UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

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 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

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. Os primeiros motores de pesquisa apenas indexavam os títulos das páginas Web e descartavam o seu conteúdo. Em 1994, foi dado um passo em frente e o conteúdo das páginas passou a ser indexado também, permitindo assim ao utilizador procurar dentro da própria página e não ficar restringido ao título apenas.

Durante alguns anos, os métodos utilizados pelos motores de pesquisa existentes eram praticamente os mesmos em todos eles até que em 1998 apareceu a Google com 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 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 [1] e que tem vindo a existir 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.

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 da 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 Gulli [4] 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 de todos os documentos da Web passa por recorrer à utilização de um meta-motor de busca [5]. Este explora um conjunto de resultados provenientes de múltiplos motores de pesquisa e como tal aumenta 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 dificuldade da relevância de um documento ser subjectiva dependendo do contexto em que está inserido. A mesma querie pode ter objectivos de pesquisa diferentes. 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 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 fazer com sucesso as funções para os quais foram criados. Existem várias abordagens, mas as três mais importantes são: ordenação, categorização e personalização [6] [7] [8] [9] [4] [10] [11] [12] [13] [14]. Existem algumas abordagens muito interessantes em que são utilizadas as referidas em cima aos pares, tais como, ordenação e categorização [8] [9], ordenação e personalização [6] [7] [10] [12] [13] [14] [15] [16], 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. 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. Daí não ser muito utilizado nos motores de pesquisa mais visitados. 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 são 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 que aqui que nós pretendemos inovar e fazer um sistema

onde utilizamos a categorização, personalização e ordenação como iremos ver no capítulo seguinte.

1.3 Objectivos

O objectivo desta tese é o de melhorar os resultados devolvidos por um sistema de recolha que informação que já utiliza categorização nos seus processos, fazendo uso dos outros dois métodos que referimos no capítulo anterior. 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.

Ambos os métodos estão numa fase um pouco experimental pois existem inúmeros métodos e tentativas mas não existe ainda nenhum que se possa dizer que é o melhor. Mas já podemos ver um exemplo de personalização a funcionar no site da Amazon (http://www.amazon.com).

O funcionamento de um motor de pesquisa é basicamente o seguinte: *Crawling*, indexação de documentos, procura e ordenação dos resultados. Os passos da personalização e da ordenação poderiam ser efectuados em qualquer um dos passos referidos anteriormente mas por motivos de simplificação, e porque estes métodos irão ser integrados num meta-motor de busca, vamos aplicá-los depois de recebermos os resultados dos motores de busca. 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 vamos guardar o histórico de utilização do meta-motor de pesquisa, armazenar as queries efectuadas, as categorias mais vistas, entre outros. 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 no campo dos sistemas de recolha de informação nos motores de busca. 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 é também apresentado o trabalho proposto no âmbito desta tese.

O capítulo 3 apresenta todo o trabalho desenvolvido ao longo da tese, mais propriamente os métodos construídos para a criação de um perfil de utilizador bem como de um perfil do nível de conhecimento do utilizador.

No capítulo 4 será abordado o processo de reordenação dos resultados de acordo com o perfil de utilizador criado.

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.

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 1960, Gerard Salton desenvolveu o SMART. Este era um protótipo de um sistema de recolha de informação que serviu para testar muitos algoritmos que automaticamente indexavam e devolviam documentos. Este processo era muito baseado na teoria do Text Mining. Eram utilizados métodos que ainda são utilizados hoje em dia como por exemplo, análise estatística de texto, extracção da raiz das palavras, 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, 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 [17], 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 por este valor. É contudo, o modelo de espaço vectorial 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.

Um estudo de 2002 [17] 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, Yahoo, Bing, entre outros. 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.

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 imagem seguinte.

