

Sistema de Recomendação para Plataformas de e-learning

Bruno Mário Tavares

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Arquiteturas, Sistemas e Redes**

Orientador: Prof. Doutor Luiz Faria

Co-orientador: Prof. Dr. Constantino Martins

Júri:

Presidente:

Vogais:

Dedicatória

À minha mulher, Sílvia, pelo apoio, carinho e compreensão que me proporcionaram condições para a realização do mestrado.

Resumo

O aumento do número de recursos digitais disponíveis dificulta a tarefa de pesquisa dos recursos mais relevantes, no sentido de se obter o que é mais relevante. Assim sendo, um novo tipo de ferramentas, capaz de recomendar os recursos mais apropriados às necessidades do utilizador, torna-se cada vez mais necessário. O objetivo deste trabalho de I&D é o de implementar um módulo de recomendação inteligente para plataformas de *e-learning*. As recomendações baseiam-se, por um lado, no perfil do utilizador durante o processo de formação e, por outro lado, nos pedidos efetuados pelo utilizador, através de pesquisas [Tavares, Faria e Martins, 2012].

O e-learning 3.0 é um projeto QREN desenvolvido por um conjunto de organizações e tem com objetivo principal implementar uma plataforma de e-learning. Este trabalho encontra-se inserido no projeto e-learning 3.0 e consiste no desenvolvimento de um módulo de recomendação inteligente (MRI). O MRI utiliza diferentes técnicas de recomendação já aplicadas noutros sistemas de recomendação. Estas técnicas são utilizadas para criar um sistema de recomendação híbrido direcionado para a plataforma de *e-learning*. Para representar a informação relevante, sobre cada utilizador, foi construído um modelo de utilizador. Toda a informação necessária para efetuar a recomendação será representada no modelo do utilizador, sendo este modelo atualizado sempre que necessário. Os dados existentes no modelo de utilizador serão utilizados para personalizar as recomendações produzidas.

As recomendações estão divididas em dois tipos, a formal e a não formal. Na recomendação formal o objetivo é fazer sugestões relacionadas a um curso específico. Na recomendação não-formal, o objetivo é fazer sugestões mais abrangentes onde as recomendações não estão associadas a nenhum curso. O sistema proposto é capaz de sugerir recursos de aprendizagem, com base no perfil do utilizador, através da combinação de técnicas de similaridade de palavras, um algoritmo de *clustering* e técnicas de filtragem [Tavares, Faria e Martins, 2012].

Palavras-chave: modelo de utilizador, sistemas adaptativos, sistemas de recomendação, sistemas de *e-learning*, *query clustering*.

Abstract

As more and more digital resources are available, finding the appropriate document becomes harder. Thus, a new kind of tools, able to recommend the more appropriated resources according the user needs, becomes even more necessary. The objective of this I&D work is to implement an intelligent recommendation module (MRI) for e-learning platforms. The recommendations are based on one hand, the performance of the user profile and on the other hand, the requests made by the user in the form of search queries [Tavares, Faria e Martins, 2012].

The e-learning 3.0 is a project developed by a group of organizations and has as primary objective the development of an e-learning platform. This work is inserted in the project e-learning 3.0 being responsible for the MRI. The MRI uses different techniques, which are already being used in recommendation systems, and apply those techniques to create a hybrid tutoring system for an e-learning platform. A user model was built to represent the relevant information about each user. All the information needed to do a recommendation is represented in that model, the model will be updated every time it is necessary. The data in the user model will be used to personalize the produced recommendations.

The recommendations are divided into two types, the formal recommendation and the non-formal recommendation. In the formal recommendation the goal is to make suggestions related to a specific course. In the non-formal recommendation the purpose is to make suggestions that are not associated with any course at all. The solution is capable of suggesting learning resources, based in a user profile, by combining string similarity techniques, clustering algorithms and filtering techniques [Tavares, Faria e Martins, 2012].

Keywords: user modeling, user-adapted systems, recommending systems, e-learning systems, query clustering.

Agradecimentos

Na realização desta dissertação existiram vários intervenientes que diretamente ou indiretamente colaboram ou prestaram apoio, todos eles merecem o meu reconhecimento e gratidão.

Gostaria de agradecer principalmente aos meus orientadores Prof. Doutor Luiz Faria e o Prof. Dr. Constantino Martins, pela paciência, empenho e disponibilidade demonstrada ao longo deste projeto, assim como os comentários e sugestões.

Gostaria também de agradecer ao GECAD, Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão, pela oportunidade proporcionada e pelos meios disponibilizados para a realização deste projeto.

De modo especial gostaria de agradecer à minha família a compreensão, apoio, motivação e incentivo que sempre deram.

Índice

| | | |
|----------|--|-----------|
| 1 | Introdução | 19 |
| 1.1 | Apresentação do GECAD e do consórcio | 19 |
| 1.2 | Objetivos | 20 |
| 1.3 | Relevância do tema | 21 |
| 1.4 | Motivação | 22 |
| 1.5 | Contributos | 22 |
| 1.6 | Estrutura da dissertação | 22 |
| 2 | Caracterização do problema | 25 |
| 2.1 | Metodologias e técnicas | 25 |
| 2.1.1 | Filtros | 26 |
| 2.1.2 | Algoritmos agregadores (<i>Clustering</i>) | 27 |
| 2.1.3 | Exemplos de sistemas de recomendação | 30 |
| 2.2 | Especificação do problema | 32 |
| 2.3 | Resumo | 33 |
| 3 | Desenvolvimento de modelos | 35 |
| 3.1 | Modelo de utilizador | 37 |
| 3.2 | Contextos de aprendizagem | 39 |
| 3.3 | Stemming | 40 |
| 3.4 | SCORM | 41 |
| 3.5 | Clustering | 41 |
| 3.6 | Similaridade | 42 |
| 3.7 | Filtros implementados | 44 |
| 3.7.1 | Filtro baseado em conteúdo | 44 |
| 3.7.2 | Filtro baseado em colaboração | 45 |
| 3.8 | Sistema de recomendação híbrido | 46 |
| 3.9 | Resumo | 47 |
| 4 | Implementação | 49 |
| 4.1 | Descrição técnica | 49 |
| 4.2 | Arquitetura do sistema | 50 |
| 4.3 | Modelo da base de dados | 52 |
| 4.4 | Ferramentas utilizadas | 53 |
| 4.5 | Serviços externos | 55 |
| 4.5.1 | URIs dos serviços web | 56 |

| | | |
|----------|------------------------------------|-----------|
| 4.5.2 | Parâmetros dos serviços web..... | 58 |
| 4.6 | Serviços internos | 61 |
| 4.7 | Exemplo de utilização | 61 |
| 4.8 | Resumo | 63 |
| 5 | Conclusões | 65 |
| 5.1 | Conclusões e análise crítica | 65 |
| 5.2 | Trabalho futuro | 66 |
| 6 | Anexos..... | 73 |
| 6.1 | Estruturas de dados | 73 |
| 6.2 | Exemplos..... | 75 |

Lista de Figuras

| | |
|---|----|
| Figura 1 - Exemplo da aplicação do algoritmo DBSCAN [DBSCAN, 2012] | 29 |
| Figura 2 - Módulo de Recomendação Inteligente | 35 |
| Figura 3 - Pedido não formal de recomendação | 36 |
| Figura 4 - Pedido formal de recomendação | 37 |
| Figura 5 - Algoritmo do MRI simplificado | 39 |
| Figura 6 - Pseudo código do DBSCAN [DBSCAN, 2012] | 42 |
| Figura 7 - Processo de produção de recomendações do MRI | 46 |
| Figura 8 - Arquitetura do MRI, serviços externos | 50 |
| Figura 9 - Modelo da base de dados do MRI | 52 |
| Figura 10 - Resultados do cálculo da distância entre a <i>query</i> e os diferentes <i>clusters</i> | 62 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 - Vantagens e desvantagens do DBSCAN | 28 |
| Tabela 2 - Vantagens e desvantagens de SOM | 29 |
| Tabela 3 - Sistemas de recomendação e <i>data-mining</i> | 30 |
| Tabela 4 - Vantagens e desvantagens dos sistemas apresentados | 31 |
| Tabela 5 - Modelo de utilizador | 38 |
| Tabela 6 - Principais tecnologias utilizadas no desenvolvimento do MRI | 53 |
| Tabela 7 - Serviços web..... | 55 |
| Tabela 8 - URIs dos serviços web dos utilizadores | 56 |
| Tabela 9 - URIs dos serviços web dos recursos..... | 57 |
| Tabela 10 - URIs dos serviços web de recomendações..... | 58 |
| Tabela 11 - Parâmetros dos serviços web dos utilizadores | 59 |
| Tabela 12 - Parâmetros dos serviços dos recursos | 60 |
| Tabela 13 - Parâmetros dos serviços web de recomendação..... | 60 |
| Tabela 14 - Exemplo de <i>clusters</i> criados | 61 |
| Tabela 15 - Tabela de resultados do exemplo MRI..... | 63 |

Acrónimos e Símbolos

Lista de Acrónimos

| | |
|---------------|---|
| API | <i>Application Programming Interface</i> |
| CBF | <i>Content-Based Filtering</i> |
| CF | <i>Collaborative-based Filtering</i> |
| CMD | <i>Command</i> |
| DBSCAN | <i>Density-based Spatial Clustering of Applications with Noise</i> |
| DDD | <i>Domain Dependent Data</i> |
| DID | <i>Domain Independent Data</i> |
| FCT | Fundação para a Ciência e Tecnologia |
| GECAD | Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão |
| GNU | <i>General Public License</i> |
| GUI | <i>Graphical User Interface</i> |
| HTTP | <i>Hypertext Transfer Protocol</i> |
| IEEE | <i>Institute of Electrical and Electronics Engineers</i> |
| IPP | Instituto Politécnico do Porto |
| ISEP | Instituto Superior de Engenharia do Porto |
| ITS | <i>Intelligent Tutoring System</i> |
| JSON | <i>JavaScript Object Notation</i> |
| KBF | <i>Knowledge Based Filtering</i> |
| LCMS | <i>Learning Content Management System</i> |
| LOM | <i>Learning Object Metadata</i> |
| MVC | <i>Model-View-Controller</i> |
| MRI | Módulo de Recomendação Inteligente |
| QREN | Quadro de Referência Estratégico Nacional |

| | |
|--------------|---|
| REST | <i>Representational State Transfer</i> |
| SCORM | <i>Shareable Content Object Reference Model</i> |
| SOM | <i>Self-Organizing Map</i> |
| UBI | Universidade da Beira Interior |
| URI | <i>Uniform Resource Identifier</i> |
| xHTML | <i>eXtensible Hypertext Markup Language</i> |
| XML | <i>eXtensible Markup Language</i> |

1 Introdução

Neste capítulo é apresentada a unidade de investigação e desenvolvimento onde este projeto foi desenvolvido, bem como as organizações envolvidas no desenvolvimento do projeto e-learning 3.0. Os objetivos gerais do projeto e-learning 3.0 e a forma como este trabalho está inserido nesse projeto, está descrito no subcapítulo da apresentação. Os objetivos deste trabalho encontram-se no segundo ponto onde são apresentados os objetivos gerais e se encontram detalhados os objetivos específicos. Posteriormente é apresentada a motivação existente em torno desta área e os contributos deste trabalho. No final deste capítulo é descrita a estrutura da dissertação.

1.1 Apresentação do GECAD e do consórcio

O grupo de investigação em Engenharia do Conhecimento e Apoio à Decisão (GECAD) é uma unidade de investigação e desenvolvimento, reconhecida pela Fundação para a Ciência e Tecnologia (FCT), com sede no Instituto Superior de Engenharia do Porto (ISEP) do Instituto Politécnico do Porto (IPP), tendo como principal objetivo a promoção e desenvolvimento de investigação científica no domínio da Inteligência Artificial, Sistemas Baseados em Conhecimento e Sistemas de Apoio à Decisão O e-learning 3.0 é um projeto financiado pelo Quadro de Referência Estratégico Nacional (QREN) que está a ser desenvolvido por um consórcio de onde são copromotores as seguintes entidades: GECAD, a Universidade da Beira Interior (UBI), a empresa I-Zone *Knowledge Systems* (I.Zone KS) e a empresa MAISIS – Sistemas de Informação [Tavares, Faria e Martins, 2012] .

O projeto e-learning 3.0 pretende tirar partido de tecnologias existentes em plataformas LCMS (*Learning Content Management System*) e de apoio à formação/educação. Desta forma, ser capaz de estimular o acesso a conteúdos que permitam que a aprendizagem se realize quer num contexto de formação profissional, quer em contextos de educação. Pretende-se ainda tirar partido de dispositivos móveis e redes sociais de forma a ser criado um ambiente de fácil acesso para os utilizadores onde as competências profissionais sejam interligadas com

a necessidade de aprendizagem ao longo da vida. Um dos requisitos definidos, para atingir este objetivo, foi o de criar um sistema inteligente capaz de sugerir conteúdos, ou outros utilizadores, adaptadas às necessidades de aprendizagem do utilizador. O trabalho desenvolvido no GECAD e descrito nesta dissertação apresenta uma solução para este objetivo específico.

1.2 Objetivos

Este trabalho apresenta um sistema desenvolvido como um módulo de recomendação inteligente (MRI) para plataformas de *e-learning*. Este trabalho surgiu para responder a um objetivo do projeto e-learning 3.0. Também foi definido no projeto principal que os módulos a serem desenvolvidos deveriam ser independentes da plataforma principal. Cada parceiro desenvolveu o módulo de que é responsável, de forma que no final os módulos sejam integrados num protótipo. No âmbito deste trabalho a expressão 'plataforma principal' será generalizada passando a ser utilizado o termo LCMS. O MRI permite adicionar informação sobre qualquer tipo de recurso de aprendizagem, não sendo necessário este estar associado a qualquer curso. A implementação do MRI tem como objetivo adaptar as ofertas formativas às necessidades do utilizador, oferecendo uma experiência de aprendizagem personalizada a cada indivíduo.

O MRI tem como objetivo principal apresentar recomendações aos utilizadores. Estas recomendações podem ser apresentadas na forma de hiperligações para recursos disponíveis no LCMS e são compatíveis com o perfil do utilizador. O modelo de utilizador é usado com o objetivo de representar os utilizadores. Este modelo é atualizado pelo LCMS, através dos serviços disponibilizados, tendo em consideração a interação do utilizador com a plataforma de aprendizagem [Tavares, Faria e Martins, 2012] .

Um dos requisitos propostos inicialmente para o LCMS era o desenvolvimento de um mecanismo que ofereça suporte inteligente aos utilizadores, na forma de sugestões personalizadas de recursos ou utilizadores. A inclusão deste tipo de mecanismos tem como objetivo sugerir os recursos potencialmente mais relevantes para cada indivíduo [Freedman, 2000][Lane, 2006].

As recomendações produzidas correspondem a recursos ou outros utilizadores. Os recursos são pacotes do tipo SCORM (*Shareable Content Object Reference Model*) disponíveis num repositório. Os utilizadores correspondem a outros utilizadores da plataforma cujo os perfis e competências podem contribuir para o processo de formação do aluno, a recomendação de utilizadores visa a potenciar o processo de aprendizagem colaborativa.

De forma resumida, este projeto visa implementar um sistema de recomendação inteligente para sistemas de *e-learning* que permita alcançar os seguintes objetivos específicos:

- O MRI deve ser capaz de sugerir conteúdos ou utilizadores adaptadas ao perfil do utilizador;
- As recomendações efetuadas pelo MRI devem estar enquadradas num contexto de aprendizagem formal ou num contexto de aprendizagem não formal;
- Definir um modelo de utilizador, que permite representar utilizadores;
- Gestão da informação sobre os utilizadores e recursos, relevante ao MRI;
- Criação de uma ferramenta de apreciação de recursos (gosto/não gosto).

1.3 Relevância do tema

Na sociedade atual do conhecimento, a formação ao longo da vida promove uma constante aprendizagem, aumentando assim a procura por sistemas de aprendizagem. Um dos maiores desafios na área do *e-learning* é dotar os LCMS com a capacidade adaptar o processo de aprendizagem a diferentes utilizadores [Ong and Ramachandran, 2003]. De forma a sugerir os recursos existentes ao aluno, foi desenvolvido um sistema de recomendação inteligente orientado a serviços. Este sistema foi desenvolvido na forma de um serviço web mantendo assim a sua independência e facilitando a integração com qualquer tipo de LCMS.

A introdução de tecnologias de apoio à educação e formação já é uma prática comum nas organizações. De forma geral, o desenvolvimento de um sistema de recomendação no domínio da educação tem como objetivo o desenvolvimento de um sistema inteligente capaz de auxiliar os formadores e os formandos num ambiente de educação ou formação. Geralmente, nos sistemas de aprendizagem existe noção dos objetos de aprendizagem existentes, conteúdos, testes, questões e respostas. Todos os objetos de aprendizagem estão interligados, sendo possível ao sistema saber, por exemplo, quais foram as questões que o utilizador respondeu, correta ou incorretamente, e a que conteúdos estão ligados essas questões [Lane, 2006] .

Este tipo de abordagem torna sistema mais fechado onde existe uma maior necessidade de manutenção. A criação de conteúdos próprios para o sistema e a interligação entre os diferentes tipos de conteúdos é necessária. Nestes sistemas é necessário recorrer a metodologias mais complexas, para efetuar tarefas como, recomendar um livro ao utilizador ou detetar o nível de conhecimento que o utilizador tem sobre os temas abordados num livro, [Lane, 2006] .

