento das acções do utilizador de forma transparente para este.

No exemplo referido anteriormente, referente ao questionário efectuado por um médico para a criação de um modelo de utilizador, o utilizador provavelmente não se mostrará incomodado ao responder às perguntas e não se importará de perder alguns minutos a responder às mesmas. Isto deve-se ao facto da pessoa confiar em quem está a fazer o questionário e saber qual a finalidade do inquérito. No contexto da Web já não acontece o mesmo. Quando estamos na Web é raro o utilizador que está disposto a perder o seu tempo a responder a um questionário, que até pode ser pequeno mas parece sempre infindável, mesmo que a sua finalidade seja conhecida. Hoje em dia, qualquer utilizador da Web exige que qualquer tarefa seja feita rapidamente. Outro problema que se levanta aqui é a questão da privacidade. A Web é um mundo enorme e fantástico mas nunca se sabe quem está a "espreitar do outro lado". Por isso acontece que muitos utilizadores não confiam na autoridade da entidade que lhes está a pedir a informação [18]. É por isso que métodos alternativos têm sido estudados para que seja possível devolver ao utilizador os resultados que ele pretende mas sem que ele tenha que perder tempo a fornecer informação pessoal. Para recolher informação sobre o utilizador são utilizadas algumas metodologias que operam sobre os documentos lidos ou sobre um histórico de queries.

Para recolher e guardar automaticamente informação relevante para o utilizador é necessário fazer uma análise aos seus padrões de interacção para com o sistema. Ou seja, o comportamento de utilização do sistema tem que ser analisado cuidadosamente. Só desta forma é possível saber qual a informação importante para este. Para tal é necessária a criação de uma ferramenta, neste caso um browser, que permita a realização normal das suas tarefas mas que registe também as suas acções. Ou seja, temos que ter um browser que nos permita saber que documentos o utilizador leu e os quais achou relevantes. Saber que pesquisas efectuou com resultados considerados úteis é também muito importante para a construção de um modelo de utilizador.

De modo a avaliar se um documento é, ou não relevante, podem ser utilizadas muitas variáveis, tais como:

- O tempo que decorre desde a abertura de um documento até ao seu fecho,
- A movimentação do cursor sobre um documento,
- O conteúdo seleccionado no documento,
- O *scroll* efectuado no documento,
- O número de cliques dado no rato,
- A impressão do documento,
- A adição aos favoritos, ou mesmo até,
- A movimentação dos olhos sobre um documento.

Ao cruzar os valores obtidos por algumas destas variáveis é possível saber que documentos são realmente interessantes para o utilizador em questão. É de realçar que das variáveis referidas em cima existe uma que é mais representativa do interesse do utilizador por um determinado documento. Esta variável é o tempo que decorre desde a abertura de um documento até ao seu fecho como referido em [24] [25] [26] [27] [28].

Morita e Shinoda [24] trabalham sobre um contexto controlado que são os documentos do Usenet News Articles com intuito de tentar identificar o interesse dos utilizadores nestes mesmos artigos. Depois de analisarem algumas variáveis concluem que apenas o tempo disponibilizado pelo utilizador na leitura do artigo é um indicador

de interesse nesse mesmo documento. De salientar também o facto de os documentos pelos quais o utilizador demonstrou menos interesse não foram vistos até ao fim. Ou seja, o facto de terem analisado o *scroll* do documento permitiu-lhes saber que um utilizador não lê um documento até ao final se achar que a informação contida nos primeiros parágrafos não é relevante.

Claypool, Le, Waseda e Brown [26] estudam a correlação entre a indicação explícita de interesse e diversos factores implícitos extraídos a partir do comportamento de navegação do utilizador num browser denominado Curious Browser. As variáveis tomadas em consideração neste estudo são o tempo total passado na visualização de um documento, o tempo perdido a mover o rato, o tempo e quantidade de *scroll* e o número de cliques efectuados nesse documento. Este estudo prova que o tempo perdido na leitura e o tempo e quantidade de *scroll* são bons indicadores do interesse do utilizador no documento. Por outro lado os movimentos do rato parecem ser apenas relevantes para a determinação dos documentos que têm menos importância para o utilizador.

Goecks e Shavlik [27] optaram por uma abordagem mais arrojada e desenharam uma rede neuronal para aprender os interesses dos utilizadores a partir dos cliques no rato e as suas movimentações e também a partir do *scroll* pelos documentos. Concluíram, aliás como todos os outros casos descritos anteriormente, que apenas a monitorização da actividade do rato não era suficiente para detectar uma correlação com o interesse do utilizador num documento.

Chan [28], introduziu algumas métricas para estimar o interesse para cada página visitada por um utilizador. O interesse do utilizador por uma página é definido através de uma função que recebe os seguintes cinco parâmetros de entrada:

- Frequência de visitas a essa mesma página,
- Valor booleano que indica se a página foi marcada como sendo favorita,
- O tempo gasto na página normalizado pelo seu tamanho total,
- O tempo passado desde que a página foi visitada pela última vez,
- O número de hiperligações visitadas sobre o número de hiperligações existentes.

2.3.2 Trabalho relacionado com perfis de utilizador

Aktas, Nacar e Menczer [29] propõem a criação de um sistema de personalização que tem como base o algoritmo *PageRank*. Este novo sistema passa por introduzir no cálculo do algoritmo informação relativa aos interesses do utilizador por certos domínios específicos. Por conseguinte focam-se na análise das características do url de cada página visitada. Na sua implementação foram escolhidas nove categorias, sendo que três são geográficas e as restantes seis são relativas aos tópicos comercial (.com), militar (.mil), governamental (.gov), organizações sem fins lucrativos (.org), organizações da rede (.net) e educacional (.edu). Este sistema de personalização funciona de forma explícita, ou seja, um utilizador tem que especificar os seus interesses sobre a forma de um vector binário correspondente ao interesse, ou não, numa determinada categoria. Dado este vector de entrada, o sistema processa o valor de *PageRank* para cada página tendo como base a comparação do domínio do url. O facto das categorias serem estáticas pode ser um problema pois restringe o perfil de utilizador apenas aos temos escolhidos.

Tamine, Boughanem e Zemirli [11] inferem os interesses do utilizador a partir do histórico das suas procuras. Do ponto de vista destes investigadores, um perfil de utilizador expressa os seus interesses ao longo de um período de tempo, sendo que estes estão presentes no histórico das suas pesquisas sobre a forma de palavras. Afirmam que as palavras mais utilizadas por entre os documentos considerados relevantes são a chave para o sucesso deste método de criação de perfis. A construção de perfis utilizando este método está dividida em dois passos. Primeiro existe um espaço de tempo onde são armazenados os dados de utilização de forma a poderem ser processados, dando assim origem ao modelo do utilizador. A segunda etapa consiste na monitorização de uma possível actualização do perfil.

Ferragina e Gulli [3] propõem um tipo de personalização em que não é necessário recolher informação da utilização do motor de pesquisa, afastando assim os problemas relacionados com a privacidade. A sua forma de personalização passa por permitir ao utilizador seleccionar tópicos existentes na árvore de conceitos gerada no processo de categorização, ou clustering. Desta forma, o utilizador acaba por criar o seu perfil de utilizador sem se aperceber.

Sieg, Mobasher e Burke [12] propõem um sistema baseado em ontologias como representação dos interesses do utilizador. Cada ontologia é inicialmente uma instân-

cia de uma ontologia de referência constituída por vários conceitos. Conceitos estes que inicialmente terão valor igual a 1. À medida que o utilizador vai interagindo com o sistema a ontologia é actualizada e os valores para o grau de interesse de cada conceito são modificados através de uma acção de propagação. A escolha dos conceitos da ontologia de base foi baseada no Open Directory Project[9], que é organizado numa hierarquia de tópicos e páginas relacionadas com esses mesmos tópicos. A ligação de termos aos conceitos presentes na ontologia é feita através do cálculo dos termos mais relevantes tendo como base a medida TF-IDF [21]. Os resultados devolvidos são ordenados por ordem crescente de valor de importância. Este valor é calculado com base da multiplicação de três outros valores: a distância semântica do documento à querie (a medida de similaridade Cosine), o valor de interesse para o melhor conceito encontrado para o documento e o valor da distância entre o conceito e a querie.