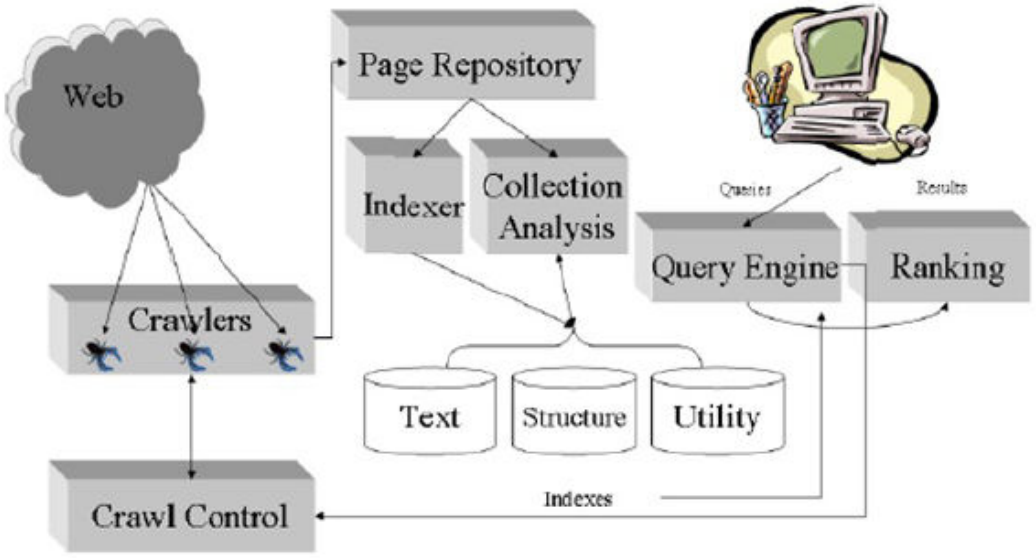

Fig. 1 - Funcionamento de um sistema de recolha de informação

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. Após os documentos terem sido armazenados no repositório vão ser processados e indexados num grafo de forma a permitir um acesso mais rápido à estrutura e conteúdo dos mesmos. Neste grafo 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 [18]. 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 Modelos de utilizador

O facto da informação disponível na Web ser tanta e tão desorganizada impõe requisitos muito altos aos sistemas de recolha de informação que implementam a personalização utilizando modelos de utilizador. 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 milhares de documentos sobre vários temas distintos. O que implica que o utilizador irá necessariamente passar vários minutos à procura da informação que pretende.

De forma a tentar evitar esta lacuna, alguns motores de pesquisa optaram por recorrer à categorização. É assim extraído dos documentos algum conhecimento de forma invisível para o utilizador, de modo a possibilitar que o mesmo possa filtrar, por categoria, os documentos que pretende consultar. Este sistema é inegavelmente uma ajuda mas não é a solução para todos os problemas pois, muitas vezes, os tópicos gerados automaticamente a partir do texto não são suficientes fortes ou 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. A consciência deste 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.

2.2.1 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 convém 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 pretender 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 de modo a que uma segunda visita não demore tanto tempo com estes pormenores e que seja de alguma forma personalizada.

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 do 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 (http://www.amazon.com) 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 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.

2.2.2 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 feita muita pesquisa 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 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 as coisas sejam feitas rapidamente e bem. 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. 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 informação relevante para o utilizador automaticamente 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

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 [19] [20] [21] [22] [23].

Morita e Shinoda [19] 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 [21] 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 [22] 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 a apenas a monitorização da actividade do rato não era suficiente para detectar uma correlação com o interesse do utilizador num documento.

Chan [23], 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 booleando 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

2.2.3 Trabalho relacionado

Aktas, Nacar e Menczer [24] propõem a criação de um sistema de personalização que tem como base o algoritmo PageRank. Este novo sistema passa por introduzir no cálculo do algoritmom 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 [10] 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.

Gulli e Ferragina [4] 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 [11] propõem um sistema baseado em ontologias como representação dos interesses do utilizador. Cada ontologia é inicialmente uma instância de uma ontologia de referência constituída por vários conceitos. Conceitos estes que inicialmente terão valor igual a 1. Á 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, 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 *TFIDF*. 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 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.

A abordagem de Tanner [7] 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 e posteriormente *stemming*. A partir deste momento, cada documento passa a ser representado por frases cujos termos são todos significativos. A estas frases é aplicado um algoritmo denominado "Divisive Hierarchical Clustering" de modo a extrair possíveis pares de palavras. O *output* deste algoritmo 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 calculado através do somatório do peso atribuído a cada termo existente na árvore e no documento.

2.3 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 motor de busca.

Mas estes sistemas têm também muita importância noutras àreas, tais como a educação [2]. 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: "A soma de propriedades linguísticas num texto, que tornam o texto mais ou menos compreensível para o leitor".

De acordo com Klare [3] 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 será também mostrado algum trabalho existente nesta área.

2.3.1 Construção de um modelo de 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. 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 ser dependente 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.

2.3.2 Trabalho relacionado

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. Passo a explicar cada uma em mais detalhe:

1. **Lorges** (1939) revista em 1948, classifica os textos em anos escolares (3-12)

   $$Ano = 0.07sl + 0.1073w_d + 0.1301pp + 1.6126 \quad (1)$$

   - *sl = comprimento médio de uma frase*
   - w_d *= número de palavras difíceis diferentes por cada 100 palavras. As palavras consideradas difíceis são todas as palavras que não estão da lista de 769 palavras de Dale*
   - *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.

2. **Dale-Challs** (1948) revista em 1995, classifica os textos em anos escolares (3-12)

   $$Ano\ Escolar = 0.596sl + 0.1579w_d + 3.6365 \quad (2)$$

   - *sl = comprimento médio de uma frase*
   - w_d *= número de palavras que não ocorrem na lista de 3000 palavras de Dale*

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 de 3000 palavras.