Os sistemas de recomendação permitem utilizar a informação existente no sistema de forma a produzir sugestões de recursos, sem existir um aumento significativo na necessidade de gestão e manutenção do sistema. Quando o ser humano é confrontado com um novo domínio de conhecimento, este frequentemente depende da experiência e conhecimento de outros

indivíduos. Os sistemas de recomendação são uma resposta tecnológica para esta situação, é possível observar que de uma forma geral, recomendações, são muitas das vezes aceites de forma natural e intuitiva [Berka and Salzburg, 2004].

1.4 Motivação

Num mundo sempre a mudar, a necessidade de uma educação continuada aumenta dia para dia, a necessidade de aceder à educação a diferentes níveis e a diferentes ritmos encontra-se refletida na procura por novas técnicas que ajudam a identificar a melhor estratégia para satisfazer as necessidades de cada utilizador [Rane, 2005].

A utilização massificada de sistemas tecnológicos apresenta oportunidades em diversas áreas. Tirando partido de informação, que muitos sistemas já possuem, é possível oferecer ao utilizador um sistema melhorado. Este trabalho explora técnicas de recomendação, já utilizadas noutras áreas, de forma a dotar sistemas de *e-learning* com a capacidade de adaptação às necessidades de aprendizagem do utilizador, tendo em consideração o seu nível de conhecimento e a progressão que estes têm no processo de aprendizagem.

1.5 Contributos

Este trabalho contribuiu de forma específica para o desenvolvimento de uma plataforma de *e-learning* onde foram interligadas diferentes tecnologias. De uma forma geral este trabalho apresenta a definição de novos modelos que podem ser aplicados a outros problemas e utilizados noutros ambientes.

O algoritmo de similaridade desenvolvido mostra que a simplificação é muitas vezes benéfica e nem sempre é acompanhada de uma diminuição de eficiência. A criação do algoritmo híbrido permite aproveitar vantagens existentes nas várias técnicas existentes. A forma como essas tecnologias e técnicas foram interligadas permite melhorar a produção de recomendações.

1.6 Estrutura da dissertação

Esta tese é composta por 5 capítulos. O segundo capítulo apresenta o problema, sendo realizada primeiro uma breve revisão sobre a evolução dos sistemas de recomendação ligados a plataformas de aprendizagem, bem como um resumo do estado da arte. Neste capítulo são também descritos os pré-requisitos do projeto e no final é efetuado um pequeno resumo e uma análise crítica.

No terceiro capítulo é apresentada a análise teórica e uma solução para o projeto. Dos diferentes pontos abordados neste capítulo será dada especial importância ao sistema de recomendação híbrido.

O quarto capítulo apresenta uma descrição da implementação onde é detalhada a parte mais relevante desenvolvida bem como realizada uma análise das escolhas efetuadas. Neste capítulo também é demonstrado um exemplo de utilização. Este exemplo foi construído a partir de dados fictícios que servem apenas para testar o sistema num ambiente controlado onde é possível prever os resultados desejados.

O quinto e último capítulo desta tese apresenta as conclusões retiradas neste projeto. No final também são mencionados algumas funcionalidades e trabalho que pode ser desenvolvido no futuro, de modo a melhorar este trabalho. No final do documento estão apresentadas em anexos as estruturas utilizadas bem como exemplos de utilização.

2 Caracterização do problema

Este capítulo tem por finalidade fazer um levantamento das principais técnicas usadas nos sistemas de recomendação, existentes no mercado. São também analisadas diferentes abordagens à resolução de problemas idênticos. Finalmente de forma sucinta e mais concreta é descrito o problema que este projeto pretende resolver.

2.1 Metodologias e técnicas

A utilização de sistemas inteligentes em plataformas de aprendizagem tem como objetivo facilitar a aprendizagem aos utilizadores. Estes sistemas inteligentes têm vindo a evoluir ao longo dos anos, surgindo inicialmente divididos em duas grandes categorias: sistemas orientados à pedagogia e sistemas orientados à execução. Sistemas orientados à pedagogia são sistemas focados em sequenciar e ensinar conteúdo, enquanto sistemas orientados à execução são basicamente sistemas de simulação. Nos sistemas orientados à execução é disponibilizado um ambiente controlado onde um estudante pode praticar e ganhar competências [Tom Murray, 1999] [Lane, 2006] [Goodkovsky, 2006].

Com a evolução, estes dois tipos de sistemas, foram-se fundindo sendo comum hoje em dia encontrar sistemas orientados à pedagogia que incluem ambientes de simulação e que incentivam o estudante a interagir de uma forma prática com o sistema [Tom Murray, 1999] [Lane, 2006]. A massificação da utilização da internet levou ao aparecimento de plataformas de aprendizagem *online*. De forma a aumentar a eficácia e eficiência destas plataformas estão a ser incluídos módulos inteligentes com o objetivo de auxiliar e personalizar a experiência de cada utilizador [Gemmis and Semeraro, 2011].

Devido à complexidade existente nos sistemas de tutoria, estes sistemas são divididos em partes. Cada parte pretende resolver um fragmento do problema e não o processo de tutoria inteiro. O comportamento destes sistemas depende principalmente da experiência do utilizador, da implementação das melhores práticas, em paradigmas pedagógicos específicos, bem como em técnicas de inteligência artificial e tecnologias disponíveis. Na altura em que começa a surgir o conceito de sistemas inteligentes de tutoria, os arquitetos de *software* tendencialmente conceberam bibliotecas ou *frameworks* conceptuais de forma a facilitar e tornar mais eficiente a construção de um ITS (*Intelligent Tutoring System*). No entanto, por definição, as generalizações são sempre limitadas e propensas a erros. Os sistemas que são desenvolvidos nesta área são geralmente muito complexos, muito específicos e não reutilizáveis [Tom Murray, 1999].

Este trabalho foca-se na parte dos sistemas de recomendação inteligente em sistemas de aprendizagem, assim sendo, a plataforma desenvolvida é um sistema que propicia a aprendizagem autónoma onde é considerado o domínio no qual o sistema está inserido bem como as diferentes características e competências de cada utilizador.

Diversas técnicas são usadas na implementação de sistemas de recomendação, no entanto a maior parte destes sistemas usam somente uma ou duas técnicas. Na solução implementada optou-se por fazer a junção de várias técnicas de forma a criar um sistema híbrido. Os sistemas híbridos combinam diferentes técnicas de forma a tirarem partido do melhor que cada técnica tem para oferecer [Berka and Salzburg, 2004].

Os sistemas de recomendação inteligente combinam técnicas utilizadas na modelação de utilizadores, filtragem de informação, agregação de informação em grupos e métodos de classificação de conteúdos. Com a combinação destas técnicas é possível ao sistema efetuar uma recomendação mais inteligente aos utilizadores. A utilização deste tipo de sistemas leva à prestação de um melhor serviço quando comparada com a utilização de filtros, porque têm a capacidade de providenciar informação personalizada e com significado para o utilizador [Shani and Gunawardana, 2009].

2.1.1 Filtros

Existem duas técnicas de recomendação que se destacaram: filtros baseados no conteúdo, CBF (*content-based filtering*) e filtros baseados em colaboração CF (*collaborative-based filtering*). Os filtros baseados no conteúdo efetuam recomendações ao utilizador cujo conteúdo seja semelhante a conteúdos consultados anteriormente pelo utilizador. Os filtros colaborativos efetuam recomendações tendo em consideração a avaliação efetuada por outros utilizadores cujo perfil seja semelhante ao do utilizador [Gemmis and Semeraro, 2011] [Berka and Salzburg, 2004].

Outra técnica de filtragem que tem vindo a ganhar alguma notoriedade nesta área são os filtros baseados em conhecimento, *KBF (Knowledge based Filtering)*. Esta técnica depende da representação explícita do conhecimento, geralmente coleções de afirmações, ontologias ou outras formas de sistemas de regras. Apesar da alta performance e flexibilidade desta técnica, em sistemas com grande dependência em conteúdos ou semânticas sociais é possível obter resultados de uma forma mais eficiente usando técnicas mais especializadas, tais como os filtros baseados no conteúdo e os filtros baseados em colaboração [Berka and Salzburg, 2004].

A avaliação, dos resultados, efetuada pelos utilizadores desempenha um papel muito importante na eficácia dos filtros utilizados. Assim sendo, a interação entre o LCMS e os utilizadores, é refletida diretamente nos resultados das recomendações. Métodos de classificação e funcionalidades de *feedback* que dão ao utilizador a capacidade de avaliar um determinado conteúdo são muito importantes, bem como a capacidade de manter um histórico do número de consultas e qual o utilizador que efetuou essa consultas [Gemmis and Semeraro, 2011] [Shani and Gunawardana, 2009].

2.1.2 Algoritmos agregadores (*Clustering*)

Em *data mining* as técnicas de *clustering* foram desenvolvidas com o intuito de identificar e agrupar objetos. Esta classificação é efetuada com base na análise da informação existente sobre esses objetos. Usando este tipo de técnicas é possível criar conjuntos de objetos que de alguma forma contenham similaridades entre si. Ao longo do tempo foram surgindo diferentes algoritmos de *clustering* com o objetivo de aumentar a qualidade dos resultados obtidos. Existem diferentes tipos de modelos de *clustering*, como por exemplo: modelos de conectividade, modelos centrados, modelos distribuídos, modelos de densidade, entre outros [Andritsos, 2002] [Fung, 2001]. Neste projeto foram considerados diferentes algoritmos de *clustering* entre os quais se destacaram o DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) e SOM (*self-organizing map*).

O modelo de *clustering* baseado na densidade define os conjuntos de objetos ou *clusters*, criando relações entre os objetos que se encontram nas áreas mais densas da região. O algoritmo DBSCAN é um algoritmo de baseado no modelo de densidade [Kriegel, Sander and Xu, 1998] [Rehman, 2006]. O DBSCAN é um dos algoritmos de *clustering* mais usados. No DBSCAN é necessário definir dois parâmetros globais o *Eps* e o *minPoints*, sendo estes valores comuns a todos os *clusters*. O *Eps* define a distância máxima entre dois pontos para que estes possam pertencer ao mesmo *cluster*. O *minPoints* define o número mínimo de objetos necessários para se criar *cluster*. Uma outra característica que torna o DBSCAN apelativo como algoritmo de *clustering* é a noção de ruído que permite remover objetos não desejados dos *clusters* ou agrupar todos os objetos considerados ruído num *cluster* específico [Kriegel, Sander and Xu, 1998] [Rehman, 2006] [Ji-Rong, Jian-Yun and Hong-Jiang, 2001].

Na tabela 1, são apresentadas algumas vantagens e desvantagens do algoritmo de *clustering* DBSCAN no contexto deste projeto. Uma das vantagens apresentadas na utilização do

DBSCAN é o facto de não ser necessário definir inicialmente o número de *clusters*, permitindo que o sistema se adapte a diferentes ambientes e minimiza a necessidade de intervenção com o aumento ou diminuição de objetos disponíveis ao longo do tempo [DBSCAN, 2012].

Tabela 1 - Vantagens e desvantagens do DBSCAN

| Vantagens do DBSCAN | Desvantagens do DBSCAN |
|--|---|
| <ul style="list-style-type: none"> - Não é necessário especificar o número de <i>clusters</i> à priori; - Consegue definir <i>clusters</i> de diferentes formas, como por exemplo criar um <i>cluster</i> à volta de outro; - Utiliza a noção de ruído; - Apenas requer dois parâmetros de entrada; - Não depende da ordem pela qual os pontos estão organizados (exceto em caso de objetos que se encontram na fronteira). | <ul style="list-style-type: none"> - Depende fortemente da medida de distância utilizada. - Não consegue criar <i>clusters</i> de objetos de uma região que contenha grandes diferenças de densidade. |

A figura 1 apresenta um exemplo da aplicação do algoritmo DBSCAN a um conjunto de dados. Nesta imagem, cada *cluster* está representado por uma cor. Assim, é possível observar a capacidade que o DBSCAN tem de criar *clusters* com formas diferentes. A figura 1 mostra, um *cluster* azul e um *cluster* vermelho. Os pontos a cinzento, na região apresentada, não foram inseridos em qualquer *cluster*, tendo sido considerados ruído [DBSCAN, 2012].

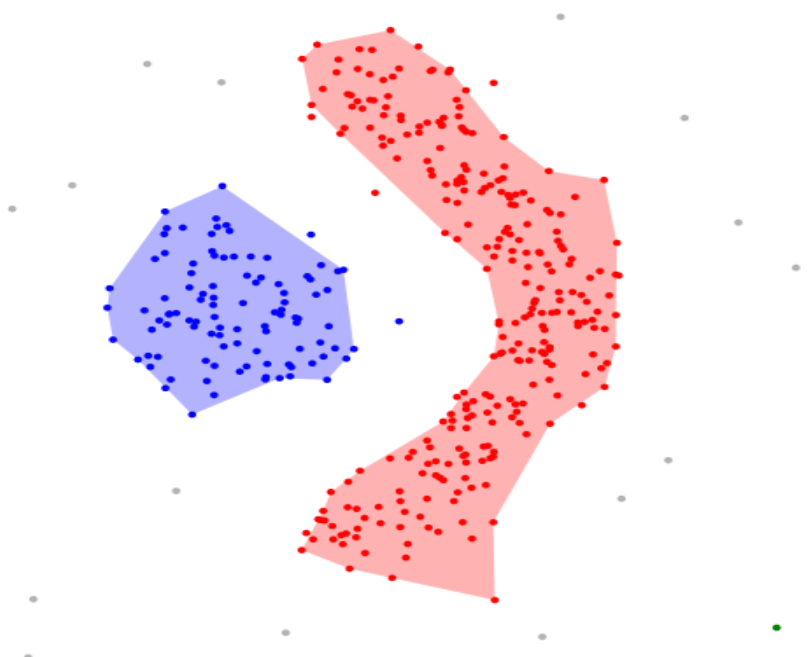


Figura 1 - Exemplo da aplicação do algoritmo DBSCAN [DBSCAN, 2012]

O SOM (*self-organizing maps*), ou redes *Kohonen*, é um tipo de rede neuronal artificial com a capacidade de efetuar *clustering* sobre informação e capaz de descobrir padrões que muitas das vezes não são detetados facilmente. Nas redes neuronais artificiais o processo geralmente é dividido em dois modos, um modo de treino, onde a rede artificial é treinada, e um modo de mapeamento onde a rede processa a informação [Marinai, 2007] [Fort, Letremy and Cottrell, 2002]. O treino das redes neuronais pode ser supervisionado ou não supervisionado. Uma das vantagens da utilização de SOM é o facto de utilizarem um mecanismo de aprendizagem não-supervisionada [Smith and Alan, 2003].

A tabela 2 descreve as vantagens e desvantagens de SOM no contexto deste projeto. De realçar a capacidade de agrupar informação onde a similaridade não é facilmente visível. No entanto, este método exige a especificação do número inicial de *clusters*.

Tabela 2 - Vantagens e desvantagens de SOM

| Vantagens de SOM | Desvantagens de SOM |
|--|--|
| <ul style="list-style-type: none"> - Capacidade de agrupar informação onde a similaridade não seja facilmente visível; - Rapidez de convergência; - Aprendizagem não-supervisionada; - Não é necessário ajustar parâmetros iniciais. | <ul style="list-style-type: none"> - O desempenho depende da medida de distância utilizada; - É necessário especificar o número de <i>clusters</i> inicialmente. |

2.1.3 Exemplos de sistemas de recomendação

Neste subcapítulo foram estudados alguns sistemas de recomendação. Na tabela 3 são apresentados os sistemas que disponibilizam técnicas de filtragem e algoritmos de agregação de informação. Os sistemas descritos na tabela 3 com uma licença que permite o seu uso de forma livre, são projetos que, ou se encontram em constante desenvolvimento ou foram descontinuados.

Tabela 3 - Sistemas de recomendação e *data-mining*

| Nome | Tipo | Tecnologias | Interface | Código | Licença |
|---|----------------------|-------------------------------|---------------|--------------------|-----------------------------|
| Suggest [Suggest, 2012] | Biblioteca | ANSI C | API | <i>Open Source</i> | Limitada |
| Easyrec [Easyrec, 2012] | Aplicação Web | Java, Mysql, Restfull, tomcat | Rest API, WUI | <i>Open source</i> | Limitada |
| LensKit [LensKit, 2012] | Biblioteca | Java, Maven | API | <i>Open source</i> | GNU General Public License |
| Apache Mahout [Apache Mahout, 2012] | Biblioteca | Java, Maven | CMD | <i>Open source</i> | Apache License, Version 2.0 |
| Duine Framework [Duine Framework, 2012] | Biblioteca | Java, Maven | API | <i>Open source</i> | GNU General Public License |
| Weka [Weka, 2012] | Biblioteca, software | Java | API, GUI | <i>Open source</i> | GNU General Public License |
| Lingpipe [Lingpipe, 2012] | Biblioteca | Java, mysql | Web, GUI, CMD | <i>Open source</i> | Limitada |
| Rapidminer [Rapidminer, 2012] | Biblioteca, software | Java | API, GUI | <i>Open Source</i> | GNU AGPL |
| Knime [Knime, 2012] | Biblioteca, software | Java | API, GUI | <i>Open Source</i> | GNU General Public License |
| Elki [Elki, 2012] | Biblioteca, software | Java | API, GUI | <i>Open Source</i> | AGPLv3 |

A tabela 4 pretende mostrar as principais vantagens e desvantagens dos sistemas apresentados na tabela 3. As vantagens e desvantagens descritas são pareceres relativos ao MRI.