A abordagem de Tanner [5] para a personalização dos motores de pesquisa aponta para a criação de uma hierarquia de interesses do utilizador. O método aqui consiste na consulta do histórico das páginas favoritas do utilizador. Após reunir o conteúdo de todas as páginas, efectua processos de remoção de palavras sem significado (ou palavras vazias) e posteriormente stemming. A partir deste momento, cada documento passa a ser representado por frases cujos termos são todos significativos. Ao conjunto das palavras destas frases é então aplicado um algoritmo denominado "Divisive Hierarchical Clustering"cujo resultado é uma árvore de interesses. A raiz desta árvore é um cluster com todas as palavras, sendo que os seus filhos são subconjuntos do cluster pai. O valor para cada documento é então calculado através do somatório do peso atribuído a cada termo existente na árvore e no documento.

2.4 Modelos de conhecimento

A pesquisa na área da complexidade de textos teve o seu ponto alto entre 1930 e 1960 quando foram descobertos grande parte dos métodos clássicos. Métodos estes que ainda se utilizam hoje em dia. Os modelos de conhecimento originados a partir dos métodos clássicos são hoje em dia vastamente utilizados e são essenciais em sistemas onde a personalização está presente, pois são de fácil construção e funcionam muito bem quando aplicados a sistemas de recolha de informação, como por exemplo, um

[9] http://www.dmoz.org

motor de busca.

Mas estes sistemas têm também muita importância noutras àreas, tais como a educação [30]. A criação de um modelo de conhecimento para um aluno é um indicador que pode ajudar um professor a determinar a evolução deste em temas como a escrita ou a leitura. Se um aluno demorar muito tempo a ler um texto quando comparado com os seus colegas isso aponta para um evolução inferior aos seus colegas podendo significar dificuldades a determinados níveis.

A classificação do nível de complexidade pode ser definida de várias formas, mas a que eu, pessoalmente, acho mais correcta é a proferida por Björnsson em 1971 [31]: "É a soma das propriedades linguísticas num texto, que o tornam mais ou menos compreensível para o leitor".

De acordo com Klare [32] o termo complexidade de leitura pode ser aplicado de três maneiras distintas na área da investigação:

1. Para indicar a legibilidade da leitura ou escrita,
2. Para indicar a facilidade da leitura dado o nível de interesse ou o prazer da escrita,
3. Para indicar a facilidade de compreensão devido ao estilo da escrita.

No âmbito desta tese vamos focar-nos no terceiro ponto desta enumeração. Logo quando nos referirmos à complexidade de leitura será sempre a complexidade relativa às palavras e à formação do texto em análise.

De seguida será descrito mais em detalhe como é criado um modelo de conhecimento do utilizador e serão também mostrados alguns trabalhos existentes nesta área.

2.4.1 Construção de um modelo de conhecimento do utilizador

A construção de um modelo de conhecimento do utilizador é, como já foi referido anteriormente, um processo simples mas importante num sistema de recolha de informação. Antes de mais é importante esclarecer o que é um modelo de conhecimento do utilizador e como é construído.

Para construir um modelo de conhecimento do utilizador é necessário analisar os documentos vistos por este utilizador e classificá-los em níveis de complexidade.

Para isso existem as fórmulas tradicionais ou alguns métodos automáticos, um pouco mais avançados, mas que requerem treino [30]. Ou seja, para utilizarmos os métodos automáticos temos que os treinar a partir de textos que já tenham sido classificados anteriormente em diferentes níveis e aperfeiçoar os seus parâmetros de modo a que os resultados sejam os mais parecidos com a classificação original.

Um dos grandes problemas da criação de modelos de conhecimento de um utilizador reside no facto de grande parte dos algoritmos clássicos serem dependentes da língua. Este facto torna impossível adaptar alguns métodos que funcionam muito bem para, por exemplo, o Alemão ao Sueco. Este facto fez com que as pesquisas mais recentes se afastassem um pouco dos métodos clássicos e recorressem a técnicas mais complexas como por exemplo, as redes neuronais.

2.4.2 Trabalho relacionado com modelos de conhecimento

Como já foi referido anteriormente existem duas formas distintas de análise de complexidade de documentos: as fórmulas clássicas e os métodos de classificação por meio de aprendizagem.

As fórmulas clássicas concentram-se basicamente numa tentativa de classificar as dificuldades de leitura ao nível da palavra ou até mesmo da frase. Quase todas as fórmulas clássicas contêm parâmetros que representam a complexidade semântica ou sintáctica. O output destas fórmulas é um valor que reflecte o grau de complexidade de um documento ou um valor que indica o nível de escolaridade que um utilizador tem que ter para conseguir perceber o documento. As fórmulas clássicas mais conhecidas são cinco [30].

A primeira é a de Dalle-Challs(1948) revista em 1995, classifica os textos em anos escolares(3-12), segundo a fórmula 2.1.

$$Ano = 0.596sl + 0.1579w_d + 3.6365 \tag{2.1}$$

- sl = comprimento médio de uma frase.

- w_d = número de palavras que não ocorrem na lista de 3000 palavras de Dale [33].

Um ponto que torna esta fórmula muito pouco utilizada é o facto de consumir

muito tempo pois, para cada palavra processada temos que comparar com a lista das 3000 palavras.

A segunda é a fórmula de Lorges (1939) revista em 1948, que classifica os textos em ano escolares(3-12) tal como a anterior. Esta fórmula é representada pela expressão 2.2.

$$Ano = 0.07sl + 0.1073w_d + 0.1301pp + 1.6126 \tag{2.2}$$

- sl = comprimento médio de uma frase.
- w_d = número de palavras difíceis diferentes por cada 100 palavras. As palavras consideradas difícies são todas aquelas que não estão na lista de 769 palavras de Dale [33].
- pp = número de frases com preposições em cada 100 palavras.

Esta é considerada uma das melhores fórmulas de entre as primeiras fórmulas a ser criadas. O facto de ser muito fácil de utilizar tornou-a bastante popular.

A terceira fórmula é a fórmula de Flesch Readin Ease (1948) revista várias vezes ao longo dos anos e devolve uma pontuação onde o valor mais alto é o texto mais difícil de ler. (Ver equação 2.3)

$$Complexidade = 206.835 - 1.015sl - 0.846wl \tag{2.3}$$

- sl = número médio de palavras por frase.
- wl = número de sílabas por 100 palavras.

Esta fórmula devolve um número entre 0-100, onde o valor mais alto indica que o documento é mais difícil de ler. Esta é uma fórmula bastante utilizada pois só necessita que o texto tenha 100 palavras e só tem dois critérios para verificar. É o método utilizado pelo governo dos Estados Unidos da América em grande parte dos seus sistemas.

A quarta é a fórmula de Flesh-Kincaid Grade Level (1975). Esta fórmula classifica os textos em anos escolares Americanos através da fórmula 2.4.

$$Ano = 0.39sl + 11.8wl - 15.59 \tag{2.4}$$

- sl = número médio de palavras por frase.

- wl = número de sílabas por palavra.

Esta fórmula é uma modificação da anterior. Esta modificação permite receber desta fórmula directamente um resultado referente ao sistema de ensino Americano.

A última é conhecida LIX(1968). Foi desenvolvida para a língua Sueca e é representada pela fórmula 2.5.

$$LIX = \frac{wl}{s} + 100 \times \frac{w_d}{wl} \tag{2.5}$$

- wl = número de palavras no documento.