3. **Flesch Reading Ease** (1948) revista várias vezes ao longo do anos e devolve uma pontuação onde o valor mais alto é o texto mais difícil de ler

$$Complexidade = 206.835 - 1.015sl - 0.846wl \quad (3)$$

   - *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.

4. **Flesch-Kincaid Grade Level** (1975), classifica os textos em anos escolares Americanos

$$Ano\ Escolar = 0.39sl + 11.8wl - 15.59 \quad (4)$$

   - *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.

5. **LIX** (1968) desenvolvida para a língua Sueca

$$LIX = \frac{wl}{s} + 100 * \frac{w_d}{wl} \quad (5)$$

   - *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 LYX tem que ser verificada a partir de uma tabela designada por LYX-interpreter apresentada em baixo. A vantagem da

fórmula LYX é o facto de poder ser aplicada facilmente a outra língua bastando para isso modificar a escala.

Tabela 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:

1. *Language Modeling Aproach to Predicting Reading Dificulty* 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.

2. *Automatic Recognition of Reading Levels from User Queries*, tal como o método apresentado em cima, tenta classificar texto 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 verdadeiras, 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's* (Support Vector Machines) 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.

3. Coh-Metrix: Analysis of text on cohesion and language [26], tenta criar um método baseado em dois conceitos principais: coesão e coerência do texto. Estudos recentes na área da psicologia e

linguística [25] [26] 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 de texto.

2.4 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 de que ainda não foi encontrado um método que tenha resultados minimamente aceitáveis.

Agichtein, Brill e Dumais [16] utilizam métodos implícitos 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 que utilizam para fazer a ordenação.

Sieg, Mobasher e Burke [30] recorrem a um método de criação de perfis baseado num perfil de base e posteriormente fazem a ordenação recorrendo a um algoritmo por eles definido. Este algoritmo baseia-se na similaridade de conceitos, ou seja, este método não se baseia apenas nas palavras. Tem também uma noção do contexto associado à palavra. A base de comparação é a ontologia criada no primeiro passo.

Pretschner e Gauch [13] fazem a reordenação dos resultados devolvidos por um motor de pesquisa público, neste caso o ProFusion (www.profusion.com). Os autores vão a 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.

2.5 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. Pois bem, visto que todo o trabalho desta tese será integrado num sistema que já vem a ser trabalhado há algum tempo e que já contém a componente da categorização, vamos focar-nos 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. 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.

Após uma análise bastante atenta ao estado da arte chegámos à conclusão que existem métodos para os quais existe uma grande probabilidade de conseguirmos ter bons resultados se os cruzarmos. Decidimos em primeiro lugar utilizar um método de personalização baseado em ontologias geradas automaticamente através das palavras mais relevantes extraídas dos documentos visitados pelo utilizador. O processo feito aqui será explicado mais em detalhe no capítulo seguinte mas a essência deste processo é o seguinte, passo a explicar.

Ao longo do tempo vamos recolhendo informação sobre as páginas que o utilizador visita, as queries que faz e as categorias que selecciona para chegar mais prontamente aos resultados que pretende. Depois, e este será um processo feito *"offline"* periodicamente, são extraídas as palavras mais relevantes de cada documento visitado, sendo que os documentos visitados com mais frequência terão um peso maior na avaliação final. Posteriormente, estas palavras são processadas por um algoritmo assimétrico de co-ocorrência onde vou estudar a correlação das palavras. E por fim,

aplico o algoritmo de pré-topologia para criar a ontologia relativa ao utilizador.

O perfil de conhecimento do utilizador é gerado a partir da média do valor de complexidade de cada documento que o utilizador leu.

Agora que temos os perfis de utilizador e de conhecimento criados podemos proceder à ordenação. Aqui vamos proceder a uma reordenação por categoria que pode ser grau de generalidade ou especificidade. Neste ponto é calculada similaridade entre as várias categorias e os diferentes níveis da ontologia, sendo que quanto mais alto for o nível da ontologia mais peso irá ter a categoria. E no final, no caso de haver categorias com pesos semelhantes vamos ordenar por grau de conhecimento. Ou seja, a ordenação é feita primeiro de acordo com o perfil de utilizador e só depois por grau de conhecimento.

3. Criação de modelos de utilizador

3.1 Criação do modelo de utilizador

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 a 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.

Pois a nossa escolha passou pela criação de um sistema implícito, no qual o utilizador não se irá aperceber de que os dados estão a ser recolhidos. Muitos estudos [19] [20] [21] [22] [23] 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 pelos seguintes índices:

- Abertura de um ou mais documentos
- Em pelo menos um dos documentos abertos o utilizador passou mais do que *s* segundos na sua leitura

Se a querie introduzida pelo utilizador satisfaz ambos os requisitos, então é adicionada ao histórico da sessão de pesquisa do utilizador.