Tabela 4 - Vantagens e desvantagens dos sistemas apresentados

| Nome | Vantagens | Desvantagens |
|-----------------|---|--|
| Suggest | Algoritmos de recomendação escaláveis. | Licença limitada, não tem algoritmos de <i>clustering</i> , linguagem diferente da linguagem utilizada no MRI. |
| Easyrec | Facilidade de uso, integração e escalável. Utiliza as mesmas tecnologias que o MRI. | Licença limitada, desenvolvida especificamente para comércio electrónico, seria necessário adaptar. |
| LensKit | Licença livre, desenvolvido para aplicações web. | Em desenvolvimento. Não utiliza algoritmos de <i>clustering</i> . |
| Apache Mahout | Licença livre, algoritmos de <i>clustering</i> e filtros colaborativos | Não tem o algoritmo de clustering DBSCAN, ainda se encontra num estado inicial de desenvolvimento. Necessidade de implementação de métodos de classificação de informação. |
| Duine Framework | Licença livre, algoritmos de recomendação, filtros colaborativos. | <i>Software</i> beta, encontra-se em desenvolvimento, falta de documentação. |
| Weka | Licença livre, algoritmo de <i>clustering</i> DBSCAN. | Documentação desatualizada, é Necessidade de implementação de métodos de classificação de informação. |
| Lingpipe | Algoritmo de <i>clustering</i> DBSCAN. Comparação de palavras. | Licença limitada, não possui algoritmos de recomendação |
| Rapidminer | Licença livre, algoritmo de <i>clustering</i> DBSCAN. | Não possui algoritmos de recomendação. Necessidade de implementação de métodos de classificação de informação. Desenvolvimento mais direcionado para interfaces gráficas. |
| Knime | Licença livre, algoritmo de <i>clustering</i> DBSCAN. | Não possui algoritmos de recomendação. Necessidade de implementação de métodos de classificação de informação. Desenvolvimento mais direcionado para interfaces gráficas. |
| Elki | Licença livre, algoritmo de <i>clustering</i> DBSCAN. | Não possui algoritmos de recomendação. Necessidade de implementação de métodos de classificação de informação. |

2.2 Especificação do problema

A individualização é um fator chave para alcançar uma boa aprendizagem [Lane, 2006]. Quando se menciona individualização ou customização em sistemas de aprendizagem refere-se à capacidade que o sistema tem de se adaptar a diferentes utilizadores com ritmos de aprendizagem e níveis de conhecimento diferentes. De forma a alcançar esta característica, os mais avançados LCMS usam técnicas provenientes da área dos sistemas de tutoria inteligente [Lane, 2006]. A utilização destas técnicas tenta completar o papel do tutor, tendo como objetivo recomendar os objetos de aprendizagem mais indicados para cada utilizador. Neste projeto foi desenvolvido um sistema híbrido de recomendação inteligente que tem como finalidade sugerir recursos ou outros utilizadores a um aluno. Para que o sistema funcione corretamente este deve ser capaz de avaliar o conhecimento que cada utilizador possui nos diferentes conceitos e sugerir corretamente o próximo recurso que esse utilizador deve estudar [Gamboa and Fred, 2003] [Lane, 2006].

Outros problemas que surgem no desenvolvimento de um sistema de recomendação é a necessidade de gerir uma elevada quantidade de dados [Cattuto, Benz, Hotho and Stumme, 2008] [Markines, Cattuto, Menczer, Benz, Hotho and Stumme, 2009]. É essencial submeter todos os dados a um processo onde é necessário abordar problemas como palavras iguais que em contextos diferentes apresentam significados semânticos distintos.

Em plataformas *online* a informação é geralmente introduzida pelos próprios utilizadores. A forma como essa informação é tratada na sua inserção depende do LCMS, no entanto é natural que exista informação incompleta ou com erros no sistema. Quando é necessário lidar com uma elevada quantidade de informação é necessário ter em mente que a informação muda ao longo do tempo, nova informação é adicionada, alguma atualizada, outra tornam-se obsoleta e alguma é removida [Tavares, Faria e Martins, 2012].

De forma a diminuir ao ruído existente nos dados da plataforma, é necessário processar a informação utilizando uma técnica de *stemming*. Relativamente aos recursos, estes deverão ser do tipo SCORM e estar representados por uma meta-descrição. Pretende-se assim, incentivar a uniformização da informação a ser utilizada na produção de recomendações.

Numa plataforma de aprendizagem os utilizadores podem estar associados a um determinado curso ou não ter nenhum curso associado. Assim sendo, existe a necessidade de criar dois tipos de abordagens diferentes, para estes dois tipos de contextos. De forma, a dotar o sistema com a capacidade de produzir a melhor recomendação possível, para um utilizador em específico, o nosso problema foi então dividido em dois tipos de contextos de aprendizagem, aprendizagem formal e aprendizagem não formal. Com dois tipos de contextos de aprendizagem diferentes é possível, por um lado, responder à necessidade de recomendar recursos de aprendizagem a utilizadores cujo objetivo é completar um curso e, por outro lado, recomendar recursos de aprendizagem a utilizadores que não estão associados a qualquer curso [Tavares, Faria e Martins, 2012].

2.3 Resumo

Este capítulo resume metodologias e técnicas de sistemas de tutoria e sistemas de recomendação. Analisar a história dos sistemas inteligentes ligados ao ensino e formação permite compreender a evolução dos mesmos e de que forma estes têm sido integrados, é possível observar a fusão de diferentes técnicas em sistemas únicos.

Existem diferentes tipos de sistemas de recomendação já implementados, no entanto, nem todas as soluções existentes apresentam uma solução adequada ao projeto apresentado. Assim sendo, a adoção de um destes sistemas, para suporte ao desenvolvimento do MRI, é acompanhado de um risco. Caso seja um sistema em desenvolvimento, a própria falta de maturidade do sistema pode causar problemas.

Ao analisar os sistemas e técnicas apresentados neste capítulo, foi possível concluir que apesar de existirem diferentes formas de abordar a implementação de um sistema inteligente de recomendação para o apoio ao ensino, o mais indicado será optar pelo desenvolvimento de um sistema híbrido. Um sistema híbrido tem a capacidade de combinar técnicas existentes, tirando assim partido do melhor que cada uma tem para oferecer.

Na segunda parte deste capítulo foi especificado o problema. No capítulo seguinte será apresentada a solução desenvolvida para responder a este problema bem como uma análise detalhada dos modelos desenvolvidos.

3 Desenvolvimento de modelos

O módulo de recomendação híbrido disponibiliza serviços que permitem ao LCMS solicitar recomendações específicas para cada utilizador. De forma a ser efetuada a atualização dos dados no MRI, estão disponíveis serviços que permitem a criação, atualização e remoção da informação dos utilizadores e da informação dos recursos sempre que seja administrativamente necessário. As recomendações podem ser obtidas em dois contextos, num contexto formal de aprendizagem ou num contexto não formal de aprendizagem.

O diagrama de componentes apresentados na figura 2 reflete a arquitetura distribuída em que as recomendações acerca dos recursos de aprendizagem mais relevantes para o utilizador e os mecanismos de modelação do utilizador são disponibilizados por três conjuntos de serviços distintos: os serviços do modelo de utilizador, os serviços dos recursos e os serviços de recomendação.

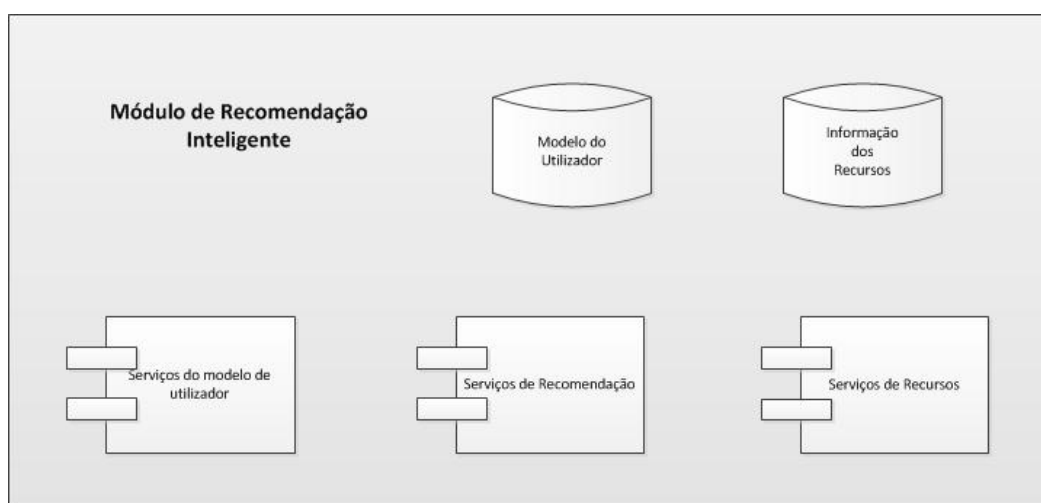


Figura 2 - Módulo de Recomendação Inteligente

As recomendações são geradas com base, quer no modelo de utilizador, quer nas pesquisas efetuadas. Esta informação é usada para encontrar recursos com meta-descrição compatível. As pesquisas efetuadas, bem como a informação existente no sistema, são processadas utilizando uma técnica de *stemming*.

Uma pesquisa é definida por uma *query* que consiste num conjunto de palavras-chave. Neste documento vamos utilizar a palavra *query* para designar uma pesquisa que é composta por um conjunto de palavras-chave.

Na figura 3 é apresentado um diagrama de sequência que define o processamento de um pedido não formal de recomendação. Neste pedido o utilizador, após autenticado, efetua um pedido de recomendação não formal ao efetuar uma *query*. Este pedido é efetuado pelo utilizador ao LCMS. O LCMS faz o pedido de sugestões ao MRI utilizando o identificador do utilizador e a *query* inserida pelo mesmo. No final, o resultado é devolvido ao utilizador.

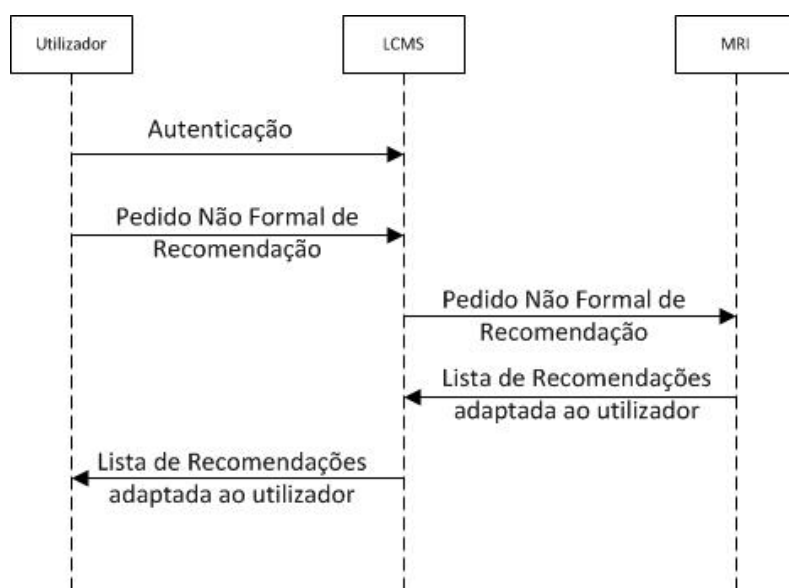


Figura 3 - Pedido não formal de recomendação

Na figura 4 é apresentado um diagrama de sequência de um pedido formal de recomendação. Neste pedido o LCMS utiliza o identificador do utilizador autenticado e efetua um pedido de sugestões ao MRI. No final o resultado é devolvido ao utilizador. O LCMS pode utilizar os serviços do MRI de forma a responder a diferentes tipos de abordagens, por exemplo, o utilizador pode iniciar o pedido de recomendação formal. Os diagramas de sequência apresentados apenas pretendem demonstrar uma possibilidade de utilização.

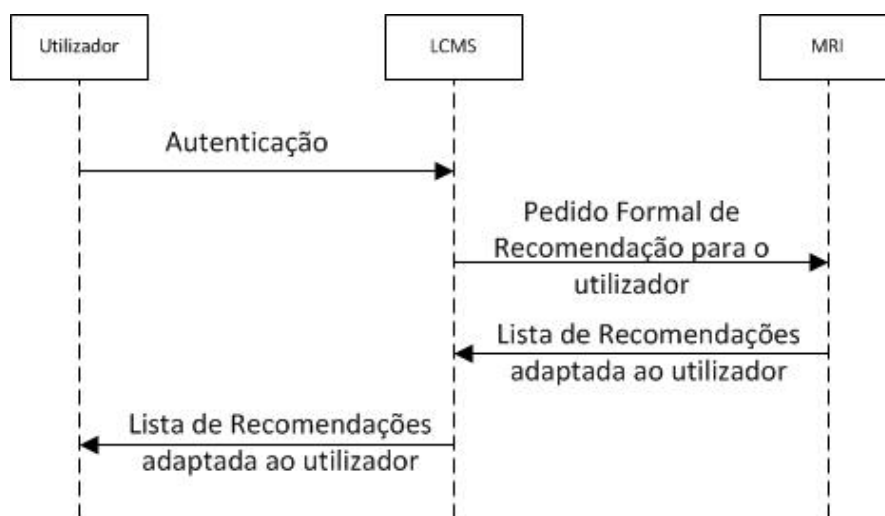


Figura 4 - Pedido formal de recomendação

A gestão de dados em grandes repositórios pode ser complexa e exigente em termos de recursos de computação [Song, Zhang and Giles, 2008]. Assim, os conteúdos foram agrupados em *clusters* utilizando o DBSCAN. Esta operação permite agrupar os conteúdos em *clusters*, tendo em consideração as semelhanças partilhadas nas suas descrições. De forma a tirar o melhor partido dos algoritmos de *clustering* foram analisadas diferentes métricas, como por exemplo distância euclidiana que é usada para medir a distância entre objetos [Bontchev and Vassileva, 2011] [Lee and Chun, 2008] (subcapítulo da 3.6.).

As técnicas usadas têm como objetivo produzir recomendações personalizadas, com base não só no perfil do utilizador, mas também considerar os comentários efetuados por outros utilizadores com um perfil idêntico. A avaliação dos recursos efetuada por outros utilizadores é considerada um bom indicador da utilidade do recurso [Ji-Rong, Jian-Yun and Hong-Jiang, 2002] [Rane, 2005]. A implementação de técnicas de filtragem tem como objetivo utilizar os dados das avaliações efetuadas de forma a melhorar os resultados da recomendação. No final o MRI deverá ser capaz de produzir recomendações de forma personalizada [Ji-Rong, Jian-Yun and Hong-Jiang, 2002] [Markines, Cattuto, Menczer, Benz, Hotho and Stumme, 2009].

3.1 Modelo de utilizador

A maioria das plataformas de aprendizagem possui um modelo de utilizador. No entanto, as plataformas não tiram partido destes dados [Cail, He, Li, Ma and Wen, 2004] [Cattuto, Benz, Hotho and Stumme, 2008] [Coelho, Figueiredo and Martins, 2009]. Para utilizar esta informação é necessário recolher e catalogar essa informação. De forma a atingir este propósito foi desenvolvido um modelo de utilizador, permitindo desta forma a uniformização dos dados disponíveis melhorando a compatibilidade entre os utilizadores [Coelho, Martins and Almeida, 2009] [Coelho, Martins, Figueiredo, 2009]. A criação de um modelo de utilizador

desenhado especificamente para o MRI facilita a gestão da informação, guardando os dados relevantes e ignorando os dados irrelevantes. Por exemplo, informação como o nome do utilizador, endereço electrónico ou a fotografia não é relevante para o MRI.

Pretende-se que as recomendações geradas sejam adequadas ao perfil do utilizador. Assim, existe a necessidade de modelar os utilizadores e de manter os respetivos modelos. De forma a facilitar a estruturação dos dados, o modelo de utilizador foi subdividido em dois domínios, o DID (*Domain Independent Data*) e o DDD (*Domain Dependent Data*) [Cail, He, Li, Ma and Wen, 2004] [Cattuto, Benz, Hotho and Stumme, 2008] [Coelho, Figueiredo and Martins, 2009] [Limam, Coquil, Kosch and Brunie, 2010].

A tabela 5 apresenta os dados identificados como potencialmente relevantes para o MRI. Assim sendo, a tabela apresenta dados que apesar de estarem identificados não estão a ser utilizados de momento. Com a identificação destes dados é pretendido prever necessidades que possam surgir numa fase mais avançada do projeto e-learning 3.0.

Tabela 5 - Modelo de utilizador

| Modelo de Utilizador | | | | |
|--------------------------------------|----------------|------------------------|------------------------------------|------------------------------|
| <i>Domain Independent Data (DID)</i> | | | <i>Domain Dependent Data (DDD)</i> | |
| Dados Pessoais | Profissional | Sociais | Curso | |
| Idade | Profissão | Histórico de Interação | - Domínio | - Conhecimento do utilizador |
| Género | Educação | Palavras-chave | - Tópicos | - Histórico do utilizador |
| Estado Civil | Especialidades | Favoritos | - Duração | - Performance do utilizador |
| Nº de Filhos | Línguas | Desporto | - Sequência | - Avaliação |
| Localização | | Deficiência | | |

A informação do DID é processada por técnicas de descoberta de conhecimento de forma a encontrar padrões que podem ser usados pelo MRI. Usando este tipo de informação é possível inferir gostos e conhecimento do utilizador. A informação existente no DID não necessita de constante atualização, a única exceção é a informação histórica do utilizador. O DID agrega informação do utilizador que não depende de nenhum domínio, sendo maioritariamente usada num contexto de aprendizagem informal, em que as recomendações não dependem da associação a um curso.