- s = número de frases no documento.

- w_d = número de palavras difíceis no documento, onde são entendidas por palavras difíceis todas aquelas que tenham mais de 6 letras.

A escala do LIX tem que ser verificada a partir de uma tabela designada por *LIX-interpreter* apresentada na tabela 2.1. A vantagem da fórmula LIX é o facto de poder ser aplicada facilmente a outra língua bastando para isso modificar a escala.

Tabela 2.1: Lix-interpreter

Valor	Descrição
20	Muito Fácil
30	Fácil
40	Médio
50	Difícil
60	Muito Difícil

Relativamente aos métodos de classificação por meio de aprendizagem existem três estudos com resultados significativos.

O primeiro é *Language Modeling Aproach to Predicting Reading Dificulty* [34]. Este é uma tentativa de resolver os problemas de classificação em níveis de complexidade. Isto é conseguido utilizando um classificador Naive-Bayes multinomial baseado em unigramas de palavras. Este modelo teve resultados superiores aos

métodos clássicos para textos recolhidos da Web mas para textos retirados de livros os resultados não foram tão bons.

O segundo estudo é *Automatic Recognition of Reading Levels from User Queries* [35]. Tal como o método apresentado anteriormente este método, tenta classificar textos em níveis de complexidade. Mas desta vez baseia-se nas queries realizadas no motor de pesquisa. Isto requer que o modelo seja treinado sobre queries completas, pois as queries feitas por um utilizador no seu dia-a-dia são frases pequenas e normalmente incompletas. O modelo é induzido a partir de SVM treinadas sobre um número de características semânticas e sintácticas derivadas das queries realizadas pelo utilizador. Exemplos destas características são o comprimento de uma frase, número médio de sílabas por palavra, entre outras.

O *Coh-Metrix* citebib37, tenta criar um método baseado em dois conceitos principais: coesão e coerência do texto. Estudos recentes na área da psicologia e linguística [36] [37] mostram que um facto importante na compreensão de um texto é a coesão do mesmo. A coesão é o nível de relação entre os vários componentes do texto. Este método é um dos mais complexos de utilizar pois requer muitos conhecimentos ao nível da análise semântica do texto.

2.5 Reordenação de resultados

Os métodos de reordenação dos resultados devolvidos por um sistema de recolha de informação são nos nossos dias imensos e as abordagens aos mesmos são também de uma enorme variedade. Este facto tem inerente a ideia que ainda não foi encontrado um método que tenha resultados minimamente aceitáveis.

Agichtein, Brill e Dumais [17] utilizam métodos explícitos de ordenação de resultados, ou seja, o utilizador vai participar no processo de recolha de informação, dando informações sobre si e ajudando o sistema a criar o seu perfil e consequente ordenação dos resultados. Mas estes autores vão ainda mais longe, pois estudam mesmo a efectividade de alguns métodos de aprendizagem automática para fazer a ordenação.

Como já vimos, Sieg, Mobasher e Burke [38] recorrem a um método de criação de perfis baseado num perfil de base e posteriormente fazem a ordenação dos documentos recorrendo a um algoritmo que se baseia na similaridade de conceitos. Ou seja, este

método não se baseia apenas nas palavras mas tem também uma noção do contexto associado às palavras. A base de comparação é a ontologia criada no primeiro passo.

Pretschner e Gauch [14] fazem a reordenação dos resultados devolvidos por um motor de pesquisa público, neste caso o ProFusion[10]. Os autores analisam todos os documentos devolvidos e tentam encontrar um tema para cada um destes. De seguida tentam encontrar uma relação entre este tema e os interesses existentes no perfil de utilizador criado.

Da bibliografia que foi estudada e analisada este três documentos são aqueles que implementam soluções mais inovadoras daí terem sido os escolhidos para evidenciar no estado da arte. Existiram outros documentos analisados mas, por questões de fracos resultados na avaliação ou por questões de semelhança com alguns dos referidos anteriormente, foram deixados de fora. Estes documentos são no entanto referidos na bibliografia em [11] [15] [16] [25] [28].

2.6 Proposta de trabalho

Como foi referido nos capítulos anteriores existem 3 metodologias que estão em voga na área dos motores de pesquisa. Estas são a categorização, a personalização e a ordenação. Visto que todo o trabalho desta tese será integrado no sistema VIPAccess que já vem a ser trabalhado há algum tempo pelo HULTIG e que já contém a componente da categorização, iremos focar-nos, nesta tese, como é natural nos outros dois temas: a personalização e a ordenação.

Nesta tese, o que nos propomos a fazer é uma componente de personalização para um meta-motor de busca que será independente da língua, onde o perfil de utilizador será gerado automaticamente sem uma ontologia de base, seguindo a ideia de [5]. Será também criado um perfil de conhecimento geral do utilizador. E a reordenação dos resultados será feita por categoria e por grau de generalidade ou especificidade.

Depois de ter estudado exaustivamente o estado da arte sobre classificação, personalização e ordenação propomos um sistema inovador a vários níveis. Em primeiro lugar, introduzimos uma nova metodologia para construir micro-ontologias de utilizadores de forma completamente automática baseada no formalismo da Pré-topologia [39]. Em segundo lugar, introduzimos a noção de conhecimento do utilizador que poderá

[10] http://ww.profusion.com

vir a ser aplicado a cada ramo da micro-ontologia de cada utilizador. Em terceiro lugar, propomos a ordenação dos resultados por categorias e documentos seguindo uma estratégia dupla de organização por grau de generalidade (o utilizador vê em primeiro lugar os documentos mais gerais de acordo com o seu interesse) ou por grau de especificidade (os mais específicos de acordo com o seu interesse). Este sistema tem como particularidade ser independente da língua, não supervisionado e juntar grau de interesse e grau de conhecimento no processo de ordenação. Consequentemente, a sua aplicação na área da recolha de informação em dispositivos móveis (Mobile Information Retrieval) é evidente. Neste âmbito, iremos propor um protótipo desenvolvido para a plataforma Windows Phone 7. Do nosso conhecimento, este trabalho atinge um grau de personalização ainda não experimentado por nenhuma equipa de investigação, o que abre um conjunto de novas direcções de investigação na área da personalização Web.

Capítulo 3

Metodologia

3.1 Criação do modelo de utilizador

3.1.1 Recolha dos dados

Tal como foi referido em capítulos anteriores, para que seja possível criar um modelo de utilizador é necessário recolher informação que seja realmente relevante para este. Existem duas formas de o fazer, implicitamente ou explicitamente. O problema de reunir dados implicitamente reside na escolha dos melhores indicadores que permitam realmente extrair informação útil. Por outro lado, o problema de reunir informação explicitamente é uma questão complicada pois não temos qualquer tipo de certeza que o utilizador vá colaborar.

A nossa escolha passou então pela criação de um sistema de recolha implícito, no qual o utilizador não se irá aperceber que os dados estão a ser recolhidos. No entanto, de forma a prevenir problemas de privacidade (que são limitados em dispositivos móveis) iremos aplicar recomendações explicitadas em [18], como por exemplo alertas e possibilidade de remoção dos dados privados. Muitos estudos [24] [25] [26] [27] [28] revelam que os factores implícitos indicadores de maior interesse de um utilizador num documento são o tempo que este passa a ver um documento e as movimentações do cursor no mesmo. A nossa opção para saber se uma querie foi, ou não, produtiva passa, nesta fase do trabalho, pelo tempo dispendido pelo utilizador na leitura de um documento. Assim, se em pelo menos um dos documentos abertos pelo utilizador passou mais do que s segundos na sua leitura, então este será adicionado ao histórico

da sessão de pesquisa do utilizador.