Fig. 2 - Exemplo de uma pesquisa no VIPAccess Mobile.

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 de sessão serão enviados 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 do utilizador, bem como as categorias associadas a estas queries e que foram obtidas a partir do algoritmo de categorização já existente. 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.

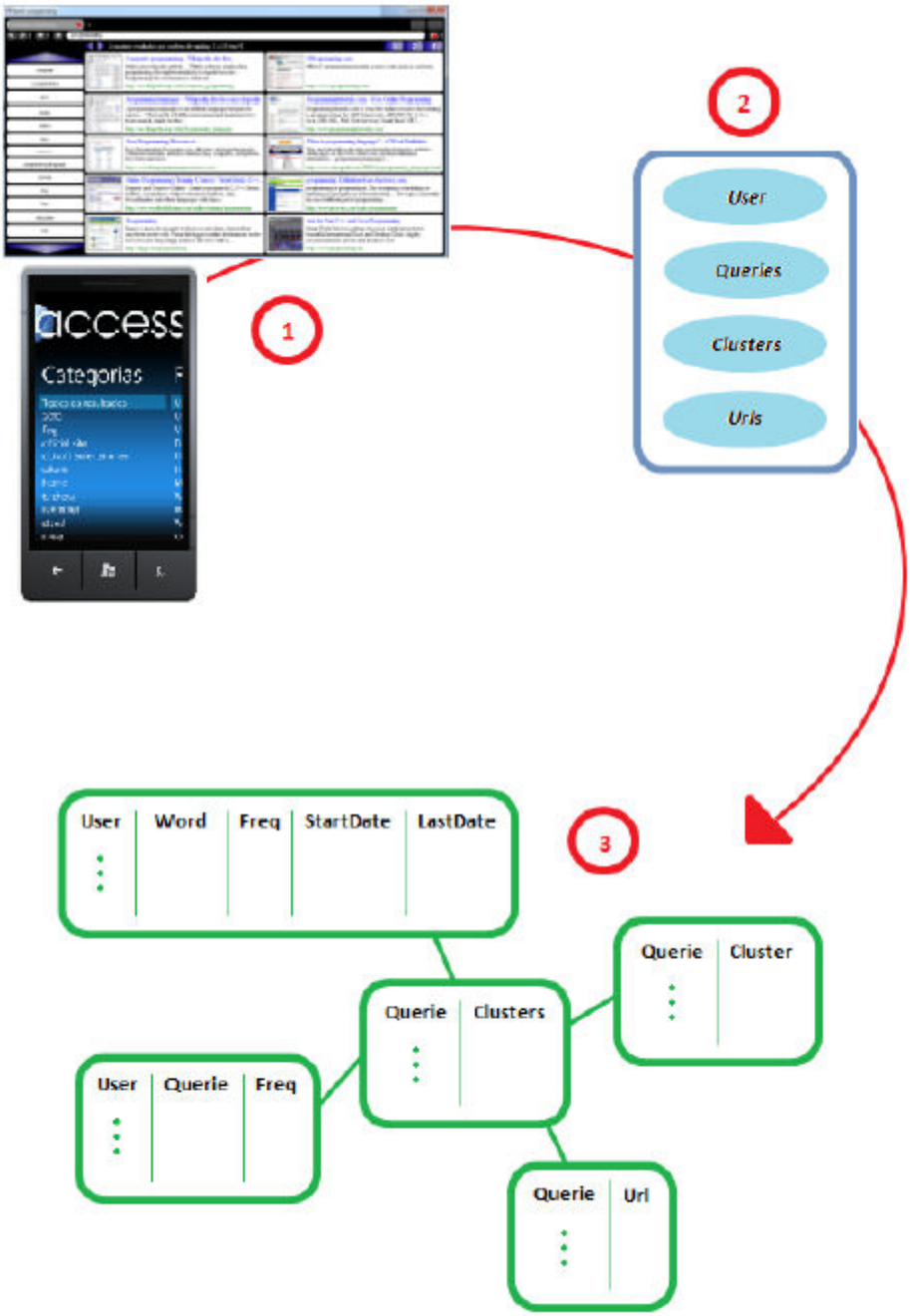

Fig. 3 - Diagrama do processo de recolha de informação sobre o utilizador.

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, não quando um utilizador faz uma pesquisa mas sim quando o utilizador não está a realizar nenhuma pesquisa. É um processo que corre com uma determinada frequência e cujo intervalo de tempo pode ser facilmente modificado pois é um

parâmetro. 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 de computação e ao nível de tempo, o que não é minimamente preocupante pelo facto de ser não ser executado em tempo real.