O DDD guarda a informação dos utilizadores ligada a um domínio de aprendizagem. Assim, a informação contida no DDD é utilizada por técnicas de descoberta de conhecimento para obter dados relevantes que podem ser utilizados por métodos de filtragem de acordo com os objetivos educacionais do utilizador. De forma a permitir ao MRI recomendar os recursos mais apropriados, os dados existentes no DDD necessitam de ser constantemente atualizados. O

DDD pretende guardar toda a informação que está de alguma forma associada com um domínio, sendo assim possível efetuar recomendações num contexto formal de aprendizagem.

3.2 Contextos de aprendizagem

As recomendações são produzidas em dois diferentes contextos de aprendizagem, o contexto formal e não formal. O contexto de aprendizagem indica o tipo de ambiente no qual o pedido de recomendação é solicitado, afetando assim o comportamento do MRI.

A figura 5 ilustra de forma simplificada os contextos existentes no algoritmo do MRI, sendo que num contexto de aprendizagem não formal as palavras-chave de pesquisa são inseridas pelo utilizador. No contexto formal de aprendizagem o próprio MRI vai buscar conceitos ao perfil do utilizador.

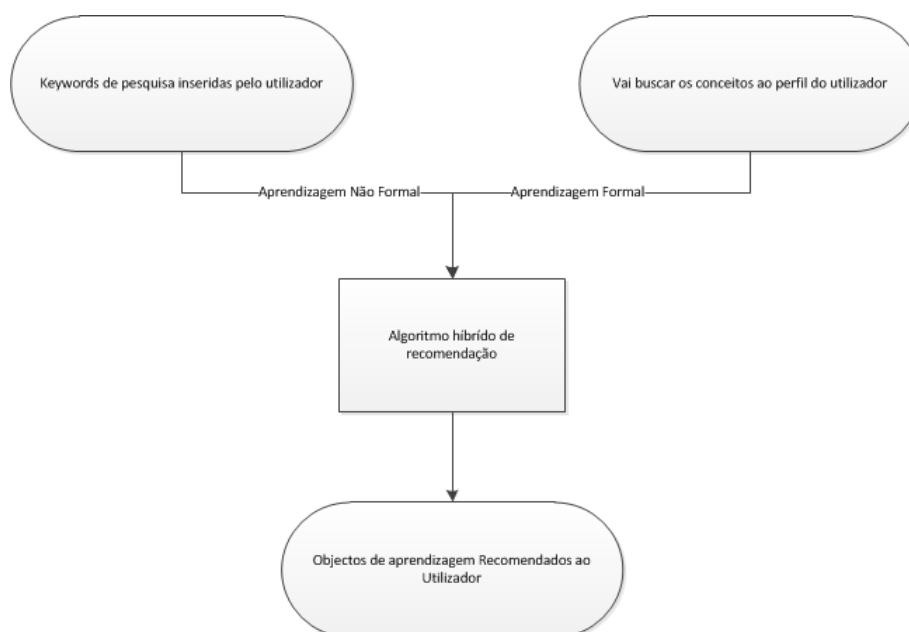


Figura 5 - Algoritmo do MRI simplificado

No contexto de aprendizagem formal o utilizador está associado a um curso que por sua vez terá um curriculum associado. Quando um utilizador faz um pedido de recomendação, os resultados do MRI vão ter em consideração o domínio e os tópicos que estão associados ao curso pretendido. As recomendações têm em consideração a informação existente no modelo de utilizador, sendo a informação existente do DDD a que mais influencia as sugestões dadas no contexto de aprendizagem formal. Os recursos sugeridos devem auxiliar o utilizador a progredir no curso de forma a completar o curso.

No contexto de aprendizagem não formal, o utilizador tem a iniciativa de pesquisar recursos existentes, sem ter qualquer plano de formação pré-definido. Enquanto que no contexto de aprendizagem formal existe a necessidade de completar um curso, no caso de um contexto de aprendizagem não formal essa limitação não existe, permitindo ao MRI recomendar

diferentes tipos de recursos que podem ser relevantes para o utilizador. Com este tipo de abordagem é possível utilizar um outro tipo de informação diretamente, que de outra forma provavelmente nunca seria considerada. Um exemplo é a informação contida no DID do modelo de utilizador que contém, dados sociais ou dados profissionais que podem ser usados pelo sistema de recomendação, como indicador das preferências dos utilizadores.

3.3 Stemming

A utilização de um processo de *stemming* é importante porque permite pré-processar os dados introduzidos no sistema eliminando ruído existente nessa informação. O processo de *stemming* consiste em reduzir as palavras ao seu elemento raiz ou radical, que por norma define o significado da palavra. A aplicação deste processo logo de início aumenta o sucesso da pesquisa efetuada. Para este propósito foram analisados algoritmos de *stemming* e as suas implementações em diferentes idiomas [Smirnov, 2008] [Willett, 2006]. Este processo é usado de forma a limpar os dados inseridos removendo aquilo que é considerado ruído, como por exemplo remover o plural ou gerúndio, aumentando desta forma a similaridade entre palavras. Existem 8 passos que podem ser aplicados a uma palavra:

- Remoção do Aumentativo/Diminutivo;
- Remoção do plural;
- Remoção dos acentos;
- Remoção das vogais;
- Remoção dos sufixos dos verbos;
- Remoção dos sufixos dos pronomes;
- Remoção dos advérbios;
- Remoção do feminino.

A aplicação do processo de *stemming* pretende extrair a informação que contém significado nos dados inseridos. No seguinte exemplo: 'árvore', 'arvoredo', 'arvorezinha', as palavras pertencem todas à mesma família. Na palavra, 'árvore', o acento serve apenas para indicar a pronúncia exata da palavra, a palavra 'arvoredo' é um conjunto de árvores e a palavra 'arvorezinha' significa árvore pequena. A aplicação do processo de *stemming* no exemplo apresentado iria transformar todas as palavras na palavra 'arvore', aumentando desta forma a similaridade entre os recursos ou utilizadores representados pelas diferentes palavras.

3.4 SCORM

A uniformização das informações dos recursos permite simplificar tarefas de gestão e integração do MRI com outros sistemas. Sendo o SCORM [SCORM, 2012] uma norma usada extensivamente com o objetivo de uniformizar a representação de recursos, adotar esta norma permite simplificar as tarefas de gestão da meta-informação dos recursos, bem como facilitar a reutilização e integração com outros sistemas [Tzone, Tsai, Lee and Chiu, 2007].

A norma SCORM utiliza um manifesto com a meta descrição do recurso, de acordo com a norma IEEE (*Institute of Electrical and Electronics Engineers*) LOM (*Learning Object Metadata*) [IEEE, 2002]. O MRI utiliza esta descrição acerca dos recursos para aceder diretamente à descrição do recurso.

Um objeto de aprendizagem pode conter diferentes tipos de dados como por exemplo, texto, imagem, áudio ou vídeo. Um modelo de dados é utilizado para descrever e representar um recurso [Bontchev and Vassileva, 2011]. Os recursos recomendados pelo MRI utilizarão a norma SCORM que são descritos pela informação existente no ficheiro de meta-dados.

3.5 Clustering

O aumento de conteúdos na internet fez com que a importância dos motores de pesquisa crescesse, levando os utilizadores exigirem cada vez mais uma maior eficácia nos resultados obtidos. A investigação nesta área estimulou o desenvolvimento de sistemas de recolha de informação e métodos de pesquisa utilizando técnicas de *clustering*. O algoritmo escolhido para o projeto foi o DBSCAN que é um algoritmo baseado na densidade de uma região, com a particularidade de não necessitar de ser definido o número inicial de *clusters* [Bontchev and Vassileva, 2011] [Ji-Rong and Hong-Jiang, 2003] [Ji-Rong, Jian-Yun and Hong-Jiang, 2002] [Lane, 2006].

Como a maioria dos algoritmos de *clustering*, usando o DBSCAN não é possível avaliar a qualidade dos clusters obtidos. O DBSCAN localiza os objetos próximos numa determinada vizinhança, agrupando esses objetos em *clusters*. Este algoritmo depende de duas variáveis, o ϵ e o número mínimo de pontos. O ϵ representa a distância máxima a que um objeto pode estar de um vizinho para que ambos possam pertencer ao mesmo *cluster*. E o número mínimo de pontos, que como o nome indica, define o número mínimo de pontos que é necessário para se criar um *cluster*. Caso os objetos não respeitem cumulativamente os dois critérios, serão considerados ruído, não sendo associados a qualquer *cluster*. É necessário ter em consideração que a qualidade da medida de distância utilizada influencia diretamente os resultados obtidos pelo DBSCAN.

A eficiência do processo de *clustering* depende da quantidade de informação a processar e do poder de processamento do servidor [Tsai and Wu, 2009]. Para garantir a obtenção de recomendações em tempo útil, optou-se por limitar a execução do algoritmo de *clustering*. No

entanto, este algoritmo deve ser executado periodicamente de forma a serem consideradas novas atualizações. A figura 6 contém o pseudo código utilizado por base para a implementação do algoritmo de *clustering* DBSCAN [DBSCAN, 2012].

```
// Função inicial utilizada para criar os clusters
// Deteta pontos considerados ruído
DBSCAN(D, eps, MinPts)
  C = 0
  Para cada ponto P não visitado no dataset D
    marcar P como visitado
    PontosVizinhos = getPontosVizinhos(P, eps)
    Se tamanho de PontosVizinhos < MinPts
      marcar P como ruído
    senao
      C = cluster seguinte
      expandirCluster(P, PontosVizinhos, C, eps, MinPts)

// Função utilizada para adicionar pontos
// vizinhos ao cluster
expandirCluster(P, PontosVizinhos, C, eps, MinPts)
  adicionar P ao cluster C
  Para cada ponto P' em PontosVizinhos
    Se P' nao foi visitado
      marcar P' como visitado
      PontosVizinhos' = regionQuery(P', eps)
      Se tamanho de PontosVizinhos' >= MinPts
        PontosVizinhos = PontosVizinhos + PontosVizinhos'
  Se P' nao pertencer a nenhum cluster
    adicionar P' ao cluster C

// Função utilizada para obter todos os pontos que se
// apresentam numa vizinhança do ponto P com distância
// máxima definida pelo eps
getPontosVizinhos (P, eps)
  devolve todos os pontos dentro da vizinhança(eps) de P
```

Figura 6 - Pseudo código do DBSCAN [DBSCAN, 2012]

Apesar de existirem diversas bibliotecas com este algoritmo já implementado, chegou-se à conclusão que a implementação de raiz do mesmo seria mais eficaz permitindo até alguma liberdade na forma como a representação dos objetos foi efetuada.

3.6 Similaridade

A similaridade corresponde a uma medida de semelhança entre os objetos dos *clusters*, podendo ser obtida através do cálculo de uma distância. Ser capaz de calcular corretamente a similaridade entre cada objeto irá influenciar positivamente os resultados do algoritmo de *clustering*, e por sua vez a qualidade das recomendações produzidas [Cail, He, Li, Ma and Wen, 2004] [Lee and Chun, 2008].

Ser capaz de calcular de forma eficaz a distância entre recursos é um dos processos mais importantes, porque a qualidade das recomendações efetuadas é diretamente afetada pela precisão, com a qual é calculada a distância, entre os diferentes objetos [Zanin, Cano, Buldú

and Celma, 2008] [Bilenko and Mooney, 2003]. De forma a fomentar a contínua melhoria do processo de similaridade do MRI, este processo foi dividido em diferentes passos, sendo estes: a divisão da *query* em palavras, comparação palavra a palavra, identificação da *query* com o maior número de palavras e a utilização desse número para normalização do processo.

Um dos algoritmos mais utilizados para a comparação de texto é o *Levenshtein distance algorithm* [Bilenko and Mooney, 2003]. Este algoritmo calcula o número de alterações necessárias para transformar uma palavra na palavra à qual está a ser comparada. É possível utilizar este algoritmo para obter um valor de distância entre duas palavras. No entanto, a ideia da utilização dos valores obtidos, através do algoritmo de distância de *Levenshtein*, foi abandonada durante o desenvolvimento do projeto porque a semântica das palavras no processo de comparação não foi considerada. Assim, o valor da distância não está a ser utilizado de momento, no entanto, pode vir a ser incluído no futuro caso sejam identificadas vantagens na sua utilização.

Para calcular a distância entre dois conjuntos de palavras-chave foi utilizada a seguinte fórmula, baseada na distância euclidiana:

$$d_{q_i c_j} = \sqrt{d_{p_{1q_i c_j}}^2 + d_{p_{2q_i c_j}}^2 + \dots + d_{p_{nq_i c_j}}^2} + D - D_{c_j}$$

A raiz quadrada e a potência utilizadas na distância euclidiana podem ser ignoradas no cálculo da distância. Esta simplificação não afeta a comparação a comparação de distâncias. Assim sendo, de forma a simplificar a expressão, aumentando-se a eficiência do processo de comparação, foi usada a seguinte expressão:

$$d_{q_i c_j} = d_{p_{1q_i c_j}} + d_{p_{2q_i c_j}} + \dots + d_{p_{nq_i c_j}} + D - D_{c_j}$$

Onde:

$$d_{p_{1q_i c_j}} = \begin{cases} 0 & \text{se } P_{1q_i} = P_{1c_j} \\ 1 & \text{se } P_{1q_i} \neq P_{1c_j} \end{cases}$$

Variáveis:

$d_{q_i c_j}$ - É a distância entre uma *query* e um *cluster*.

$d_{p_{1q_i c_j}}$ - É a distância entre uma palavra da *query* e o conjunto de palavras que representam um *cluster*.

Apresenta-se a seguir um exemplo que ilustra a utilização da expressão $d_{q_i c_j}$ para determinar a similaridade entre uma *query* e dois *clusters*.

Exemplo:

Query: $q = \text{'programação algoritmo java'}$

Conjunto 1: $c_1 = \text{'programação algoritmo perl iniciado analista'}$

Conjunto 2: $c_2 = \text{'programação algoritmo java sql perl ada erlang'}$

$$D_q = 3 \quad D_{c_1} = 5 \quad D_{c_2} = 7$$

Logo:

$$D = D_{c_2} = 7$$

Assim:

$$d_{q,c_1} = 0 + 0 + 1 + 1 + 1 + 7 - 5 = 5$$

$$d_{q,c_2} = 0 + 0 + 0 + 1 + 1 + 1 + 1 + 7 - 7 = 4$$

No exemplo apresentado é possível verificar que a distância entre a *query* e o conjunto 1 é superior à distância entre a *query* e o conjunto 2. Assim sendo, o grau de similaridade entre a *query* e o conjunto 2 é considerado superior.

3.7 Filtros implementados

A utilização de um sistema híbrido permite maior flexibilidade do sistema. Assim é possível beneficiar das diferentes vantagens que cada técnica de recomendação tem para oferecer, aumentando a adaptabilidade do MRI.

Esta secção foca as técnicas de filtragem utilizadas, filtros baseados em conteúdos e em colaboração, e a ferramenta de classificação.

3.7.1 Filtro baseado em conteúdo

Técnicas de filtragem baseadas em conteúdos são usadas frequentemente para filtrar informação. Nesta técnica os recursos são filtrados sendo aprovados com base na análise da meta descrição dos recursos. No MRI esta técnica é utilizada para reordenar os elementos do *cluster* atribuindo uma maior importância ao conteúdo com mais visualizações. Desta forma é possível atribuir uma maior importância a recursos mais utilizados.

Quando é efetuado o *clustering* de recursos já está a ser aplicado um filtro por proximidade, no entanto ainda é possível atribuir mais ou menos importância aos recursos sugeridos. Este filtro pode ser usado para filtrar os recursos de determinado *cluster* tendo em consideração o tipo de dificuldade ou tipo de requisitos necessários, dando maior importância a recursos menos exigentes se o utilizador for menos experiente naquela área específica ou está inscrito num curso para iniciados. Este filtro também pode ser usado de forma a bloquear conteúdo que foi previamente recomendado ao utilizador, evitando-se assim recomendações repetidas [Pazzani and Billsus, 2007] .

3.7.2 Filtro baseado em colaboração

Técnicas de filtragem baseadas em colaboração utilizam informação dada pelos utilizadores que é recolhida pela plataforma. Por exemplo, se um utilizador faz uma avaliação positiva de um recurso, utilizadores que no passado tenham avaliado positivamente recursos idênticos, terão uma maior probabilidade de considerar do novo recurso relevante. A utilização desta técnica pretende incorporar a informação dada pelos utilizadores e recolhida pela ferramenta de classificação do MRI.

Sendo esta uma técnica que já foi implementada com sucesso é necessário ter em consideração algumas das suas limitações. Os filtros colaborativos dependem fortemente dos utilizadores e precisam de tempo para obter os melhores resultados possíveis. É aconselhado não depender exclusivamente neste tipo de técnica de filtragem [Lee and Chun, 2008]. O problema pode surgir quando a apreciação for realizada por um único utilizador. No entanto, se considerarmos a avaliação efetuada por um grupo mais alargado de utilizadores, essa apreciação poderá ter mais significado.

De forma a melhorar as técnicas de filtragem é necessário utilizar algum tipo de mecanismo de classificação onde os utilizadores podem dar uma opinião sobre o recurso que estão a utilizar no momento [Jaschke, Marinho, Hotho, Schmidt-Thieme and Stumm, 2007]. Utilizando a informação recolhida pela ferramenta de classificação de recursos, o MRI está preparado para considerar opiniões como 'gosto' ou 'não gosto' sobre os recursos durante o processo de geração de sugestões de recursos.