O termo sessão ainda não tinha sido referido ao longo desta tese mas é um relativamente importante pois refere-se ao tempo que passa desde que o utilizador abre a aplicação até a encerrar. Durante este tempo toda a informação do utilizador é guardada localmente e só depois da sessão terminar, ou seja, quando o utilizador fechar a aplicação é que os dados da sessão são enviados de forma segura para o servidor através de um serviço com a finalidade de serem armazenados na base de dados. Em cada sessão são guardadas as queries relevantes (que originam resultados que o utilizador consulta durante um tempo igual ou superior a *s* segundos) do utilizador, bem como as categorias associadas a estas queries. São guardadas também as categorias escolhidas pelo utilizador através do processo de filtragem de resultados. Estas categorias vão mais tarde ser utilizadas no processo de criação do modelo de utilizador. Obviamente, os documentos visitados são também guardados.

Após os dados da sessão serem recebidos no servidor, este vai começar o processamento dos mesmos de modo a não haver informação replicada ou desactualizada. Assim sendo vamos armazenar os resultados em quatro tabelas diferentes. A primeira contém as queries efectuadas e quais as categorias associadas às mesmas. A segunda contém as queries efectuadas, as páginas visitadas no decorrer desta pesquisa e também o nível de complexidade desta página. A terceira contém as queries realizadas por determinado utilizador e a frequência com o utilizador efectua esta querie. E por último, uma tabela que contém todas as palavras existentes em queries relevantes efectuadas por determinado utilizador, assim como a frequência com que esta palavra ocorre, a primeira e última data de registo da mesma. O processo de recolha dos dados está visível na figura 3.1.

3.1.2 Construção da Micro-Ontologia

Como foi referido em capítulos anteriores o processo de criação de um perfil de utilizador é realizado "offline", ou seja, é realizado na parte do servidor e não quando um utilizador faz uma pesquisa. É um processo que corre com uma determinada frequência e cujo intervalo de tempo pode ser facilmente modificado pois é um parâmetro da nossa aplicação como pode ser visto na figura 4.1. No nosso caso optámos por definir um intervalo de 2 meses pois verificámos que para termos uma boa ontologia necessitamos de ter uma quantidade significativa de dados. Este processo pode ser pesado em termos

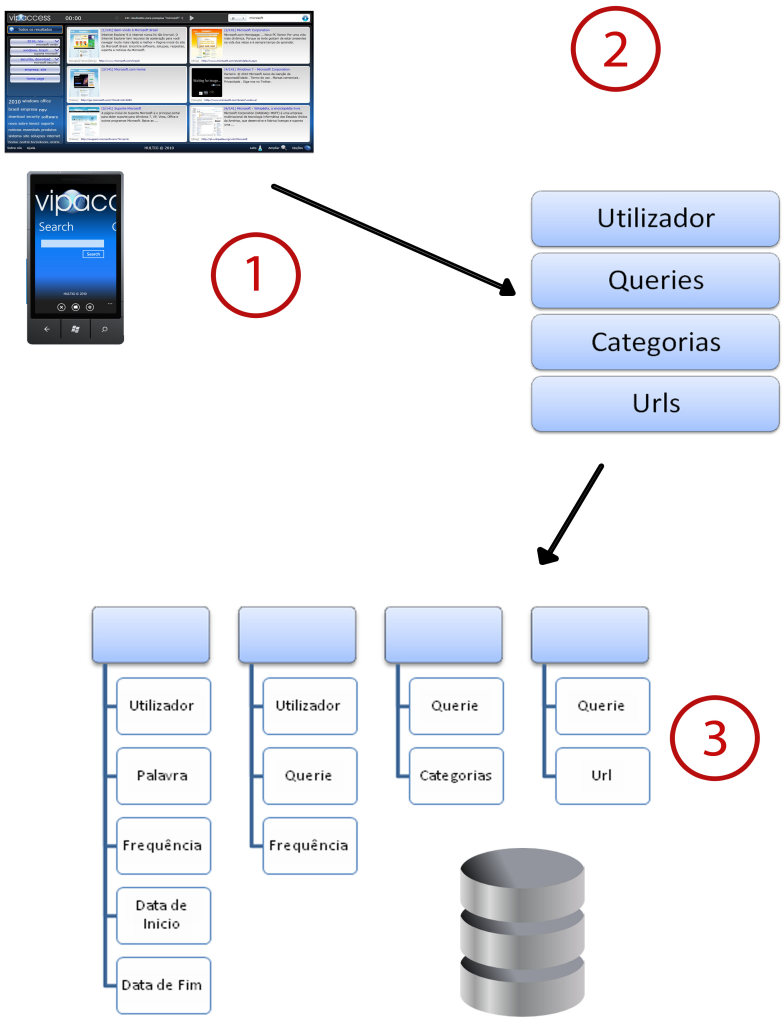

Figura 3.1: Diagrama do processo de recolha de informação sobre o utilizador

de computação e ao nível de tempo, o que não é minimamente preocupante pelo facto de não ser executado em tempo real. Mesmo assim, a complexidade dos algoritmos utilizados no processo é no máximo quadrática.

O processo de criação do perfil passa por quatro fases essenciais. A primeira fase (recolha de informação) consiste em todo o processo explicado no subcapítulo anterior.

A segunda é a extracção das palavras mais relevantes nos documentos dentro do contexto do utilizador. Aqui utilizamos um método já existente no sistema [40] e que foi utilizado anteriormente para definir as categorias dos snippets dos resultados. A diferença aqui é que este método vai ser aplicado ao conteúdo total de um documento. Este método é independente da língua o que é uma vantagem enorme. De modo

a conseguirmos aplicar este algoritmo primeiro passamos por uma fase onde vamos extrair o texto útil de um documento utilizando um parser de HTML. Em seguida aplicamos o algoritmo em questão e recebemos como resultado um conjunto de palavras simples ou compostas que iremos utilizar no segundo passo. Estes resultados já eram consideravelmente bons na criação das categorias pois conseguíamos através da análise de um snippet (quantidade de texto relativamente pequena) obter palavras bastante fortes tendo em conta o seu conteúdo. Agora aplicado a um número maior de palavras conseguimos resultados ainda mais parecidos com o que deve ser a extracção automática de palavras num sistema de recolha de informação comercial.

O terceiro passo é a criação de uma matriz de co-ocorrência assimétrica destas palavras. Aqui juntamos as palavras extraídas dos documentos com as palavras utilizadas nas queries e aquelas das categorias em que o utilizador clicou para chegar aos seus resultados. A matriz assimétrica é obtida através da probabilidade condicionada entre duas palavras. Assim, para duas palavras $w1$ e $w2$ guardamos de forma simétrica na matriz as seguintes probabilidades: $P(w1|w2)$ e $P(w2|w1)$. Esta matriz irá ser uma matriz global, ou seja, referente a todos os documentos armazenados. No caso em que uma relação já esteja presente na matriz é feita uma média entre o peso antigo e o peso novo.

Por fim, é aplicado o algoritmo de Pré-topologia [39] [41] de modo a criar uma micro-ontologia que será no final o modelo do utilizador. A Pré-topologia é um formalismo matemático que utiliza como base o conceito da proximidade. Este permite definir redes complexas a partir de um conjunto de palavras relacionadas com pesos associados. Por isso, no passo anterior criámos uma matriz com as frequências de co-ocorrência assimétrica das palavras. A Pré-topologia é baseada em seis definições:

Considerando o conjunto E um espaço finito não vazio e $P(E)$ a função que designa todos os subconjuntos de E.