O processo de criação do perfil de utilizador passa por 4 fases essenciais. A primeira é a recolha de informação do utilizador e já foi explicada em cima.

A segunda é a extracção das palavras mais relevantes nos documentos dentro do contexto do utilizador. Aqui utilizamos um método já existente no sistema [29] e que foi utilizado anteriormente para definir extrair 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 ainda resultados 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-ocorrências destas palavras. Aqui vamos juntar as palavras extraídas dos documentos com as palavras utilizadas nas queries e com as categorias em que utilizador clicou para chegar mais depressa aos resultados. De seguida vamos aplicar um algoritmo conhecido como SCP com o objectivo de obter uma matriz de co-ocorrências das palavras. Esta matriz será utilizada como base para o passo seguinte. O SCP (Symmetric Conditional Probability) é o produto de duas probabilidades condicionadas. É o produto da probabilidade de a palavra *w1* aparecer no texto juntamente com a palavra *w2*, P(w1|w2), e da probabilidade da palavra *w2* aparecer juntamente

com a palavra *w1*, P(w2|w1). Os resultados devolvidos são valores entre 0 e 1 como seria de esperar pois estamos a falar de uma probabilidade.

$$P = O_{11}^2/(R_1 \times C_1) \quad (6)$$

A nossa matriz irá ser uma matriz global, ou seja, referente a todos os documentos. No caso em que uma relação já está presente na matriz é feita uma média entre o peso antigo e o peso novo.

E por fim, é aplicado o algoritmo de Pré-topologia [27] [28] [29] de modo a criar uma ontologia que será no final o modelo do utilizador. A pré-topologia é um algoritmo 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 das palavras. A pré-topologia é baseada em 6 definições:

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

- **Definição 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 em baixo) e é definida como : $\forall A, A \subseteq E$ é o pseudo-fecho de $A, a(A) \subseteq E$ de tal forma que:
  - $a(\emptyset) = \emptyset$
  - $A \subseteq a(A)$

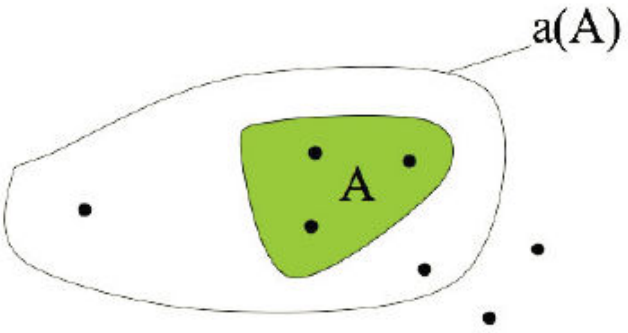

Fig. 4 - Pseudo-fecho de A

- **Definição 2** – O espaço pré-topológico $V\,(E,a)$é definido por:
  $\forall A,B,A \subseteq \mathrm{E},\ \mathrm{B} \subseteq \mathrm{E}\ \mathrm{e}\ \mathrm{A} \subseteq \mathrm{B}\ \mathrm{então}\ \mathrm{a(A)} \subseteq \mathrm{a(B)}$ (7)

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

- **Definição 3** – O espaço pré-topológico $V_d\,(E,a)$é definido por:
  $\forall A,B,A \subseteq \mathrm{E},\ \mathrm{B} \subseteq \mathrm{E},\ \mathrm{a(A \cup B)} = \mathrm{a(A)} \cup \mathrm{a(B)}$ (8)

- **Definição 4** – O espaço pré-topológico $V_s\,(E,a)$ é definido por:
  $\forall A,A \subseteq \mathrm{E},\ \mathrm{a(A)} = \bigcup_{x \in A} a(\{x\})$ (9)

- **Definição 5** – $A \in P(E)$ é fechado se e apenas se:
  $A = a(A)$ (10)

- **Definição 6** – Seja Xum 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 uma 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.

O algoritmo utilizado para a criação de ontologias é o apresentado na figura 5. E na figura 6 é apresentado um exemplo prático que serve para uma melhor e mais fácil compreensão do algoritmo.

```
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
```

Fig. 5 - Algoritmo utilizado na criação da ontologia.

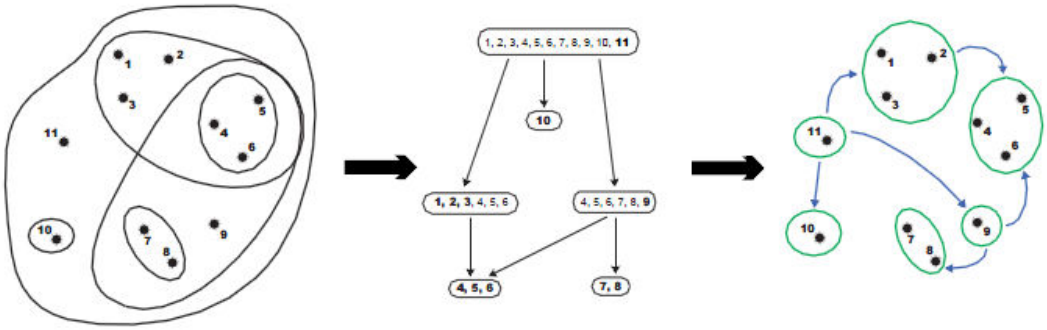

Fig. 6 - Exemplo de utilização do algoritmo de pré-topologia.