A ferramenta de apreciação permite ao utilizador dar uma opinião simples sobre um recurso. Para que esta ferramenta tenha sucesso é muito importante que esta seja apresentada de forma clara ao utilizador. É importante que esteja claro para o utilizador que ao utilizar a ferramenta de classificação as futuras sugestões serão melhoradas, não só para outros utilizadores, mas também para ele próprio. Com a informação recolhida é possível ao MRI sugerir os recursos mais indicados para cada utilizador, no caso de não existir nenhuma informação dada pelos utilizadores o MRI terá dificuldade em melhorar as sugestões dadas.

3.8 Sistema de recomendação híbrido

O MRI é um sistema híbrido composto por um algoritmo de *clustering*, um algoritmo de similaridade, uma técnica de filtragem baseada em conteúdos, uma técnica de filtragem baseada em colaboração e uma ferramenta de classificação de recursos. O algoritmo de *clustering* efetua a segmentação de recursos, sendo apenas necessário aplicar os filtros aos recursos pertencentes ao *cluster* identificado, como sendo o mais indicado para aquela pesquisa [Bagatan, Boja and Cristian, 2011]. A identificação do *cluster* é efetuada utilizando também o algoritmo de similaridade apresentado anteriormente.

A figura 7 apresenta o fluxo de tratamento que uma *query* percorre no processo de produção de recomendações. Em primeiro lugar a *query* é tratada pelo processo de *stemming*, de seguida é identificado o *cluster*. Para este passo é utilizado o algoritmo de similaridade de para comparar a *query* com os diferentes conjuntos de palavras que representam cada *cluster*. Após ser identificado o *cluster*, o algoritmo de similaridade ordena os elementos do *cluster* tendo em consideração a informação de cada elemento. Nos passos seguintes são aplicados os filtros que tem por objetivo personalizar ainda mais o resultado. No final o utilizador pode utilizar a ferramenta de apreciação para avaliar os resultados apresentados.

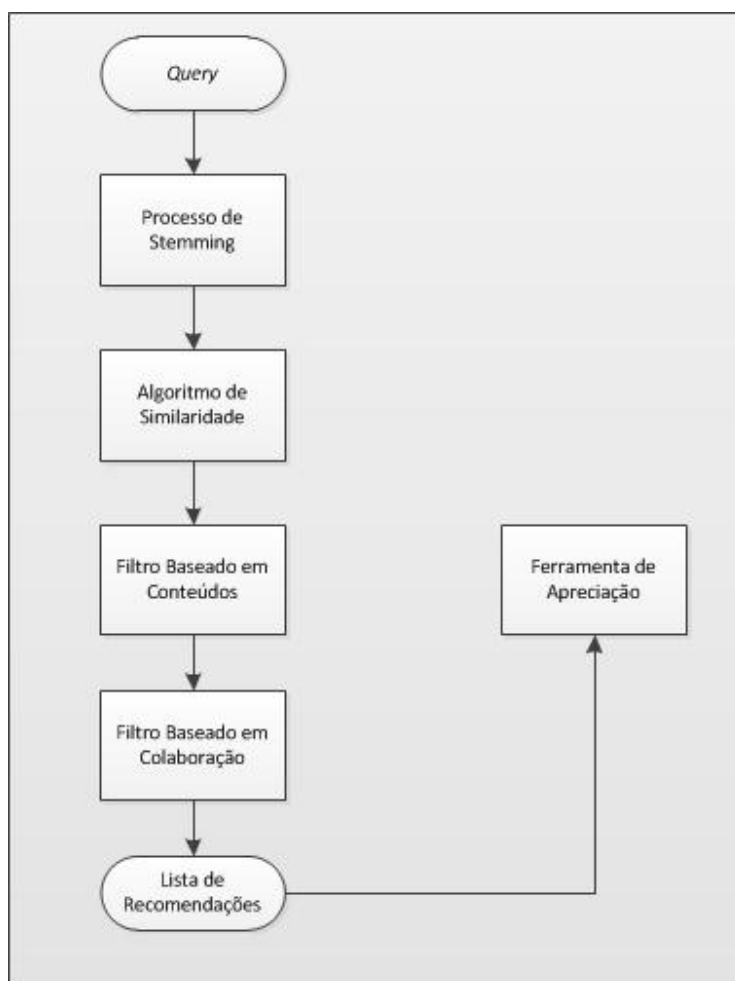


Figura 7 - Processo de produção de recomendações do MRI

Quando é efetuado um pedido de recomendação para um utilizador, a *query* inserida pelo utilizador é comparada com cada conjunto de palavras-chaves que representam cada um dos *clusters*. Assim, é identificado o *cluster* que se apresenta com maior semelhança. Após a identificação do *cluster* ter ocorrido, os elementos que o compõem são reorganizados com os filtros baseados no conteúdo e filtros colaborativos [Pazzani and Billsus, 2007]. A reorganização do *cluster* pretende ordenar os elementos, posicionando os elementos com maior relevância em primeiro lugar. As técnicas de filtragem baseadas no conteúdo analisam as meta-descrições dos recursos de forma a identificar os recursos mais indicados para o utilizador. O filtro colaborativo consiste na análise de comportamentos de utilizadores, utilizando a informação recolhida, sendo possível reorganizar os elementos do *cluster* dando maior importância aos recursos mais populares.

O sistema também considera se os recursos já foram vistos anteriormente pelo utilizador. A utilização de um sistema de recomendação híbrido ajuda a ultrapassar problemas que surgem quando são usados algoritmos de forma isolada. Por exemplo, problemas conhecidos como “*cold-start problem*” ou “*sparsity problem*”. O problema “*cold-start problem*” surge aquando da utilização recente do sistema, em que este ainda não possui muita informação. O problema “*sparsity problem*” consiste na falta de informação em diferentes campos da base de dados. Estes problemas impossibilitam o sistema de efetuar sugestões com a precisão desejada [Papagelis, Plexousakis and Kutsuras, 2005].

Como resultado existem dois tipos de recomendações: um dos tipos de recomendações corresponde a uma lista de sugestões de recursos na forma de objetos de aprendizagem, o outro tipo será uma lista de recomendações de utilizadores. Assim é possível recomendar conteúdo para estudo e também recomendar alguém especializado para dar apoio, isto no caso de ser necessário.

3.9 Resumo

O desenvolvimento do MRI como um serviço web permite aumentar a escalabilidade do sistema mantendo a sua independência. O estudo de técnicas como algoritmos de *clustering*, algoritmos de similaridade, técnicas de filtragem, entre outros, mostra que estas tecnologias estão a ser utilizadas com sucesso em diferentes campos.

No MRI existem processos que demonstraram ser mais importantes neste projeto, como é o caso do algoritmo de similaridade. O MRI tem uma dependência muito grande do algoritmo de similaridade, o que não acontece com outros processos utilizados, como por exemplo as técnicas de filtragem. As técnicas de filtragem são utilizadas para melhorar a qualidade das recomendações sugeridas.

4 Implementação

Neste capítulo é efetuada uma descrição da implementação das técnicas utilizadas para o desenvolvimento deste projeto. Encontra-se dividido em seis secções: descrição técnica, arquitetura do sistema, a base de dados, as ferramentas utilizadas, os serviços externos, os serviços internos e um exemplo de utilização. O subcapítulo da descrição técnica apresenta as tecnologias utilizadas para a implementação do sistema. Na parte da arquitetura do sistema é apresentado o esquema de alto nível do sistema. Nos restantes subcapítulos são apresentados o modelo da base de dados, as ferramentas de desenvolvimento utilizadas, os requisitos de *software* necessários para o funcionamento do sistema, uma descrição detalhada dos serviços disponibilizados e um exemplo de utilização.

4.1 Descrição técnica

O MRI foi desenvolvido como um conjunto de serviços web do tipo REST (*Representational State Transfer*) [REST, 2012]. O desenvolvimento foi efetuado sobre a *framework struts 2* [Struts2, 2012] em Java que oferece um padrão de desenvolvimento do tipo MVC (Modelo, Vista, Controlador) bastante utilizado no desenvolvimento de plataformas web dinâmicas. A utilização desta *framework* permitiu utilizar o plugin *struts-rest-plugin* para a implementação dos serviços web.

A escolha da utilização do protocolo REST para o desenvolvimento dos serviços é justificada por este ser um protocolo transparente, usar métodos http (*Hypertext Transfer Protocol*)

explicitamente e ser um protocolo mais legível. Outra das vantagens deste protocolo é ser menos exigente a nível de recursos o que o torna mais utilizado em plataformas móveis.

4.2 Arquitetura do sistema

A figura 8 ilustra o sistema desenvolvido, sendo possível observar o fluxo de sugestões desejado e como alguns componentes do algoritmo de recomendação estão interligadas.

O MRI foi dividido em duas partes, os serviços externos e os serviços internos. Os serviços externos publicam a interface que pode ser utilizada pelo LCMS para usufruir dos serviços disponibilizados pelo MRI. Os serviços internos servem para aumentar a flexibilidade do MRI e são utilizados internamente. Na figura 8 estão apresentados os serviços externos.

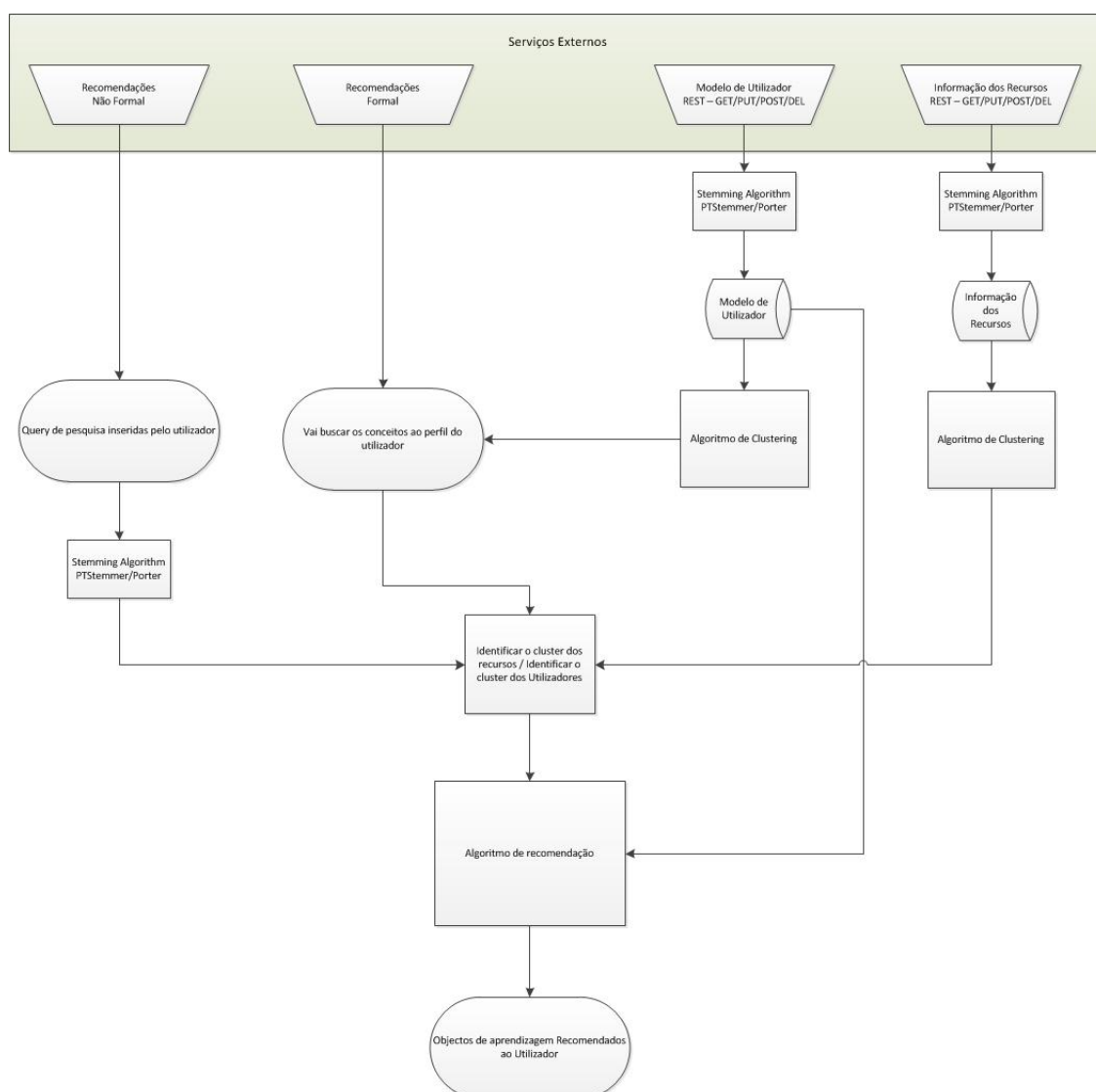


Figura 8 - Arquitetura do MRI, serviços externos

A gestão da informação sobre os recursos deve ser efetuada recorrendo ao uso dos serviços externos disponibilizados para o efeito. Estes serviços permitem implementar as funções de visualização, criação, atualização ou remoção da informação que descreve um recurso. A gestão da informação dos utilizadores está disponível através de serviços externos. Estes serviços permitem visualizar, criar, atualizar e remover a informação dos utilizadores. Os serviços internos permitem efetuar a criação dos *clusters*, de recursos e utilizadores, assim como visualizar os *clusters* criados. O MRI possui um ficheiro de configuração onde é possível definir algumas das propriedades do sistema, como por exemplo, ativar ou desativar algoritmos ou filtros.

Os serviços relacionados com os recursos disponibilizam apenas métodos para gerir a informação sobre os recursos útil ao MRI. Os objetivos do MRI não contemplam a criação ou gestão de recursos de aprendizagem. Estes devem estar alojados num repositório externo. O funcionamento do MRI requer apenas o acesso à meta-descrição dos recursos, bem como a indicação onde esses recursos se encontram.

O algoritmo de *clustering* não é só usado para agrupar recursos mas também é usado para agrupar os utilizadores. O *clustering* dos utilizadores é baseado na informação existente no modelo de utilizador, como por exemplo as competências profissionais ou competências sociais, e tem como objetivo agrupar utilizadores com perfis idênticos. Assim, na etapa responsável por obter os conceitos do modelo de utilizador, é possível expandir os conceitos do modelo de um utilizador. Ao aumentar a informação sobre o utilizador é possível aumentar a variedade das sugestões produzidas. O aumento de informação é conseguido com recurso a informação existente em modelos de utilizadores idênticos.

As *queries* usadas para definir um pedido de recomendação, bem como a informação dos utilizadores e recursos inserida no MRI, são processadas através do algoritmo de *stemming*. Este procedimento permite reduzir as palavras à sua forma radical. Palavras similares ou palavras da mesma família são reduzidas à mesma palavra, aumentando o sucesso do processo de comparação de palavras (ver subcapítulo 3.6).

No processo de identificação do *cluster*, é efetuada uma comparação entre as palavras-chave inseridas pelo utilizador, no contexto não formal de aprendizagem, e as palavras-chave que representam os *clusters*, sendo identificado o *cluster* que apresenta uma maior proximidade. De forma a aumentar a eficiência do processo de identificação do *cluster*, todos os *clusters* são representados por palavras-chave. As palavras-chave que representam o *cluster*, consistem nas palavras-chave mais frequentes na informação dos elementos que compõem o *cluster*. Desta forma, quando é necessário selecionar qual o *cluster* mais indicado para uma determinada pesquisa, apenas é necessário efetuar uma comparação com os conjuntos de palavras-chave que representam cada *cluster*. O mesmo processo de identificação é usado para efetuar a comparação entre os conceitos existentes no modelo do utilizador, no contexto formal de aprendizagem, e o *cluster* de recursos que apresenta uma maior proximidade.

No sistema de recomendação onde é aplicado o algoritmo de recomendação, são aplicados os filtros e ordenados os elementos do *cluster*. Para este efeito são utilizadas técnicas de filtragem baseadas em colaboração e técnicas de filtragem baseadas em conteúdo. O *cluster* obtido é ordenado de forma personalizada ao utilizador para o qual as sugestões são alvo. É dada prioridade a recursos que detêm melhor aceitação por parte de indivíduos que se encontram agrupados no mesmo *cluster* do utilizador.

4.3 Modelo da base de dados

Para suportar o sistema foi desenvolvida uma base de dados onde é guardada toda a informação necessária ao MRI. A figura 9 apresenta o modelo de dados da base de dados desenvolvida para o MRI. A base de dados do MRI guarda informações do modelo de utilizador, bem como as informações necessárias aos módulos, como é o caso do identificador do *cluster*, ou do número de ‘gostos’ e ‘cliques’ que o utilizador efetuou. Na base de dados também está armazenada a informação sobre os recursos e respetivos *clusters*. O histórico de ‘gostos’ e ‘cliques’ relaciona os utilizadores com os recursos ou outros utilizadores.