1. O espaço pré-topológico é o par (E, a) onde a é a função de mapeamento $a(.) : P(E) \rightarrow P(E)$ chamada de pseudo-fecho (figura 3.2) e é definida como : $\forall A, A \subseteq E$ é o pseudo-fecho de A, $a(A) \subseteq E$ de tal forma que:

   - $a(\emptyset) = \emptyset$
   - $A \subseteq a(A)$

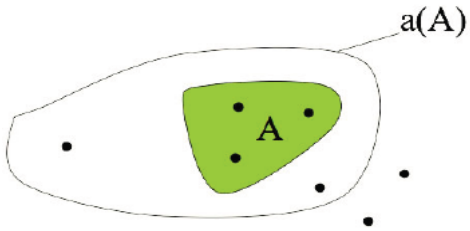

Figura 3.2: Pseudo-fecho de A

2. O espaço pré-topológico $V(E, a)$ é definido por:

$$\forall A, B, A \subseteq E, B \subseteq E e A \subseteq B \text{ então } a(A) \subseteq a(B) \tag{3.1}$$

Este tipo de espaço é muito poderoso para descrever propriedades complexas mas perde muita da sua força pois é baseado em pressupostos fracos.

3. O espaço Pré-topológico $V_d(E, a)$ é definido por:

$$\forall A, B, A \subseteq E, B \subseteq E, a(A \cup B) = a(A) \cup a(B) \tag{3.2}$$

4. O espaço Pré-topológico $V_s(E, a)$ é definido por:

$$\forall A, A \subseteq E, a(A) = \cup_{(x \in A)} a(\{x\}) \tag{3.3}$$

5. $A \in P(E)$ é fechado se e apenas se:

$$A = a(A) \tag{3.4}$$

6. Seja X um conjunto. Seja I um conjunto finito de índices. Seja $\{a_i, i \in I\}$ o conjunto das Pré-topologias em X . A família de espaços Pré-topológicos $\{(X, a_i), i \in I\}$ define a rede em X .

Na altura da escolha das metodologias que iriam ser utilizadas para criar a ontologia optámos por esta pois achámos que era um algoritmo através do qual se obtinham bons resultados e cuja aplicação nesta área ainda era muito pouco significativa. Aliás, não encontrámos nenhum documento onde fosse aplicada a noção de Pré-topologia na criação de perfis de utilizador.

```
Method structure(E : set)
vars: FN : family, FM: family
(note: sets in family are unique)
Begin
  FN = Fe(E, a) − FM(E, a)
  FM = FM(E, a)
  while FM ≠ ∅ do
      take F of FM
      remove F of FM
      successor(F)
  end while
  return extracted_structure
End
```

```
Method successor(F : set)
vars: FF : family
Begin
  FF = {G ∈ FN | / G ⊂ F}
  if FF ≠ ∅ then
      FM_F = MaxClosedSubsets(FF)
      for each V ∈ FM_F do
          V is a successor of F
          successor(V)
      end for
  end if
End
```

Figura 3.3: Algoritmo utilizado na criação da ontologia

Tabela 3.1: Tabela de aderências da Pré-topologia

x	F_x
1	{1, 2, 3, 4, 5, 6}
2	{1, 2, 3, 4, 5, 6}
3	{1, 2, 3, 4, 5, 6}
4	{4, 5, 6}
5	{4, 5, 6}
6	{4, 5, 6}
7	{7, 8}
8	{7, 8}
9	{7, 8, 9}
10	{10}
11	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

O algoritmo utilizado para a criação de ontologias é o apresentado na figura 3.3 que é a versão inversa do algoritmo de [41]. Na tabela 3.1 está a tabela de aderências que nos permite chegar ao resultado final exibido na figura 3.4.

No algoritmo exibido na figura 5 podemos ver uma função chamada *MaxClosed-Subsets*. A forma de calcular o fecho dentro desta função está dependente de dois parâmetros, α e β que medem a distância entre os vários elementos do conjunto. O valor ideal segundo a nossa experiência para α é um valor que ronda os 50% numa escala normalizada de possíveis valores de α. Para β, o valor ideal encontra-se entre os 60% e 80% na escala normalizada de possíveis valores de β.

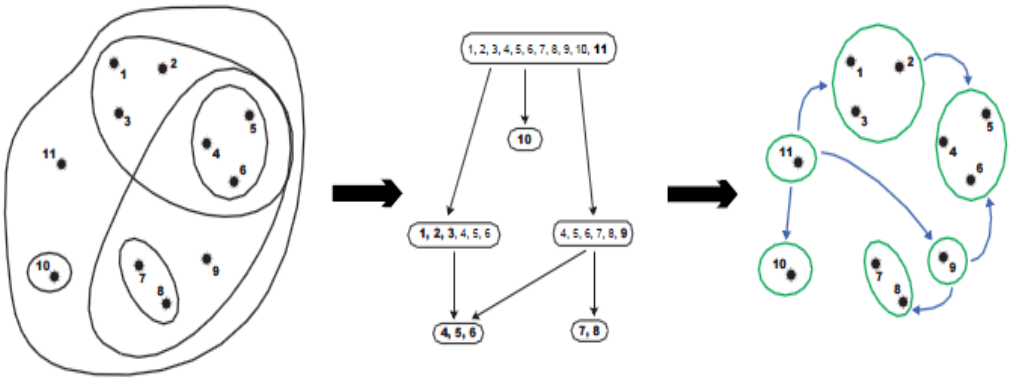

Figura 3.4: Exemplo de utilização do algoritmo de Pré-topologia

Com este método não é possível fazer actualização da ontologia, ou seja, é necessária refazer o processo todo desde o inicio. O que para nós é o ideal pois como sabemos, e já foi referido anteriormente, os interesses de um utilizador não são imutáveis portanto o perfil de utilizador tem que ser gerado com alguma frequência. Assim sendo, e falando em termos de capacidade de computação, torna-se mais rápido e simples criar um perfil de novo em vez de modificar nós da ontologia separadamente. No entanto, trabalho futuro é necessário para ter em conta a análise temporal das micro-ontologias.

3.2 Criação do modelo de conhecimento do utilizador

No processo de criação do modelo do conhecimento do utilizador optámos por utilizar um método tradicional, mais concretamente o LIX (equação 2.5). Este método, como foi referido e explicado detalhadamente no capítulo 2 e apesar de ter sido desenhado para a língua Sueca pode ser aplicado com sucesso noutras línguas como afirma Larsson [30]. Após alguns testes verificámos a veracidade desta afirmação e decidimos mesmo utilizar esta forma por defeito. Esta metodologia é aplicada no nosso sistema da seguinte forma:

- "Cada vez que um documento é processado para a criação da ontologia de um utilizador é processado o conteúdo do mesmo de forma a obter um valor actualizado relativo à informação que este contém,

- "Cada resultado que é devolvido na pesquisa é avaliado ao nível da complexidade do seu texto para questões futuras de ordenação.

A aplicação deste algoritmo nestas duas fases distintas do processamento torna o nosso sistema um pouco pesado, principalmente no segundo ponto pois o primeiro é realizado *"offline"*. Daí termos optado no segundo ponto por avaliar apenas os snippets e não o conteúdo total do documento. Isto permitiu-nos aumentar um pouco o desempenho do nosso sistema.

3.3 Reordenação dos resultados

A reordenação de resultados é um processo muito em voga por parte dos motores de pesquisa, até mesmo ao nível dos motores de pesquisa comerciais. O que não acontece com tanta frequência é a reordenação de resultados consoante um perfil de utilizador. O facto de falarmos em reordenação em vez de ordenação é propositado. Pois no âmbito desta tese é mesmo isso que estamos a fazer, visto que o contexto associado a este projecto é um meta-motor de busca. Ou seja, os resultados que são devolvidos a partir dos vários motores de busca já têm uma ordem. Ordem essa que é imposta pelos motores de pesquisa.

Existem algumas referências no meio académico a sistemas que fazem reordenação de resultados de acordo com um modelo de utilizador [12] [38] [42] mas cada um faz a ordenação de maneira diferente. Daí que tentámos simplificar ao máximo este processo e cingimo-nos à similaridade entre as categorias geradas automaticamente pelo meta-motor de pesquisa e ao perfil gerado também automaticamente pelos processos explicados no capítulo anterior.