No algoritmo exibido na figura 5 podemos ver uma função chamada *MaxClosedSubsets*. A forma de calcular o fecho dentro desta função está dependente de dois parâmetros, α e ß que medem a *distância* entre os vários elementos do conjunto. O valor ideal [29] para α é um valor que ronda os 50% numa escala normalizada de possíveis valores de α. Para ß, o valor ideal encontra-se entre os 60% e 80% na escala normalizada de possíveis valores de ß.

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.

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 LYX [2].

$$LIX = \frac{wl}{s} + 100 * \frac{w_d}{wl} \qquad (11)$$

Este método, como foi referido e explicado detalhadamente num capítulo anterior e apesar de ter sido desenhado para a língua Sueca pode ser aplicado com sucesso noutras línguas como afirma Larsson [2]. 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.

4. 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 [11] [30] [31] 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. A primeira coisa a fazer é pegar no conjunto das categorias devolvidas (conjunto A), em seguida pegamos na ontologia criada anteriormente (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:

- Pegamos nas 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 $X \times N$, onde N é o nível da ontologia onde a palavra se encontra

- Caso o resultado da comparação seja negativa o mesmo peso é decrementado com um valor $\mathrm{Y} \times \mathrm{N}$, onde N é o nível da ontologia onde a palavra se encontra

O facto de existirem dois valores diferentes, X e Y, 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 $X > Y$.

Como foi referido o método de reordenação utilizado neste sistema não é de grande complexidade mas desempenha o seu trabalho bastante bem. 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 mesmo uma alteração na ordem das categorias que corresponde ao perfil do utilizador criado.

5. Discussão dos resultados

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 [32].

Fig. 7 - Exemplo de uma pesquisa no VIPAccess Mobile.

As aplicações para Windows Phone 7 (WP7) são todas feitas em Silverlight [33] 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, 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.

No âmbito desta tese, e como já foi referido anteriormente, fizemos uma versão do nosso meta-motor de busca, que já se encontra a correr na Web, 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 em baixo.

Umas 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.

Fig. 8 - Controlo de panorama do Windows Phone 7.

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

Como podemos verificar na imagem exibida no início deste capítulo, 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.

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.

5.1 Modelo de utilizador

Como referimos no capítulo sobre a pré-topologia o resultado deste processo é uma ontologia, que é um conjunto de palavras relacionadas e com noções de hierarquia, ou seja, existe uma disposição das mesmas consoante a sua subjectividade ou objectividade. 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 3 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, o segundo é um fã de música e adoro ir a concertos, sendo que o terceiro elemento é uma pessoa que trabalha em informática, mais concretamente com tecnologias Microsoft. Os perfis gerados para cada um destes utilizadores são apresentados em baixo.

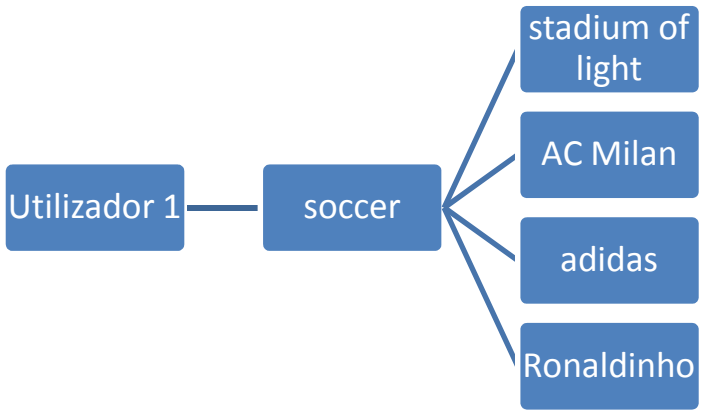

Fig. 9 - Ontologia criada para o utilizador 1.