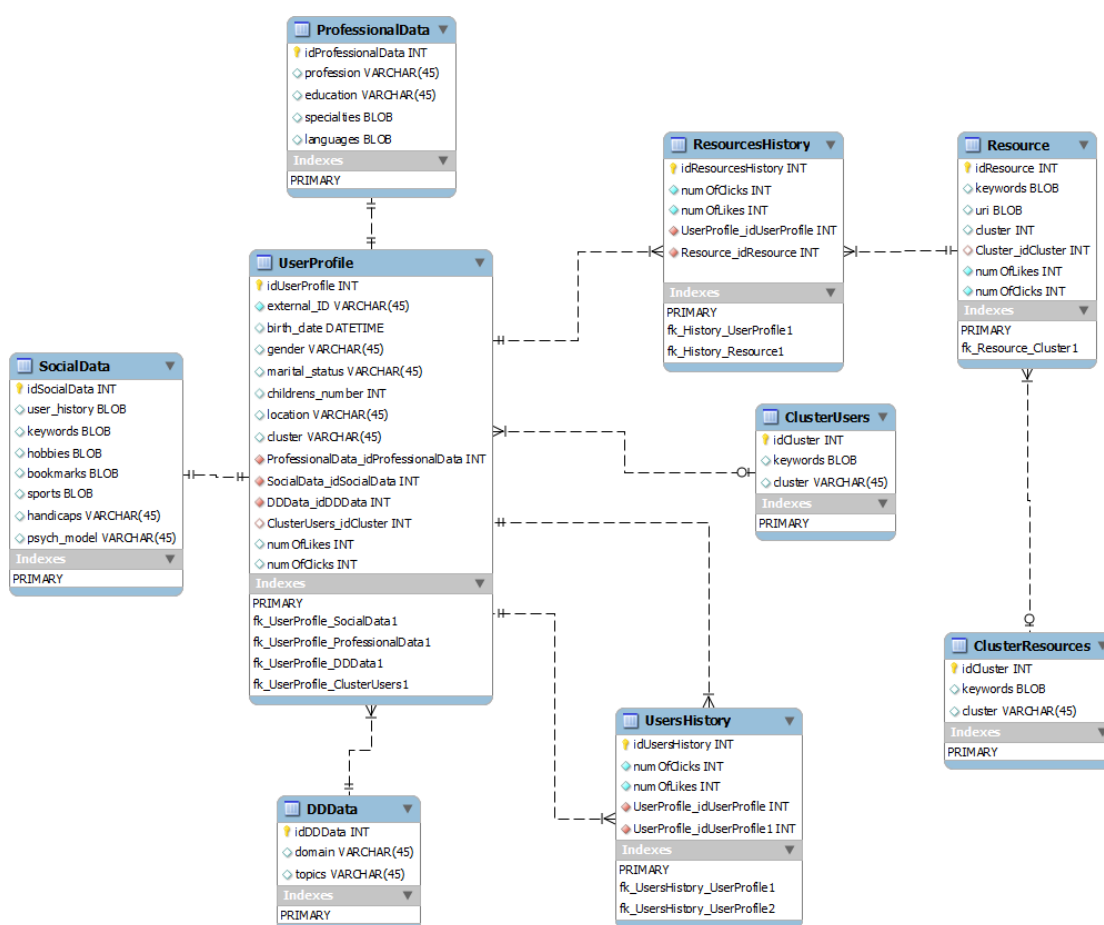


Figura 9 - Modelo da base de dados do MRI

O modelo apresentado está centrado na tabela do perfil do utilizador (*UserProfile*). O perfil do utilizador mantém relações com as tabelas de dados profissionais (*ProfessionalData*), dados sociais (*SocialData*) e a tabela que contém a informação que depende do domínio (*DDDData*). Estas relações permitem relacionar a informação do modelo de utilizador.

A tabela dos recursos (*Resource*) encontra-se relacionada com a tabela de histórico dos recursos (*ResourcesHistory*) que por sua vez mantém uma relação com a tabela do perfil de utilizador. Estas relações permitem registar que recursos foram utilizados ou classificados por cada utilizador.

Alguma da informação existente na base de dados não é usada, na versão atual do protótipo, para o processo de recomendação. No entanto, esta informação está a ser estruturada de forma à inserção da sua utilização no futuro ser agilizada, assim apesar de não ser usada pelo MRI nas recomendações produzidas pode ser utilizada com a finalidade de estudar e explorar diferentes abordagens.

4.4 Ferramentas utilizadas

A tabela 6 detalha as principais tecnologias utilizadas para o desenvolvimento e implementação do MRI.

Tabela 6 - Principais tecnologias utilizadas no desenvolvimento do MRI

| Tipo de tecnologias | Tecnologia utilizada |
|--------------------------------|-------------------------|
| Sistema Operativo | Plataforma Independente |
| Base de Dados | MySQL |
| Servidor Web | Apache Tomcat |
| Linguagem de programação | Java |
| <i>Framework</i> | Struts2 |
| Serviços Web | REST |
| <i>Stemming Framework</i> | PTStemmer |
| <i>Clustering Framework</i> | Implementado |
| Algoritmo de <i>Clustering</i> | DBSCAN |
| Algoritmo de Recomendação | Algoritmo Híbrido |
| Formato do Resultado | xHtml, XML, JSON |

Para o desenvolvimento desde projeto a escolha recaiu na utilização da ferramenta de desenvolvimento *eclipse* como ferramenta principal. A utilização desta ferramenta deve-se ao facto de ser uma das ferramentas de desenvolvimento *Java*, de livre utilização, mais aceite pela comunidade. O servidor web utilizado para disponibilizar o serviço é o *Tomcat*. O *Tomcat* é um servidor web *Java* e é um *software* de livre utilização.

O SGBD (Sistema Gestor de Base de Dados) utilizado para a implementação da base de dados do MRI é o *MySQL* [MySQL, 2012]. Esta escolha deve-se ao fato de ser uma plataforma de uso livre, que oferece as funcionalidades necessárias para suportar o sistema desenvolvido. De forma a gerir a base de dados foi utilizada o *MySQL Workbench* que é uma ferramenta de gestão que permite desenhar a base de dados, desenvolver as queries *SQL* necessárias e efetuar a administração da base de dados.

Para gestão e apoio ao desenvolvimento do código do projeto foi utilizada a ferramenta *git* [Git, 2012]. O *git* é uma ferramenta de controlo de versões usada para gestão de versões, esta ferramenta é utilizada em diferentes tipos de projetos para gestão e manutenção do código desenvolvido. Sendo uma ferramenta de acesso livre e uso gratuito a sua utilização é uma mais-valia no desenvolvimento de *software*.

Neste projeto a versão *standard* do *Visual Studio* foi utilizada para desenvolver pequenos programas clientes na linguagem de programação *C#*, de forma a testar os serviços web disponibilizados pelo MRI. Este projeto foi desenvolvido em torno de serviços de forma a facilitar a integração com outras tecnologias e plataformas. Assim sendo, testar os serviços numa das plataformas mais usadas no momento é um passo importante de validação.

O MRI foi desenvolvido tirando partido das mais variadas tecnologias livres existentes no mercado. Assim sendo, os requisitos de *software* necessários para o correto funcionamento do sistema são:

- *Java*: versão 1.7.0_01
- *MySQL*: versão 5.1.49-3
- *Apache Tomcat*: versão 7.0.25

O sistema desenvolvido é independente do sistema operativo. Assim sendo, qualquer sistema que suporte corretamente o *software* mencionado nos requisitos por ser utilizado.

4.5 Serviços externos

A tabela 7 descreve todos os serviços web disponibilizados pela solução MRI. Na primeira coluna é identificado o serviço, a segunda descreve os métodos que podem ser utilizados e a terceira coluna contém uma breve descrição do serviço.

Tabela 7 - Serviços web

| Serviço | Método | Descrição |
|------------------------------------|------------------------|---|
| UModel | GET, PUT, DELETE, POST | Este serviço gere a informação do modelo de utilizador. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |
| Resource | GET, PUT, DELETE, POST | Este serviço gere a informação dos recursos relevante para o MRI. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |
| Non-Formal Resource Recommendation | GET, POST | Este serviço permite obter uma lista de recursos recomendados no contexto não formal de aprendizagem. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |
| Formal Resource Recommendation | GET, POST | Este serviço permite obter uma lista de recursos recomendados no contexto formal de aprendizagem. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |
| Non Formal User Recommendation | GET, POST | Este serviço permite obter uma lista de utilizadores recomendados no contexto não formal de aprendizagem. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |
| Formal User Recommendation | GET, POST | Este serviço permite obter uma lista de utilizadores recomendados no contexto formal de aprendizagem. <i>Media types:</i> application/x-www-form-urlencoded, application/xml, application/json |

4.5.1 URIs dos serviços web

É possível usar o mesmo serviço para obter o resultado em diferentes formatos de dados como JSON (*JavaScript Object Notation*) ou XML (*eXtensible Markup Language*). No entanto, para que isso aconteça é necessário utilizar uma pequena variação no URI (*Uniform Resource Identifier*) do serviço web. Ao utilizar o URI normal os dados retornados serão do tipo XHTML (*eXtensible Hypertext Markup Language*). De forma a obter dados do tipo JSON é necessário adicionar a extensão `‘.json’` ao URI do serviço. De modo idêntico, para a obter os dados no tipo XML é necessário adicionar a extensão `‘.xml’`.

A tabela 8 detalha os URIs do serviço web dos utilizadores, sendo também possível identificar os formatos de dados que o serviço suporta bem como o método que é necessário utilizar.

Tabela 8 - URIs dos serviços web dos utilizadores

| Serviço | URI | Tipo Retornado | Método | Descrição |
|---------|---|----------------------|--------|--|
| UModel | ../mri/umodel ../mri/umodel.json ../mri/umodel.xml | xhtml json xml | GET | Utilize este URI para obter todos os utilizadores. |
| | | | POST | Utilize este URI para criar um novo utilizador. |
| | ../mri/umodel/{id} ../mri/umodel/{id}.json ../mri/umodel/{id}.xml | xhtml json xml | GET | Utilize este URI para obter um utilizador com id = {id}. |
| | | | PUT | Utilize este URI para editar um utilizador com id = {id}. |
| | ../mri/umodel/{id} ?_method=DELETE | Code: 200 | POST | Utilize este URI para apagar um utilizador com id = {id}. |
| | ../mri/umodel/{id}/like | Code: 200 | POST | Utilize este URI para gostar do utilizador com id = {id}. |
| | ../mri/umodel/{id}/unlike | Code: 200 | POST | Utilize este URI para desgostar um utilizador com id = {id}. |
| | ../mri/umodel/{id}/click | Code: 200 | POST | Utilize este URI para clicar um utilizador com id = {id}. |

A tabela 9 mostra os serviços web que o MRI disponibiliza para a gestão da informação dos recursos, sendo também possível identificar qual é os formatos de dados retornados por cada serviço e qual é o método que é necessário utilizar.

Tabela 9 - URIs dos serviços web dos recursos

| Serviço | URI | Tipo Retornado | Método | Descrição |
|----------|--|----------------------|--------|---|
| Resource | ../mri/resource | xhtml json xml | GET | Utilize este URI para obter todos os dados dos recursos. |
| | | | POST | Utilize este URI para criar um recurso. |
| | ../mri/resource/{id} ../mri/resource/{id} | xhtml json xml | GET | Utilize este URI para obter um recurso com id = {id}. |
| | | | PUT | Utilize este URI para editar um recurso com id = {id}. |
| | ../mri/resource/{id}?_method=DELETE | Code: 200 | POST | Utilize este URI para apagar um recurso com id = {id}. |
| | ../mri/resource/{id}/like | Code: 200 | POST | Utilize este URI para gostar um recurso com id = {id}. |
| | ../mri/resource/{id}/unlike | Code: 200 | POST | Utilize este URI para desgostar um recurso com id = {id}. |
| | ../mri/resource/{id}/click | Code: 200 | POST | Utilize este URI para clicar um recurso com id = {id}. |

A tabela 10 detalha os URIs dos serviço web de recomendações, sendo possível identificar os formatos de dados retornados por cada serviço e qual o método que é necessário utilizar.

Tabela 10 - URIs dos serviços web de recomendações

| Serviço | URI | Tipo Retornado | Método | Descrição |
|------------------------------------|---|----------------------|--------|--|
| Non-Formal Resource Recommendation | ../mri/recommend ../mri/recommend.json ../mri/recommend.xml | xhtml json xml | POST | Utilize este URI para obter uma lista de sugestões de recursos (não formal). |
| Formal Resource Recommendation | ../mri/recommendformal ../mri/recommendformal.json ../mri/recommendformal.xml | xhtml json xml | POST | Utilize este URI para obter uma lista de sugestões de recursos (formal). |
| Non Formal User Recommendation | ../mri/recommenduser ../mri/recommenduser.json ../mri/recommenduser.xml | xhtml json xml | POST | Utilize este URI para obter uma lista de sugestões de utilizadores (não formal). |
| Formal User Recommendation | ../mri/recommendformaluser ../mri/recommendformaluser.json ../mri/recommendformaluser.xml | xhtml json xml | POST | Utilize este URI para obter uma lista de sugestões de utilizadores (formal). |

4.5.2 Parâmetros dos serviços web

Para utilizar os serviços web do MRI é necessário passar os parâmetros corretos no uso dos serviços. Os serviços do MRI podem ser invocados utilizando diferentes formatos nos parâmetros. É possível invocar os serviços utilizando parâmetros HTTP normais ou utilizar uma estrutura de dados pré-definida em JSON ou XML.

A tabela 11 detalha com uma breve descrição os parâmetros necessários para a utilização do serviço web dos utilizadores.

Tabela 11 - Parâmetros dos serviços web dos utilizadores

| Serviços | Parâmetros | Descrição |
|----------------|---|--|
| UModel | id | Identifica o utilizador. <i>Type: String</i> |
| | external_ID | Identifica o utilizador no LCMS. Utilizado quando é criado/atualizado/apagado um modelo de utilizador. <i>Type: String</i> |
| | birth_date | Data de nascimento do utilizador. <i>Type: String</i> |
| | gender | Gênero do utilizador. <i>Type: String</i> |
| | marital_status | Estado civil do utilizador. <i>Type: String</i> |
| | childrens_number | Número de filhos do utilizador. <i>Type: String</i> |
| | location | Localização do utilizador. <i>Type: String</i> |
| | profession | Profissão do utilizador. <i>Type: String</i> |
| | education | Nível de educação do utilizador. <i>Type: String</i> |
| | specialties | Especialidades do utilizador. <i>Type: String</i> |
| | languages | Linguagens que o utilizador conhece. <i>Type: String</i> |
| | user_history | Histórico do utilizador. <i>Type: String</i> |
| | keywords | Palavras-chave que descrevem o utilizador. <i>Type: String</i> |
| | hobbies | Passatempos do utilizador. <i>Type: String</i> |
| | bookmarks | Favoritos do utilizador. <i>Type: String</i> |
| | sports | Desporto que o utilizador demonstra interesse. <i>Type: String</i> |
| | handicaps | Limitações/deficiências do utilizador. <i>Type: String</i> |
| | psych_model | Modelo psicológico do utilizador. <i>Type: String</i> |
| | courseName | Nome do curso no qual o utilizador está inscrito. <i>Type: String</i> |
| | subject | Assunto do curso. <i>Type: String</i> |
| | topics | Tópicos do curso. <i>Type: String</i> |
| | duration | Duração do curso. <i>Type: String</i> |
| | sequence | Sequência do curso. <i>Type: String</i> |
| | userKnowledge | Nível de conhecimento que o utilizador tem sobre este curso. <i>Type: String</i> |
| | history | Histórico do utilizador relacionado com o curso. <i>Type: String</i> |
| | timeSpend | Tempo dispendido pelo utilizador neste curso. <i>Type: String</i> |
| | evaluation | Avalização do utilizador. <i>Type: String</i> |
| performance | Performance do utilizador. <i>Type: String</i> | |
| feedbackUserId | Usado apenas para identificação do utilizador que está a dar <i>feedback</i> . <i>Type: String</i> | |

A estrutura pré-definida, em XML e JSON, relativa à tabela 11 está descrita na parte de estruturas dos anexos. Esta estrutura pode ser utilizada como parâmetro, para passar a informação do utilizador, na utilização do serviço.

Os serviços relacionados com os recursos não efetuam a gestão dos recursos, apenas gerem a informação relevante ao MRI sobre esses recursos. A tabela 12 detalha os parâmetros necessários para a utilização dos serviços dos recursos.

Tabela 12 - Parâmetros dos serviços dos recursos

| Serviços | Parâmetros | Descrição |
|----------|----------------|---|
| Resource | id | Identifica os dados de um recurso. <i>Type: String</i> |
| | resourceId | Identifica o recurso no LCMS. <i>Type: String</i> |
| | keywords | Palavras-chave que descrevem o recurso. <i>Type: String</i> |
| | uri | Hiperligação para o recurso. <i>Type: String</i> |
| | xmlMetadata | Estrutura xml dos meta-dados da norma SCORM. Pode ser usada para criar ou atualizar a informação do recurso em vez dos parâmetros 'keywords' e 'uri'. <i>Type: String</i> |
| | feedbackUserId | Usado apenas para identificação do utilizador que está a dar <i>feedback</i> . <i>Type: String</i> |

A estrutura pré-definida, em XML e JSON, relativa à tabela 12 está descrita na parte de estruturas dos anexos. Esta estrutura pode ser utilizada como parâmetro, para passar a informação do recurso, na utilização do serviço.

A tabela 13 descreve os parâmetros necessários para a utilização dos serviços de recomendação.

Tabela 13 - Parâmetros dos serviços web de recomendação

| Serviço | Parâmetros | Descrição |
|------------------------------------|------------|---|
| Non-Formal Resource Recommendation | userId | Identifica o utilizador. <i>Type: String</i> |
| | query | Usada para passar as palavras-chave da pesquisa. <i>Type: String</i> |
| Formal Resource Recommendation | userId | Identifica o utilizador. <i>Type: String</i> |
| Non-Formal User Recommendation | userId | Identifica o utilizador. <i>Type: String</i> |
| | query | Usada para passar as palavras-chave da pesquisa. <i>Type: String</i> |
| Formal User Recommendation | userId | Identifica o utilizador. <i>Type: String</i> |

4.6 Serviços internos

Foram criados dois serviços internos, um que permite iniciar o processo de criação de *clusters* dos recursos e outro que possibilita iniciar o processo de criação de *clusters* dos utilizadores. Para utilizar estes serviços foi criada uma aplicação *Java* que deve ser executada com recurso a uma tarefa agendada pelo sistema operativo. Ao criar estes dois serviços foi possível remover o processo de agendamento da execução periódica do procedimento de criação de *clusters* do módulo desenvolvido.