O processo aqui é bastante simples e segue um padrão muito básico. O primeiro processo a aplicar é juntar as categorias devolvidas (conjunto A), em seguida utilizamos a micro-ontologia do utilizador (conjunto B) e por fim inicia-se o processo de ordenação por comparação.

A cada elemento do conjunto A atribuímos um peso inicial de 1. Estes pesos vão ser em seguida actualizados de modo a que seja possível ter um ponto por onde a ordenação possa ser feita. A actualização dos pesos é feita da seguinte forma:

- Seleccionamos as palavras que compõem cada categoria devolvida nos resultados

e fazemos uma comparação directa com todos os nós da árvore,

- Caso o resultado da comparação seja positiva este peso é incrementado com um valor C_1N, onde N é o nível da ontologia onde a palavra se encontra e C_1 é uma constante parametrizável,

- Caso o resultado da comparação seja positiva este peso é incrementado com um valor C_2N, onde N é o nível da ontologia onde a palavra se encontra e C_2 é uma constante parametrizável.

O facto de existirem dois valores diferentes, C_1 e C_2, na actualização dos pesos é propositado pois pretendemos atribuir mais importância a uma categoria que se encontra na ontologia, mesmo aparecendo menos vezes. Ao invés de penalizarmos uma categoria que não aparece na ontologia. Este facto implica $C_1 > C_2$.

Relativamente à ordenação das páginas o método que utilizamos apenas se baseia no grau de conhecimento do utilizador, pois era totalmente impensável processar o texto de todas as páginas devolvidas como resultado de uma pesquisa em tempo real. Ou seja, seria necessário aplicar o método explicado em cima para ordenação de categorias a todas as páginas, sendo que cada página teria um número de palavras relativamente grande. Logo, como já referimos em cima, o método de ordenação de páginas é baseado apenas no grau de conhecimento do utilizador e das páginas. Aqui existe outro pormenor que é o facto do meta-motor de busca em questão (VIPAccess) não ter documentos indexados. Isto não nos permite ter todos os documentos devolvidos classificados por grau de conhecimento. Ou seja, apenas as páginas que já se encontrem no sistema irão ser alvo de ordenação. Qualquer outra página não sofrerá qualquer alteração.

Como foi referido o método de reordenação utilizado neste sistema não é de grande complexidade mas tem um desempenho bastante satisfatório. Como iremos ver no capítulo seguinte, onde apresentamos os resultados e fazemos algumas comparações entre o nosso sistema sem o apoio do perfil de utilizador e com o mesmo, existe uma alteração na ordem das categorias correspondente ao perfil do utilizador criado.

Capítulo 4

Resultados e Discussão

A figura apresentada em baixo mostra a forma como os resultados são devolvidos ao utilizador na aplicação criada para a recente plataforma da Microsoft, Windows Phone 7 [43].

Figura 4.1: Exemplo de uma pesquisa no VIPAccess Mobile

As aplicações para Windows Phone 7 (WP7) são todas realizadas em Silverlight [44] o que torna bastante simples a criação de uma aplicação. O Silverlight é uma tecnologia nova da Microsoft para a criação de RIA's (Rich Internet Applications). Uma das grandes vantagens do Silverlight é o facto de ser multi-plataforma e multi-

browser. Assim, corre em sistemas operativos Windows, Apple e Linux (através de um projecto denominado Moonlight) e corre também em todos os browsers existentes, desde o Internet Explorer, Safari, Chrome, Firefox, entre outros. A primeira versão do Silverlight foi lançada em 2006 e foi criada com um simples propósito, facilitar a reprodução de vídeos na Internet. Nesta versão quase toda a programação era feita à base de Javascript. Desde então tem vindo a evoluir bastante a todos os níveis e não há nenhuma tecnologia semelhante que lhe faça frente em termos de versatilidade, rapidez de desenvolvimento e qualidade no resultado final. O facto de estar assente na Framework .NET e ter como ferramentas de desenvolvido o Visual Studio[1] e o Expression Blend[2] contribuem muito para isso. Outro factor é o grande número de controlos que já existem e que aceleram, sem qualquer tipo de dúvida, o desenvolvimento de aplicações.

No âmbito desta tese, e como já foi referido anteriormente, fizemos uma versão do nosso meta-motor de busca, VIPAccess, que já se encontra a correr na Web[3], para Windows Phone 7. Nesta aplicação utilizámos um controlo que já vem com as ferramentas do Windows Phone 7 para o Visual Studio 2010. Este controlo é conhecido como panorama e apresenta o aspecto exibido na figura 4.2.

Uma das vantagens de fazer aplicações para Windows Phone 7 é o conjunto de controlos que o programador tem ao seu dispor inicialmente, o que permite que a curva de aprendizagem/desenvolvimento comece num ponto bastante avançado logo desde os primeiros instantes. Estes controlos têm estilos próprios para aplicações WP7 e estão desenhados para ter uma boa performance num dispositivo Windows Phone 7.

O panorama é basicamente um ecrã "gigante"no qual é possível fazer *scroll* horizontal. Apenas um dos "mini-ecrãs"do *panorama*a está visível em qualquer altura, o que torna este controlo bastante interessante para a utilização que lhe pretendemos atribuir.

Como podemos verificar na figura 4.1, onde mostramos o aspecto da nossa aplicação, o nosso *panorama* é composto por três ecrãs. O primeiro permite-nos fazer a pesquisa por uma querie, no segundo vemos as diversas categorias geradas automaticamente pelo sistema e por último vemos os resultados que estão contidos dentro da categoria seleccionada.

[1] http://www.microsoft.com/visualstudio
[2] http://www.microsoft.com/expression/products/blend_overview.aspx
[3] http://hultig.di.ubi.pt/vipaccess

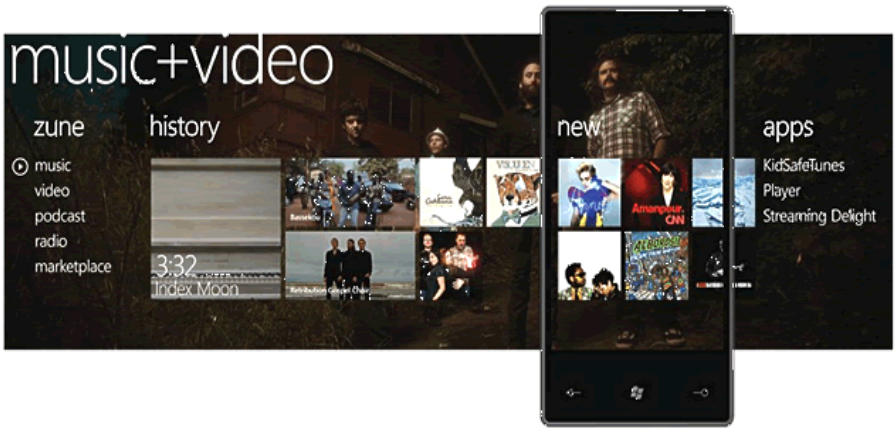

Figura 4.2: Controlo de panorama do Windows Phone 7

Nos subcapítulos que se seguem passamos a mostrar os resultados obtidos no decorrer desta tese e entramos mais em detalhe em algumas das componentes deste sistema.

4.1 Modelo de utilizador

Como referimos no capítulo sobre a Pré-topologia, o resultado deste processo é uma micro-ontologia, que é um conjunto de palavras relacionadas numa estrutura de grupo acíclico direccionado. Durante o processo de criação do perfil conseguimos ter algumas pessoas a utilizar o nosso sistema de uma forma periódica. Assim conseguimos recolher informação sobre a utilização do nosso meta-motor de busca por parte de cada utilizador. Desta forma foi possível criar para cada utilizador um perfil diferente de acordo com os seus interesses.