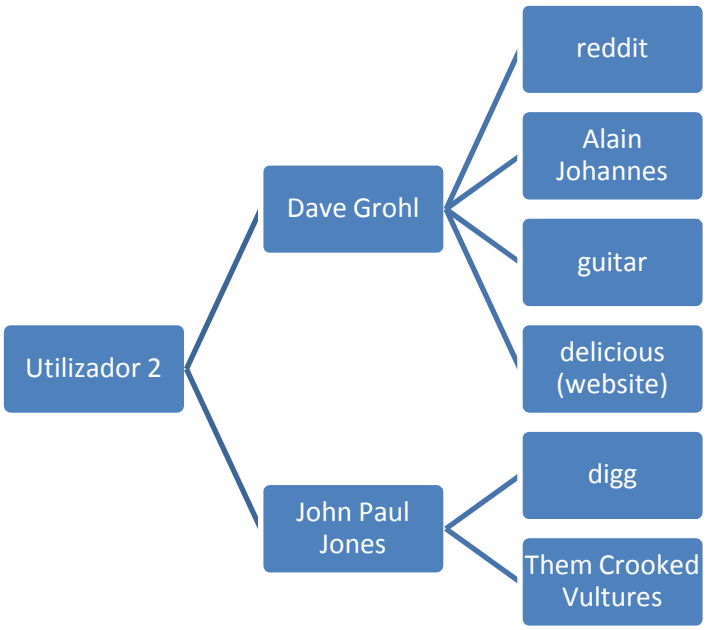

Fig. 10 - Ontologia criada para o utilizador 2.

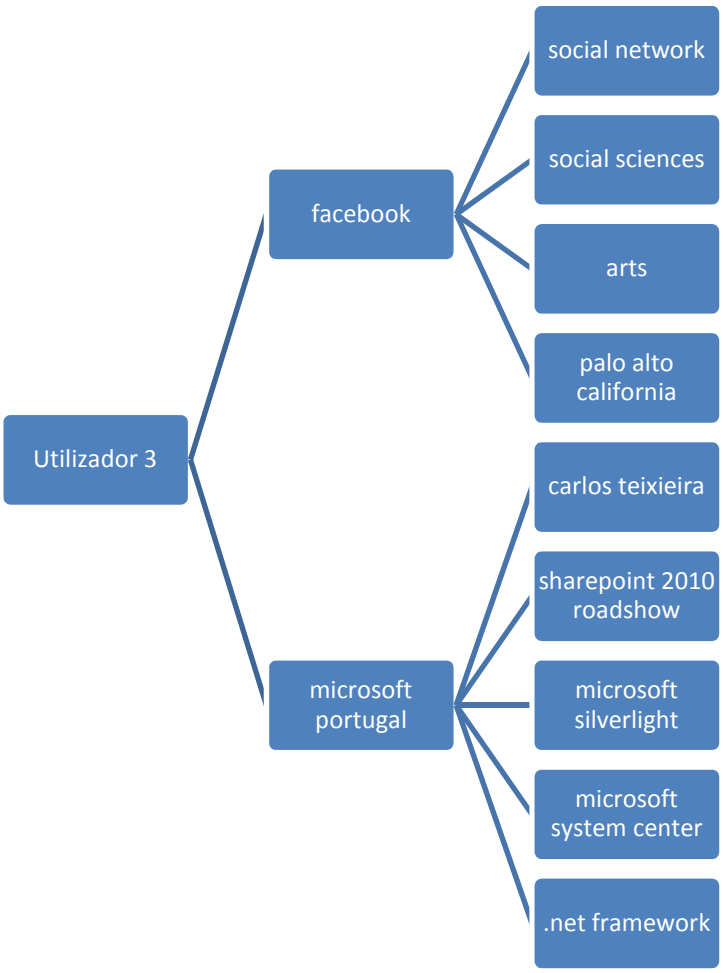

Fig. 11 - Ontologia criada para o utilizador 3.

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.

5.2 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 não contiver 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.

5.3 Reordenação dos resultados

Ao nível do processo de reordenação dos resultados, como é possível verificar pelas explicações dos capítulos anteriores, 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 imagens em baixo.

Fig. 12 - Categorias relacionadas com o perfil do utilizador 3 para a querie "Microsoft".

Fig. 13 - Categorias relacionadas com o perfil do utilizador 2 para a querie "Microsoft".

A primeira imagem reflecte uma pesquisa efectuada utilizando a querie "Microsoft" e recorrendo ao perfil de utilizador. O perfil escolhido aqui foi o perfil gerado para o 3º 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 segunda imagem apresenta os resultados para a mesma querie mas para um utilizador diferente. Neste caso é o utilizador número dois do subcapítulo anterior. Ou seja, é uma amante de música. Logo, os resultados deveram ser completamente diferentes. Neste caso, a menos que o 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 3 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. No próximo capítulo entramos mais em detalhe na discussão da possível utilização de métodos alternativos para a ordenação.

6. Conclusão e trabalho futuro

6.1 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.