Esta opção permite assim, descomplexar o sistema implementado aumentando a flexibilidade do mesmo. O procedimento de *clustering* deve ser efetuado periodicamente para que novas atualizações sejam consideradas pelo MRI.

4.7 Exemplo de utilização

Neste subcapítulo é apresentado um exemplo dos resultados gerados pelo MRI. Primeiro são apresentados os *clusters* criados bem com a informação que os elementos de cada *cluster* contém. De seguida, é apresentado um exemplo de uma *query* e a lista de sugestões produzidas para o utilizador.

A tabela 14 apresenta um exemplo simplificado de *clusters* de recursos criados. Os dados apresentados foram agrupados pelo processo de *clustering*. Os *clusters* apresentados apresentam uma lista de conjuntos de palavras-chave onde cada conjunto representa um recurso.

Tabela 14 - Exemplo de *clusters* criados

| <i>Cluster 1</i> | <i>Cluster 2</i> | <i>Cluster 3</i> |
|--------------------------------------|----------------------------|----------------------------|
| programaca algoritm inici perl | desenvolv avanc ruby | sistem arquetet java |
| programaca algoritm inici javascript | desenvolv avanc java | sistem arquetet c# |
| programaca algoritm inici html | desenvolv avanc c# | sistem arquetet c++ |
| programaca algoritm inici c# | desenvolv avanc c++ | sistem arquetet php |
| programaca algoritm inici python | desenvolv avanc perl | sistem arquetet javascript |
| programaca algoritm inici erlang | desenvolv avanc javascript | sistem arquetet html |
| programaca algoritm inici prolog | desenvolv avanc html | sistem arquetet perl |
| programaca algoritm inici sql | desenvolv avanc php | sistem arquetet python |
| programaca algoritm inici ada | | sistem arquetet erlang |
| | | sistem arquetet prolog |
| | | sistem arquetet sql |
| | | sistem arquetet ada |
| | | sistem arquetet ruby |

Utilizando o serviço de recomendação não formal para o utilizador id = 1:

Query = 'programação ada'

A *query* é processado pelo procedimento de *stemming* sendo transformada em:

Query = 'programaca ada'

Para o cálculo da distância entre a *query* e o *cluster* primeiro são identificadas as palavras que representam o *cluster*. As palavras utilizadas para representar o *cluster* são retiradas das descrições dos elementos que compõem o *cluster*. A figura 10 apresenta um diagrama onde é possível observar o resultado do cálculo efetuado.

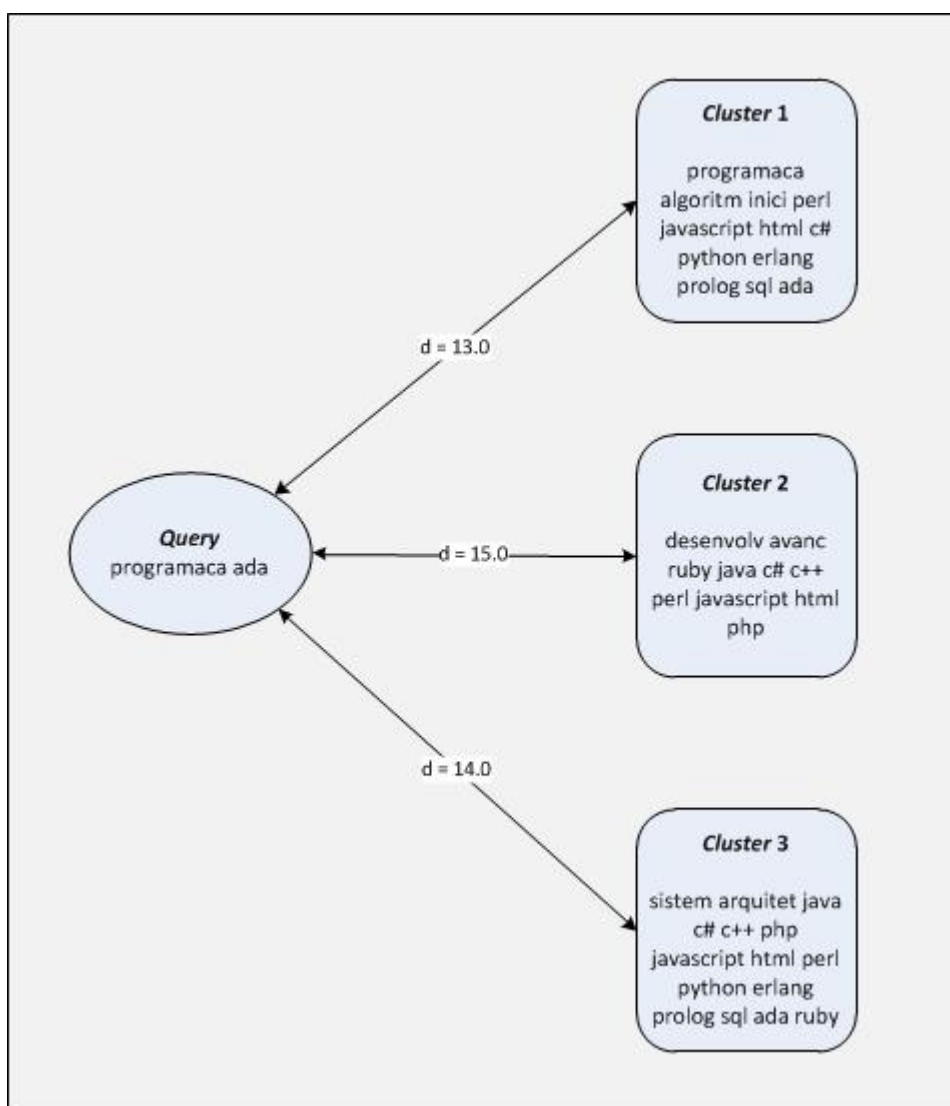


Figura 10 - Resultados do cálculo da distância entre a *query* e os diferentes *clusters*

A lista é ordenada tendo em consideração:

- 1º - A similaridade da *query* com as palavras-chave que representam cada recurso;
- 2º - A avaliação efetuada por utilizadores que pertencem ao mesmo *cluster* do indivíduo;
- 3º - Quantidade de visualizações dos recursos.

A tabela 15 apresenta uma lista dos resultados obtidos do exemplo apresentado. A primeira coluna da tabela identifica o *cluster* utilizado. A segunda coluna apresenta um URI que identifica o recurso. De seguida são apresentados a descrição do recurso número de gostos e cliques.

Tabela 15 - Tabela de resultados do exemplo MRI

| <i>Cluster</i> | URI | Descrição do recurso | Gostos | Cliques |
|----------------|--------|--------------------------------------|--------|---------|
| 1 | uri_29 | programaca algoritm inici ada | 0 | 1 |
| 1 | uri_21 | programaca algoritm inici perl | 5 | 3 |
| 1 | uri_23 | programaca algoritm inici html | 4 | 2 |
| 1 | uri_22 | programaca algoritm inici javascript | 4 | 1 |
| 1 | uri_27 | programaca algoritm inici prolog | 2 | 1 |
| 1 | uri_28 | programaca algoritm inici sql | 1 | 3 |
| 1 | uri_24 | programaca algoritm inici c# | 1 | 1 |
| 1 | uri_25 | programaca algoritm inici python | 0 | 4 |
| 1 | uri_26 | programaca algoritm inici erlang | 0 | 0 |

De toda a informação apresentada na tabela 15 na realidade apenas é necessário devolver ao utilizador o URI. No entanto, de forma a ser possível compreender os resultados e forma como as recomendações são produzidas optou-se por mostrar estes dados.

4.8 Resumo

Neste capítulo foram apresentadas as técnicas utilizadas no desenvolvimento do sistema. A opção de utilização do protocolo REST oferece maior facilidade de integração a outros sistemas da plataforma. A arquitetura do sistema é apresentada de forma a demonstrar o fluxo desejado dos diferentes procedimentos do MRI. De forma a completar a arquitetura do sistema é mostrado o modelo da base de dados. Neste modelo é possível identificar as relações que existem entre as diferentes tabelas.

No subcapítulo 4.4 são identificadas as ferramentas e tecnologias que de foram de alguma forma relevantes ao longo do desenvolvimento do projeto. Por fim, são descritos detalhadamente os

serviços disponibilizados bem como os parâmetros necessários para uma correta utilização do sistema.

Sendo o MRI um sistema desenvolvido sobre a tecnologia *struts2* em *Java*, este tira partido de tecnologias livres que diminuem o custo não só de desenvolvimento do sistema, mas também do custo de implementação, manutenção e produção. Sendo uma arquitetura orientada a serviços aumenta a flexibilidade e independência do sistema. Foram descritos os serviços desenvolvidos onde é possível ver as diferentes possibilidades oferecidas. Na parte final deste capítulo foi apresentado um exemplo de utilização e a lista de resultados obtida onde é possível observar que a combinação de todas as tecnologias utilizadas permitem obter uma lista de sugestões ordenada de forma que a primeira sugestão da lista seja a mais recomendada para um determinado utilizador.

5 Conclusões

O sistema de recomendação híbrido desenvolvido é composto por uma sequência de módulos que implementam diversas técnicas. A aplicação dessas técnicas de forma isolada na resolução de outro tipo de problemas já foi efetuada com sucesso [Lipczak and Evangelos, 2010]. No entanto, o objetivo proposto neste projeto foi o de mostrar que essas técnicas podem ser usadas em conjunto de forma a aumentar a eficácia do comportamento de sistemas de recomendação aplicados ao ensino e formação. Ao combinar os diferentes métodos já testados é possível obter uma solução capaz de produzir sugestões adequadas ao utilizador.

5.1 Conclusões e análise crítica

Foi considerado o uso de ontologias de forma a aumentar o conjunto de palavras-chave, de forma a aumentar o número de resultados para pesquisas com um número de palavras-chaves reduzido [Andreou, 2005] [Sally, El-Desouky and Sherief, 2009] [Siddharth and Hsinchun, 2007]. No entanto o uso de ontologias demonstrou-se não ser uma solução viável, isto porque seria necessário construir uma nova ontologia para cada domínio.

Técnicas de filtragem baseadas em colaboração e conteúdos são frequentemente utilizadas em plataformas sociais e de partilha e surgem como técnicas capazes de melhorar recomendações e personalizar resultados [Jaschke, Marinho, Hotho, Schmidt-Thieme and Stumm, 2007] [Tzone, Tsai, Lee and Chiu, 2007].

A utilização de um procedimento de *stemming* elimina ruído e permite aos restantes algoritmos serem mais eficientes e atingirem melhores resultados. Existem palavras que são idênticas mas têm significados diferentes. Por exemplo a palavra '*java*', é uma linguagem de programação, uma ilha da Indonésia e um tipo de café. Nestas situações, o processo de *stemming* não consegue identificar o significado de acordo com o contexto em causa. No

entanto, no decorrer do projeto concluiu-se que não seria prioritário considerar a semântica das palavras no processo de comparação porque o MRI está inserido num determinado ambiente educacional.

Ao longo do projeto o processo de similaridade foi ganhando importância. Calcular a similaridade entre palavras da melhor forma possível é provavelmente o ponto mais importante deste projeto. A qualidade dos *clusters* criados é comprometida caso o algoritmo de similaridade não seja capaz de criar corretamente uma relação de distância corretamente. Conclui-se assim, que para o sistema ser capaz de sugerir recomendações relevantes para o utilizador, é necessário este possuir um processo de similaridade eficaz.

Do resultado obtido no exemplo de utilização dado no capítulo 4 é possível observar que o MRI é capaz de identificar corretamente o *cluster* de recursos mais indicado. Após essa identificação o sistema consegue ordenar as sugestões dadas de forma personalizada. Esta personalização é conseguida considerando as avaliações efetuadas pelos utilizadores que pertencem ao mesmo *cluster* do utilizador que fez o pedido. Assim, os resultados apresentados pelo MRI num ambiente de teste vão de encontro aquilo que é esperado, cumprindo assim os objetivos propostos para este trabalho.

5.2 Trabalho futuro

Este trabalho encontra-se em fase de prototipagem, assim sendo, é importante continuar o melhoramento e desenvolvimento do sistema bem como efetuar testes com dados reais. Sendo possível desta forma estudar o comportamento do sistema num ambiente real, após a integração na plataforma 3.0. A utilização do sistema num ambiente real irá permitir identificar situações que careçam de correção.

A utilização de um sistema híbrido escalável deixa em aberto a possibilidade para atualizar o sistema com novas técnicas que possam emergir. Por exemplo, existe a possibilidade de introduzir no futuro uma opção para aumentar a importância de novos recursos. Com esta medida pretende-se dar relevância a recursos novos no sistema, permitindo que estes seja sugeridos mais rapidamente aos utilizadores, possibilitando assim a sua classificação. Outras funções analisadas que podem vir a ser incluídas no projeto é o caso do ‘*Search Spelling*’, que é uma ferramenta que permite auto completar palavras em tempo real. Com estes exemplos pretende-se demonstrar que existem diferentes abordagens alternativas que podem ser estudadas/desenvolvidas no futuro de forma a melhorar ou alterar o comportamento do MRI e o tipo de recomendações produzidas.

A ideia do uso de ontologias foi abandonada neste trabalho, no entanto, no futuro poderão ser utilizados serviços que tiram partido de ontologias de forma a melhorar as recomendações. Neste momento, está a ser analisada a integração de um serviço disponibilizado por um parceiro do consórcio, a MAISIS, que tira partido de ontologias e serviria para expandir a informação inserida no MRI. Este serviço oferece também outras funcionalidades, como a

possibilidade de extrair informação automaticamente de um *curriculum vitae*, o que poderá ser utilizado para instanciar os modelos dos utilizadores da plataforma.

A ideia da utilização dos valores obtidos, através do algoritmo de distância de *Levenshtein*, foi abandonada porque a semântica das palavras no processo de comparação não foi considerada. Assim, esses valores não estão a ser utilizados de momento, no entanto, poderão vir a ser incluídos no futuro, no caso de se adotar a utilização de uma ontologia.

Apesar de existirem diversas técnicas que teoricamente poderão melhorar os resultados do MRI, é necessário ponderar a utilização de serviços externos. Caso a solução crie uma dependência destes serviços, esta deixa de ser independente. A adoção de serviços deverá sempre ser considerada com o objetivo de melhorar o comportamento do MRI, sem que este se torne dependente destes serviços externos.

Para finalizar, um dos pontos que pode ser melhorado no futuro é o aspeto colaborativo do sistema. Para isso é necessário melhorar o processo de apreciação das recomendações produzidas pelos utilizadores. Outra funcionalidade que permitiria melhorar o sistema seria, dotar o sistema com a capacidade de inferir o comportamento do utilizador na plataforma, utilizando reconhecimento de padrões de comportamento. Para este efeito as redes *bayesianas* têm sido usadas com sucesso [Janssens D, Wets G., Brijs T., Vanhoof k. and Timmermans H, 2003].