Na literatura que consultámos para a realização desta tese, percebemos que já há estudos significativos sobre o uso de ontologias neste tipo de sistemas. Contudo, não conseguimos nenhuma referência para um exemplo de uma ontologia que tivesse sido gerada de forma automática. De tal forma que a análise da qualidade dos perfis é baseada num conhecimento prévio do interesse de cada utilizador. Isto vai permitir-nos saber se realmente o perfil gerado tem ou não a qualidade suficiente para podermos afirmar que o nosso método é aplicável no "mundo real".

Dos dois utilizadores que utilizaram o nosso sistema nos últimos tempos regularmente, sabemos que o primeiro é um adepto de futebol que tenta estar sempre ao

corrente de todas as notícias e o segundo elemento é uma pessoa que trabalha em informática, mais concretamente com tecnologias Microsoft. Os perfis gerados para cada um destes utilizadores são apresentados nas figuras 4.3 e 4.4.

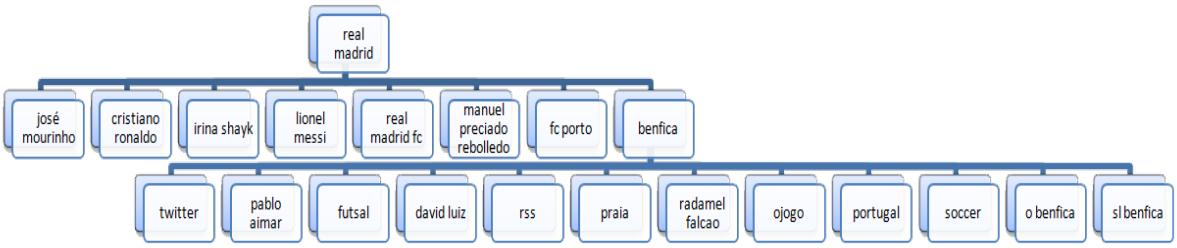

Figura 4.3: Perfil do utilizador 1

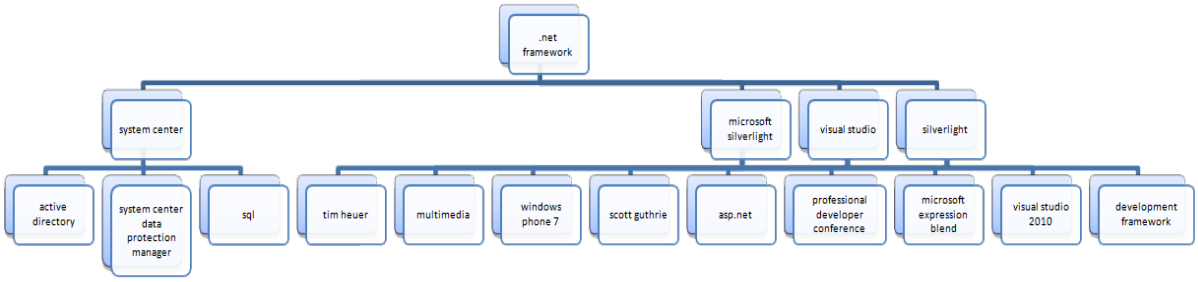

Figura 4.4: Perfil do utilizador 2

Como podemos verificar, os resultados estão relativamente bons, pois aproximam-se do interesse mais geral do utilizador. Obviamente, aparecem outras palavras na ontologia que nada têm a ver com o tema de interesse predominante do utilizador, o que é perfeitamente normal pois nenhuma pessoa pesquisa na Internet apenas sobre um tema. Contudo é bastante perceptível o facto das palavras dentro do contexto do interesse predominante serem as que mais aparecem no perfil.

4.2 Modelo de conhecimento

Os resultados obtidos com a fórmula clássica utilizada para classificar a complexidade dos textos foram bastante bons para textos cujo comprimento seja superior a 30 palavras, ou seja, funciona bem para os casos em que é aplicado ao conteúdo total

de um documento. Este algoritmo comporta-se também de forma aceitável para os snippets devolvidos pelos motores de pesquisa, sendo que aqui os resultados dependem fortemente da complexidade de cada palavra pertencente ao snippet. Se o texto a classificar for relativamente grande e contiver poucas palavras com comprimento igual ou superior a 6 letras (palavra difícil) muito dificilmente o texto será classificado de complexo, apesar de, na realidade, o poder ser.

4.3 Reordenação dos resultados

Ao nível do processo de reordenação dos resultados, como é possível verificar pelas explicações do capítulo anterior, é feita uma comparação entre as categorias devolvidas pelo sistema e a ontologia gerada em cima. Este método foi definido e estruturado por nós e podemos verificar alguns resultados obtidos nas figuras 4.5 e 4.6.

Figura 4.5: Categorias relacionadas com o perfil de utilizador 2 para a querie "Microsoft"

A figura 4.5 reflecte uma pesquisa efectuada utilizando a querie "Microsoft"e recorrendo ao perfil do utilizador. O perfil escolhido aqui foi o perfil gerado para o segundo utilizador referido no subcapítulo anterior, ou seja, é o modelo de uma pessoa que trabalha em Informática. O resultado esperado seria, nos primeiros lugares as categorias que mais se relacionam com o perfil de utilizador criado. A figura 4.6 apresenta os resultados para a mesma querie mas para um utilizador diferente. Neste caso é o primeiro utilizador do subcapítulo anterior. Ou seja, é uma amante de música. Logo, os resultados deverão ser completamente diferentes. Neste caso, a menos que o

Figura 4.6: Categorias relacionadas com o perfil de utilizador 1 para a querie "Microsoft"

utilizador também faça pesquisas sobre informática a ordem das categorias não deverá sofrer qualquer tipo de alteração.

Aqui verificámos que o método utilizado não é o melhor porque só estamos a dar importância às palavras. Isto acaba por viciar um pouco a ordenação. Passo a explicar, o facto do utilizador número dois trabalhar em Informática, mais concretamente com tecnologias Microsoft, dá origem a que no seu perfil esteja a palavra *Microsoft*, muito provavelmente no 1º nível. Isto faz com que as categorias que tenham a palavra *Microsoft* ganhem um peso muito superior.

Ao nível da ordenação por grau de conhecimento não conseguimos apresentar resultados pois para isso iriamos necessitar de ter dois utilizadores com interesses comuns mas com graus de conhecimento distintos, o que não foi o caso. Logo, será incluido em trabalhos futuros quando estiverem reunidas as condições necessárias para tal.

No próximo capítulo entramos mais em detalhe na discussão da possível utilização de métodos alternativos para a ordenação.

Capítulo 5

Conclusão e trabalho futuro

5.1 Conclusão

Em forma de conclusão podemos afirmar que o nosso sistema tem uma performance razoavelmente boa e atinge os objectivos iniciais aos quais nos propusemos. Criámos um sistema independente da língua, onde a ontologia gerada não é baseada numa já existente definida pelo criador. Este sistema faz a reordenação das categorias tendo como base o perfil de utilizador gerado e ainda cria um perfil de conhecimento do utilizador.

Obviamente que ao longo do processo de implementação deste sistema fomos chegando à conclusão que existiam alguns pontos onde poderíamos ter optado por caminhos diferentes, mas é esse mesmo o intuito de uma tese como esta. É definir um percurso e, testando, provar que é ou não possível seguir o raciocínio proposto.

Em relação à Pré-topologia atingimos os objectivos propostos e tivemos mesmo bons resultados. O facto de às vezes as palavras que compõem a ontologia final não serem muito sugestivas tem a ver com dificuldades ao nível da extracção das palavras relevantes do texto e não propriamente com o algoritmo de Pré-topologia.