6.2 Trabalho futuro

Depois de termos chegado ao fim desta tese podemos afirmar que ainda existe um longo caminho a percorrer para obtermos resultados perfeitos, mas foi feito um bom trabalho. Algumas das coisas a fazer no futuro são:

- Utilizar outro método para avaliar a complexidade de um texto. Desta vez, utilizar um método automático. Estes métodos apesar de requerem um longo processo de treino conseguem, normalmente, atingir melhores resultados do que um método clássico.
- 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 do David [26] já existente no sistema.
- Alterar o algoritmo de reordenação dos resultados pois, como já referidos 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.
- 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 verdade. Esta é uma fase bastante importante mas não conseguimos recursos para tal e a falta de tempo também foi um factor real.

Bibliografia

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

[2] Patrik Larsson, "Classification into Readability Levels", Master's thesis in Computational Linguistics, Uppsala University, Uppsala, Sweden, 2006

[3] George R. Klare, "The Measurement of Readability", The Iowa State University Press, 1963.

[4] 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, pages 801-810, 2005.

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

[6] 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.

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

[8] Zamir O. And 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, pages 46-54, 1998.

[9] Radovanovic M. And Ivanovic M., "Cats: A classification-powered meta-search engine", Advances in Web Intelligence and Data Mining, (11):191-200, August 2006.

[10] Tamine L., Boughanem M., and Zemirli N., "Inferring the user interests using the search history", LWA, 1:108-110, 2006.

[11] Sieg A., Mobasher B., and Burke R., "Ontological user profiles for personalized web search", Conference on Information and Knowledge Management Proceedings of the 16th ACM conference on Conference on Information and Knowledge Management, pages 525-534, 2008.

[12] Teevan J., Dumais S.T., Horvitz E., "Beyond the commons: Investigating the value of personalizing web search", 2005.

[13] Pretschner A., Gauch S., "Ontology based personalized search", in Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence, 391-398, Chicago, November 1999

[14] Gauch S., Chafee J., Pretschner A., "Ontology-based Personalized search and Browsing", 2001

[15] Teevan J., Dumais S.T., Horvitz E., "Personalizing Search via Automated Analysis of Interests and Activities", 2007.

[16] Agichtein E., Brill E., Dumais S., "Improving Web Search Ranking by Incorporating User Behavior Information", 2005.

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

[18] Brin s. And Page L., "The anatomy of a large-scale hypertextual web search engine", *in Proceedings of 7th International World Wide Wide Web Conference*, pages 107-117, 1998.

[19] Morita M. And Shinoda Y., "Information filtering based on user behaviour analysis and best match text retrieval", *in Proceedings of 17th ACM Annual*

International Conference on Research and Development in Information Retrieval, pages 272-281, July 1994.

[20] Kim J., Oard D.W. and Romanik K., "Using implicit feedback for user modelling in internet and intranet searching", *Tech. Rep., College of Library and Information Services, University of Maryland, College Park, 2000.*

[21] Claypool M., Le P., Waseda M. And D. Brown, "Implicit interest indicators", *In Proceedings of International Conference on Intelligent User Interfaces, 2001.*

[22] Goecks J. And Shavlik J., "Learning users interests by unobtrusively observing their normal behaviour", *In Proceedings of 5th International Conference on Intelligent User Interfaces,* pages 129-132, 2000.

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

[24] Aktas M., Nacar M. And Menczer F., "Using hyperlink features to personalize web search", *Advances in Web Mining and Web Usage Analysis,* pages 104-115, October 2006.

[25] Dufty D.F., McNamara D., Louwerse M., Cai Z. And Graesser A.C., "Automatic Evaluation of Aspects of Document Quality", Psychology Department, University of Memphis,2004

Graesser A.C., McNamara D., Louwerse M. and Cai Z.,"Coh-Metrix: Analysis of text on cohesion and language" , Psychology Department, University of Memphis,2004

[26] Machado D., "Procura Estruturada de Textos para Perfis de Utilizadores", Tese de Mestrado, Universidade da Beira Interior, Covilhã, 2009.

[27] Cleuziou G., Dias G., Levorato V., "Modélisation Prétopologique pour la Structuration Sémantico-Lexicale",2009.

[28] Largeron C., Bonnevay S. “Une method de structuration par recherché de fermés minimaux, Application à la modélisation de flux de migration inter-villes”, 2008.

[29] Levorato V., Bui. M., “Data Structures and Algorithms for Pretopology: The JAVA based software library PretopoLib”, 2010.

[30] Sieg A., Mobasher B. And Burke R., “Web Search Personalization with Ontological User Profiles”, School of Computer Science, Telecommunication and Information Systems, DePaul University, Illinois, USA, 2007.

[31] Sieg A., Mobasher B. And Burke R., “Ontological User Profiles as the Context Model in Web Search”, School of Computer Science, Telecommunication and Information Systems, DePaul University, Illinois, USA, 2006.

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

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