Referências

- [Andreou, 2005] Andreou A, Ontologies and Query expansion, School of Informatics University of Edinburgh, 2005.
- [Andritsos, 2002] Andritsos P., Data Clustering Techniques Qualifying Oral Examination Paper, University of Toronto Department of Computer Science periklis@cs.toronto.edu March 11, 2002.
- [Apache Mahout, 2012] Apache Mahout, <http://mahout.apache.org/>, [ultimo acesso, Outubro 2012].
- [Bagatan, Boja and Cristian, 2011] Batagan L., Boja C., Cristian I., Intelligent Educational Systems, Support for an Education Cluster, Economics Informatics Department, Academy of Economic Studies, Romana Square N°6, Bucharest, Romania. European Computing Conference, 2011.
- [Berka and Salzburg, 2004] Berka T. and Salzburg M. P., Designing Recommender Systems for Tourism, Research Jakob-Haringer Str. 5/III 5020 Salzburg Austria {tberka, mploessnig@salzburgresearch.at in Proceedings of the 11th International Conference on Information Technology in Travel and Tourism (ENTER), 2004.
- [Bilenko and Mooney, 2003] Bilenko M, Mooney R J, Employing Trainable String Similarity Metrics for Information Integration, Department of Computer Sciences, University of Texas at Austin, 2003.
- [Bontchev and Vassileva, 2011] Bontchev B, Vassileva D, Learning Objects Types Dependability on Styles of Learning, Department of Software Engineering, Sofia University, Faculty of Mathematics and Informatics, Bulgaria. Recent Researches in Educational Technologies, Corfu, Greece, 227-234, 2011.
- [Cail, He, Li, Ma and Wen, 2004] Cail D, He X, Li Z, Ma W, Wen J, Hierarchical Clustering of www Image Search Results Using Visual, Textual and Link Information, 2004.
- [Cattuto, Benz, Hotho and Stumme, 2008] Cattuto C, Benz D, Hotho A, Stumme G, Semantic Analysis of Tag Similarity Measures in Collaborative Tagging Systems, 2008.
- [Coelho, Figueiredo and Martins, 2009] Coelho B, Figueiredo A, Martins C, Tours Planning Decision Support, GECAD – Knowledge Engineering and Decision Support Group, Computer Science Department, Institute of Engineering – Polytechnic of Porto, Porto, Portugal, 2009.
- [Coelho, Martins and Almeida, 2009] Coelho B, Martins C, Almeida A, Adaptive Tourism Modeling and Socialization System, GECAD – Knowledge Engineering and Decision Support Group, Computer Science Department, Institute of Engineering – Polytechnic of Porto, Porto, Portugal, 2009.
- [Coelho, Martins, Figueiredo, 2009] Coelho B, Martins C, Figueiredo A, Hybrid Architecture for Tourist Modeling and Recommender System, GECAD – Knowledge Engineering and Decision Support Group, Computer Science Department, Institute of Engineering – Polytechnic of Porto, Porto, Portugal.
- [DBSCAN, 2012] DBSCAN, <http://en.wikipedia.org/wiki/DBSCAN>, [ultimo acesso, Outubro 2012].
- [Duine Framework, 2012] Duine Framework, <http://sourceforge.net/projects/duine/>, [ultimo acesso, Outubro 2012].
- [Easyrec, 2012] Easyrec, <http://easyrec.org/>, [ultimo acesso, Outubro 2012].
- [Elki, 2012] Elki, <http://elki.dbs.ifi.lmu.de/>, [ultimo acesso, Outubro 2012].
- [Fort, Letremy and Cottrell, 2002] Fort J, Letremy P., Cottrell M, Advantages and drawbacks of the Batch Kohonen algorithm, Institut Elie Cartan et SAMOS-MATISSE Université Nancy 1, F-54506 Vandoeuvre-Lès-Nancy, France fortjc@iecn.u-nancy.fr 2

- Université Paris I, SAMOS-MATISSE, UMR CNRS 8595 90 rue de Tolbiac, F-75634 Paris Cedex 13, France pley, cottrell@univ-paris1.fr, 2002.
- [Freedman, 2000] Freedman R, What is an Intelligent Tutoring System?, Northern Illinois University. Fall 2000.
- [Fung, 2001] Fung G, A Comprehensive Overview of Basic Clustering Algorithms, June 22, 2001.
- [Gamboa and Fred, 2003] Gamboa H, Fred A, Designing Intelligent Tutoring Systems: a bayesian approach, 2003.
- [Gemmis and Semeraro, 2011] Gemmis Marco de, and Semeraro Giovanni, Content-based Recommender Systems: State of the Art and Trends Pasquale Lops, Trier University, 2011.
- [Git, 2012] Git, <http://git-scm.com/>, [ultimo acesso, Outubro 2012].
- [Goodkovsky, 2006] Goodkovsky, V.A. "e-Learning Powered By Intelligent Tutoring ". AACE International Conference ELEARN 2006, Honolulu, Hawaii <http://www.goodkovsky.org/eLPIT88-original.htm>, October 13-17, 2006.
- [IEEE, 2002] IEEE, Learning Technology Standards Committee, WG12: Learning Object Metadata, <http://ltsc.ieee.org/wg12/>, December 2002.
- [Janssens D, Wets G., Brijs T., Vanhoof k. and Timmermans H, 2003] Janssens D, Wets G., Brijs T., Vanhoof k. and Timmermans H., Identifying Behavioral Principles Underlying Activity Patterns by Means of Bayesian Networks University of Limburg Data Analysis and Modeling Group Faculty of Applied Economic Sciences University Campus, 3590 Diepenbeek, Belgium
Eindhoven University of Technology Urban Planning Group, PO Box 513, 5600 MB Eindhoven, The Netherlands, 2003
- [Ji-Rong and Hong-Jiang, 2003] Ji-Rong Wen, Hong-Jiang Zhang, Query Clustering in the Web Context, Microsoft Research Asia 3F, Beijing Sigma Center No. 49, Zhichun Road, Haidian District Beijing, P.R. China 100080, December 2003.
- [Ji-Rong, Jian-Yun and Hong-Jiang, 2002] Ji-Rong Wen, Jian-Yun Nie, Hong-Jiang Zhang, Query Clustering Using User Logs, January 2002.
- [Ji-Rong, Jian-Yun and Hong-Jiang, 2001] Ji-Rong Wen, Jian-Yun Nie, Hong-Jiang Zhang, Clustering User Queries of a Search Engine, May 2001.
- [Jaschke, Marinho, Hotho, Schmidt-Thieme and Stumm, 2007] Jaschke R, Marinho L, Hotho A, Schmidt-Thieme L, Stumm G, Tag Recommendations in Folksonomies, 2007.
- [Knime, 2012] Knime, <http://www.knime.org/>, [ultimo acesso, Outubro 2012].
- [Kriegel, Sander and Xu, 1998] Kriegel H., Sander J. and Xu X., A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise Martin Ester, Institute for Computer Science, University of Munich Oettingenstr. 67, D-80538 München, Germany {ester | kriegel | sander | xwxu@informatik.uni-muenchen.de, 1998.
- [Lane, 2006] Lane H C, Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning, Institute for Creative Technologies University of Southern California. July 27, 2006.
- [Lee and Chun, 2008] Lee S O K, Chun A H, Automatic Tag Recommendation for Web 2.0 Blogosphere by Extracting Keywords from Similar Blogs, Department of Computer Science City University of Hong Kong. 7th WSEAS (ACACOS '08) Hangzhou, China, April 6-8, 2008.
- [LensKit, 2012] LensKit, <http://lenskit.grouplens.org/>, [ultimo acesso, Outubro 2012].
- [Limam, Coquil, Kosch and Brunie, 2010] Lipczak M, Milios Evangelos, Faculty of Computer Science, Dalhousie University Halifax, Canada, 2010.
- [Lingpipe, 2012] Lingpipe, <http://alias-i.com/lingpipe/>, [ultimo acesso, Outubro 2012].
- [Marinai, 2007] Marinai S, SOM clustering for text retrieval and classification with examples on Indian scripts, Dipartimento di Sistem in Informatica - Universit`a di Firenze Via S.Marta, 3 - 50139 Firenze – Italy, 2007.
- [Markines, Cattuto, Menczer F, Benz D, Hotho A, Stumme G, Evaluating

- Menczer, Benz, Hotho and Stumme, 2009] Similarity Measures for Emergent Semantics of Social Tagging, 2009.
- [Martins, Faria, Carvalho and Carrapatoso, 2008] Martins C, Faria L, Carvalho C V, Carrapatoso E, User modeling in adaptive hypermedia educational systems, *Educational Technology & Society*, 11(1), pp. 194-207, 2008.
- [MySQL, 2012] MySQL, <http://www.mysql.com/>, [ultimo acesso, Outubro 2012].
- [Ong and Ramachandran, 2003] Ong J, Ramachandran S, Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI, Stottler Henke Associates, Inc. San Mateo CA, 2003.
- [Papagelis, Plexousakis and Kutsuras, 2005] Papagelis M, Plexousakis D, Kutsuras T, Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences, Institute of Computer Science, Foundation for Research and Technology – Hellas, Department of Computer Science, University of Crete, Heraklion, Greece, 2005.
- [Pazzani and Billsus, 2007] Pazzani M J, Billsus D, Content-Based Recommendation Systems, 2007.
- [Rane, 2005] Rane A K, Intelligent Tutoring System For Marathi, Karnataka State Open University Mysore, December 2005.
- [Rapidminer, 2012] Rapidminer, <http://rapid-i.com/content/view/181/190/>, [ultimo acesso, Outubro 2012].
- [Rehman, 2006] Rehman M, Comparison of Density-based Clustering Algorithms, Lahore College for Women University Lahore, Pakistan
mariam.rehman321@gmail.com Syed Atif Mehdi University of Management and Technology Lahore, Pakistan atif.mehdi@gmail.com, 2006.
- [REST, 2012] REST, <http://www.w3.org/Submission/SA-REST/>, [ultimo acesso, Outubro 2012].
- [Sally, El-Desouky and Sherief, 2009] Sally M. El-Ghamrawy, Ali I. El-Desouky, M. Sherief, Dynamic Ontology Mapping for Communication in Distributed Multi-Agent Intelligent System, 2009.
- [SCORM, 2012] SCORM, http://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model, [ultimo acesso, Outubro 2012].
- [Shani and Gunawardana, 2009] Shani G. and Gunawardana A., Evaluating Recommendation Systems, Microsoft Research, November 2009.
- [Siddharth and Hsinchun, 2007] Siddharth Kaza, Hsinchun Chen, Evaluating ontology mapping techniques: An experiment in public safety information sharing, Department of Management Information Systems, University of Arizona, United States, December 15, 2007.
- [Smith and Alan, 2003] Smith K A, Alan Ng, Web page clustering using a self-organizing map of user navigation patterns, School of Business Systems, Monash University, P.O. Box 63B, Victoria 3800, Australia, 2003.
- [Smirnov, 2008] Smirnov I, Overview of Stemming Algorithms, DePaul University, December 2008.
- [Song, Zhang and Giles, 2008] Song Y, Zhang L, C. Lee Giles, Automatic Tag Recommendation Algorithms for Social Recommender Systems, The Pennsylvania State University, 2008.
- [Struts2, 2012] Struts2, <http://struts.apache.org/2.x/>, [ultimo acesso, Outubro 2012].
- [Suggest, 2012] Suggest, <http://glaros.dtc.umn.edu/gkhome/suggest/overview>, [ultimo acesso, Outubro 2012].
- [Tavares, Faria e Martins, 2012] Tavares B, Faria L, Martins C, Intelligent Recommendation System for e-learning Platforms, ISEP, Porto, Portugal, 14th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems MAMECTIS '12, 2012.
- [Tom Murray, 1999] Tom Murray, Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art, *International Journal of Artificial Intelligence in Education*, 10, 98-129, Computer Science Dept., University of Massachusetts,

- Amherst & School of Cognitive Science, Hampshire College, Amherst, MA
tmurray@cs.umass.edu, www.cs.umass.edu/~tmurray/, 1999.
- [Tsai and Wu, 2009] Tsai C, Wu, C, GF-DBSAN: A New Efficient and Effective Data Clustering Technique for Large Databases, Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung, Taiwan, 9th WSEAS International Conference on Multimedia Systems & Signal Processing, 2009.
- [Tzone, Tsai, Lee and Chiu, 2007] Tzone I Wang, Kun Hua Tsai, Ming Che Lee and Ti Kai Chiu, Personalized Learning Objects Recommendation based on the Semantic-Aware Discovery and the Learner Preference Pattern, National Chung Kung University, Taiwan, 2007.
- [Weka, 2012] Weka, <http://www.cs.waikato.ac.nz/ml/weka/>, [ultimo acesso, Outubro 2012].
- [Willett, 2006] Willett P, The Porter stemming algorithm: then and now. Program: electronic library and information systems, 40 (3). pp. 219-223, 2006.
- [Zanin, Cano, Buldú and Celma, 2008] Zanin M, Cano P, Buldú J M, Celma O, Complex Networks in Recommendation Systems, Universitat Pompeu Fabra, Universidad Rey Juan Carlos, Spain, 2th WSEAS (CEA'08) Acapulco, Mexico, January 25-27, 2008.
- [Zhou, Xu, Li, Josang and Cox, 2012] Zhou X., Xu Y., Li Y., Josang A. and Cox C., The State-of-the-Art in Personalized Recommender Systems for Social Networking, Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia, 2012.

6 Anexos

6.1 Estruturas de dados

Estrutura do utilizador em XML:

```
<rest.services.Umodel>
  <external__ID></external__ID>
  <gender>masculin</gender>
  <marital__status> </marital__status>
  <childrens__number></childrens__number>
  <location> </location>
  <profession> </profession>
  <education> </education>
  <specialties> </specialties>
  <languages> </languages>
  <user__history> </user__history>
  <keywords> </keywords>
  <hobbies> </hobbies>
  <bookmarks> </bookmarks>
  <sports> </sports>
  <handicaps> </handicaps>
  <psych__model> </psych__model>
  <courseName> </courseName>
  <subject> </subject>
  <topics> </topics>
  <duration> </duration>
  <sequence> </sequence>
  <userKnowledge> </userKnowledge>
  <history> </history>
  <timeSpend> </timeSpend>
  <evaluation> </evaluation>
  <performance> </performance>
</rest.services.Umodel>
```

Estrutura do utilizador em JSON:

```
{ "birth_date": "",
  "bookmarks": "",
  "childrens_number": "",
  "courseName": "",
  "duration": "",
  "education": "",
  "evaluation": "",
  "external_ID": "",
  "gender": "",
  "handicaps": "",
  "history": "",
  "hobbies": "",
  "keywords": "",
  "languages": "",
  "location": "",
  "marital_status": "",
  "performance": "",
  "profession": "",
  "psych_model": "",
  "sequence": "",
  "specialties": "",
  "sports": "",
  "subject": "",
  "timeSpend": "",
  "topics": "",
  "userKnowledge": "",
  "user_history": "" }
```

Estrutura do recurso em XML:

```
<rest.services.Resource>
  <resourceId></resourceId>
  <keywords> </keywords>
  <uri></uri>
  <xmlMetadata ></ xmlMetadata >
</rest.services.Resource>
```

Estrutura do recurso em JSON:

```
{ "keywords": "",
  "resourceId": "",
  "uri": "",
  "xmlMetadata": "" }
```

6.2 Exemplos

Exemplo em Java de uma aplicação cliente utilizando um método GET:

```
private static void GetUModel() throws IOException {
    System.out.println("Test REST MRI, GetUModel");
    //Set web service URI
    String uri = "http://hostname:port/mri/umodel/2.xml";
    URL url = new URL(uri);
    // Open http connection
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    //Set web service request method
    connection.setRequestMethod("GET");
    //Set web service request type
    connection.setRequestProperty("Accept", "application/xml");

    //call web service and save result in a buffer
    BufferedReader br = new BufferedReader(new InputStreamReader(
        (connection.getInputStream())));
    //Print output/result from the server
    String output;
    System.out.println("Output from Server: \n");
    //Print response code
    System.out.println("Response Code: "+connection.getResponseCode());
    //Print output from the server
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
    //Close http connection
    connection.disconnect();
}
```

Exemplo em Java de uma aplicação cliente utilizando um método POST:

```
private static void CreateUModel1() throws IOException {
    System.out.println("Test REST MRI, CreateUModel1");
    // Set http parameters
    String urlParameters =
        "external_ID=12&gender=sd&marital_status=ewr&childrens_number=34&location=wer&profession=wr
e&education=wre&specialties=wer&languages=wre&user_history=re&keywords=wer&hobbies=er&bookm
arks=ewr&sports=rew&handicaps=wer&psych_model=wer&courseName=re&subject=erw&topics=wer&dura
tion=erw&sequence=re&userKnowledge=rew&history=rew&timeSpend=45&evaluation=45&performance=5
";
    // Set web service URI
    String uri = "http://hostname:port/mri/umodel";
    URL url = new URL(uri);
    // Open http connection
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    // Set connection properties
    connection.setDoOutput(true);
    connection.setDoInput(true);
    connection.setInstanceFollowRedirects(false);
    connection.setRequestMethod("POST");
    connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
    connection.setRequestProperty("charset", "utf-8");
    connection.setRequestProperty("Content-Length", "" +
        Integer.toString(urlParameters.getBytes().length));
    connection.setUseCaches(false);
    //Write http parameters
    DataOutputStream wr = new DataOutputStream(connection.getOutputStream());
    wr.writeBytes(urlParameters);
    wr.flush();
    wr.close();
    //call web service and save result in a buffer
    BufferedReader br = new BufferedReader(new InputStreamReader(
        (connection.getInputStream())));
    //Print output/result from the server
    String output;
    System.out.println("Output from Server: \n");
    //Print response code
    System.out.println("Response Code: "+connection.getResponseCode());
    //Print output from the server
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
    //Close http connection
    connection.disconnect();
}
```

```
}
```

Exemplo em C# de uma aplicação cliente utilizando um método GET:

```
static void Main()
{
    Console.WriteLine("TEST REST MRI");

    // Set web service URI
    var webRequest = System.Net.WebRequest.Create("http://hostname:port/mri/umodel/1.json")
    as HttpWebRequest;

    // Set connection properties
    if (webRequest != null)
    {
        webRequest.Method = "GET";
        webRequest.ServicePoint.Expect100Continue = false;
        webRequest.Timeout = 20000;

        webRequest.ContentType = "application/json";
    }

    // Call web service and save result
    HttpWebResponse resp = (HttpWebResponse)webRequest.GetResponse();
    Stream resStream = resp.GetResponseStream();
    StreamReader reader = new StreamReader(resStream);
    String ret = reader.ReadToEnd();

    // Write result to console
    Console.WriteLine(ret);
    Console.ReadKey();
}
```

Exemplo em C# de uma aplicação cliente utilizando um método POST:

```
static void Main()
{
    Console.WriteLine("TEST REST MRI");

    // Set web service URI
    System.Net.WebRequest request =
        (HttpWebRequest)WebRequest.Create("http://hostname:port/mri/umodel");
    request.Method = WebRequestMethods.Http.Post;
    request.ContentType = "application/x-www-form-urlencoded";

    // Set http parameters
    string postData =
        "external_ID=122&gender=sd&marital_status=ewr&childrens_number=34&location=wer&profession=we&education=wre&specialties=wer&languages=wre&user_history=re&keywords=wer&hobbies=er&bookmarks=ewr&sports=rew&handicaps=wer&psych_model=wer&courseName=re&subject=erw&topics=wer&duration=erw&sequence=re&userKnowledge=rew&history=rew&timeSpend=45&evaluation=45&performance=5";

    // Write http parameters
    request.ContentLength = postData.Length;
    StreamWriter postStream = new StreamWriter(request.GetRequestStream(),
        System.Text.Encoding.ASCII);
    postStream.Write(postData);
    postStream.Close();

    // Call web service and save result
    HttpWebResponse resp = (HttpWebResponse)request.GetResponse();
    Stream resStream = resp.GetResponseStream();
    StreamReader reader = new StreamReader(resStream);
    String ret = reader.ReadToEnd();

    // Write result to console
    Console.WriteLine(ret);
    Console.ReadKey();
}
```