Ao nível do método utilizado para extrair o grau de complexidade de um texto, penso que este é óptimo para textos onde existe um maior número de palavras. Para o caso em que este é aplicado sobre os snippets os resultados nem sempre são os ideais pois obtemos valores muito baixos. Isto é um problema que tem a ver também com o facto dos snippets devolvidos pelos motores de busca serem, em grande parte dos casos, de complexidade relativamente baixa. Isto é feito propositadamente pelos sistemas de

WebIR utilizados pelos motores de busca. Estes pretendem que o utilizador consiga, pelo snippet, saber qual o seu conteúdo mas se apresentarem textos muito complexos afastam grande parte dos utilizadores.

5.2 Trabalho futuro

Depois de termos chegado ao fim desta tese podemos afirmar que ainda existe um longo caminho a percorrer para obtermos resultados óptimos. Alguns dos melhoramentos a trabalhar no futuro são:

- Utilizar outro método para avaliar a complexidade de um texto. Desta vez, utilizar um método de aprendizagem automático. Estes métodos apesar de requerem um longo processo de treino conseguem, normalmente, atingir melhores resultados do que os métodos clássicos.

- Ainda relativamente ao processo de criação de perfis de conhecimento, em trabalhos futuros, pretendemos, em vez de ter um nível geral do grau de conhecimento do utilizador, ter um grau associado a cada nó da ontologia. Ou seja, para cada área de conhecimento distinta ter também valores diferentes.

- Melhorar a extracção das palavras relevantes dos documentos. Nesta tese utilizámos a implementação referida em [9] já existente no sistema mas desenvolvida para snippets.

- Alterar o algoritmo de reordenação dos resultados. Pois, como foi referido em capítulos anteriores, os pesos atribuídos a cada categoria acabam por ficar viciados devido ao facto de utilizarmos apenas uma comparação simples de palavras. Aqui a ideia será adicionar a noção de contexto a este processo.

- Testar este sistema com pessoas com interesses semelhantes mas com idades diferentes ou pelo menos que consultem documentos sobre o mesmo tema com níveis de complexidade distintos. Desta forma poderemos verificar a qualidade da ordenação por grau de complexidade.

- Criar métodos de avaliação dos resultados por parte de utilizadores de áreas distintas de modo a podermos confirmar a qualidade dos nossos resultados. Nesta tese baseamo-nos na nossa noção de boa qualidade, o que pode não

ser necessariamente objectivo. Esta é uma fase bastante importante mas não conseguimos recursos para tal e a falta de tempo também foi um factor real.

Bibliografia

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *7th International World Wide Web Conference*, pages 107–117, October 1998.

[2] Nielsen announces may u.s. search share rankings with total searches increasing 20 percent-over-year. http://www.nielsen-online.com.

[3] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *WWW05, 14th International World Wide Web Conference*, pages 801–810, 2005.

[4] D. Dreilinger and A.E. Howe. Experiences with selecting search engines using metasearch. In *Journal of ACM Transaction on Information Systems*, pages 195–222, 1997.

[5] Tanner and Chris. Adaptive web personalization: Improving web personalization via user interest hierarchy and scoring techniques. December 2006.

[6] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *21st annual international ACM SIGIR conference on research and development in information retrieval*, pages 46–54, 1998.

[7] M. Radovanovic and M. Ivanovic. Cats: A classification-powered meta-search engine. In *Advances in Web Inteligence and Data Mining*, pages 11:191–200, August 2006.

[8] D. Machado, T. Barbosa, S. Pais, B Martins, and G. Dias. Universal mobile information retrieval. San Diego, USA, July 2009. 13th International Conference on Human Computer Interaction (HCII 2009).

[9] G. Dias, S. Pais, F. Cunha, H. Costa, D. Machado, T. Barbosa, and B. Martins. Hierarchical soft clustering and automatic text summarization for accessing the web on mobile devices for visually impaired people. Sanibel Island, USA, May 2009. 22nd International FLAIRS Conference (FLAIRS 2009).

[10] H.R. Kim and P.K. Chan. Personalized ranking of search results with learned user interest hierarchies from bookmarks. In *WEBKDD Workshop*. SIGKDD Conference 2005, 2005.

[11] L. Tamine, M. Boughanem, and N. Zemirli. Infering the user interests using the search history. In *LWA*, pages 1:108–110, 2006.

[12] A. Sieg, B. Mobasher, and R. Burke. Ontological user profiles for personalized web search. pages 525–534. 16th ACM conference on information and knowledge management, 2008.

[13] J. Teevan, S.T. Dumais, and E. Horvitz. Beyond the commons: Investigating the value of personalizating web search. 2005.

[14] A. Pretschner and S. Gauch. Ontology based personalized search. pages 391–398, Chicago, 1999. 11th IEEE International Conference on Tools with Artificial Inteligence.

[15] S. Gauch, J. Chaffe, and A. Pretschner. Ontology-based personalized search and browsing. 2001.

[16] J. Teevan, S.T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. 2007.

[17] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. 2005.

[18] A. Kobsa. *Privacy-enhanced personalization*, volume 50. Communications of the ACM, 2007.

[19] http://hultig.di.ubi.pt/vipaccess.

[20] ftp://ftp.cs.cornell.edu/pub/smart/.

[21] P.D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, February 2010.

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

[23] A.N. Langville and C.D. Meyer. Deeper inside pagerank. October 2004.

[24] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *17th ACM Annul International Conference on Research and Development in Information Retrieval*, pages 272–281, July 1994.

[25] J. Kim, D.W. Oard, and K. Romanik. Using implicit feedback for user modeling in internet and intranet searching. *Tech. Rep.*, 2000.

[26] M. Claypool, P. Le, M. Waseda, and D. Brown. Implicit interest indicators. In *International Conference on Intelligent User Interfaces*, 2001.

[27] J. Goecks and J. Shavlik. Learning users interests by unobtrusively observing their normal behavior. In *5th International Conference on Intelligent User Interfaces*, pages 129–132, 2000.

[28] P. Chan. A non-invasive learning approach to building web user profiles. In *ACM SIGKDD International Conference*, pages 7–12, 1999.

[29] M. Atkas, M. Nacar, and F. Menczer. Using hyperlink features to personalize web search. pages 104–115, October 2006.

[30] P. Larsson. Classification into readability levels. Master's thesis, Uppsala University, Sweden, 2006.

[31] C. Bjornsson. *Lasbarhet*. GEC GAD, 1971.

[32] G.R. Klare. The measurement of readability. *The Iowa State University Press*, 1963.

[33] http://rfptemplates.technologyevaluation.com/dale-chall-list-of-3000-simple words.html.

[34] K. Collins Thompson and J. Callan. A language modeling approach to predicting readin difficulty. In *HLT/NAACL 2004*, Boston, USA, May 2004.

[35] C. Liu and A. Oh. Automatic recognition of reading levels from user queries. pages 548–549, 2004.

[36] A.C. Graesser, D. McNamara, M. Louwerse, and Z. Cai. Coh-metrix: Analysis of text on cohesion and language. 2004.

[37] Dufty. D.F., D. McNamara, M. Louwerse, A.C. Graesser, and Z. Cai. Automatic evaluation of aspects of document quality. 2004.

[38] A. Sieg, B. Mobasher, and R. Burke. Web search personalization with ontological user profiles. 2007.

[39] G. Cleuziou, G. Dias, and V. Levorato. Modélisation prétopologique pour la structuration sémantico-lexicale. 2009.

[40] D. Machado. Procura estruturada de textos para perfis de utilizadores. Master's thesis, Universidade da Beira Interior, Covilhã, 2009.

[41] C. Largeron and S. Bonnevay. Une method de structuration par recherché de fermés minimaux. application à la modélisation de flux de migration intervilles. 2008.

[42] A. Sieg, B. Mobasher, and R. Burke. Ontological user profiles as the context model in web search. School of Computer Science, Telecommunication and Information Systems, 2006.

[43] http://www.windowsphone.com.

[44] http://www.silverlight.net.