• U  C •

Mestrado em Engenharia Informática

Sistema de Recomendação Baseado em Conhecimento

Luís Filipe Marçal dos Reis

Orientadores:

Hernani Costa

Luís Macedo

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

Setembro, 2012

Agradecimentos

Gostava de expressar a minha sincera gratidão ao Hernani Costa pela amizade, pelas ideias e orientação disponibilizados durante a realização do estágio desta tese.

Ao Professor Luís Macedo pelas condições e aconselhamento proporcionados na realização deste estágio.

Ao Bruno Leitão pela sua amizade e disponibilidade para discutir algumas questões relacionadas com este trabalho.

Aos meus Pais.

Ao meu Irmão.

À Marta.

Resumo

Verificamos hoje em dia um grande crescimento na quantidade de informação disponível. Este acontecimento resulta não só da massificação do acesso às tecnologias de informação e da globalização do acesso à Internet fixa, mas também é consequência do aumento do número de utilizadores que acedem à Internet a partir de dispositivos móveis.

O fenómeno das redes sociais, do trabalho colaborativo e a presença dos media *online* em muito contribuem para o crescimento exponencial da informação disponível. Deste modo, o ser humano está hoje exposto diariamente a uma enorme quantidade de informação, proveniente das mais diversas fontes, com a qual não consegue lidar. Deste modo, assistimos assim a um fenómeno claro de sobrecarga de informação para os humanos.

No contexto anteriormente descrito, os agentes pessoais e em particular os sistemas de recomendação têm cada vez mais um papel relevante, auxiliando os utilizadores de sistemas de informação a lidar com o processo de seleção e identificação dos conteúdos mais relevantes. Para que os agentes pessoais possam desempenhar o seu papel com eficácia, as preferências e os hábitos dos utilizadores têm que ser considerados no processo de seleção da informação. Torna-se assim necessário que estes possuam mecanismos capazes de identificar as necessidades e os objetivos de cada utilizador por forma a gerar recomendações úteis.

Neste sentido, esta tese propõe um sistema de agente pessoal baseado em conhecimento, capaz de recolher, categorizar e filtrar notícias autonomamente de acordo com os interesses noticiosos específicos dos utilizadores. O sistema recorre a aprendizagem computacional para identificar o perfil dos utilizadores, combinando o *feedback* dos utilizadores com conhecimento produzido a partir do conteúdo das notícias por forma a gerar recomendações.

Durante o desenvolvimento das várias etapas do trabalho são levados a cabo um conjunto de testes experimentais utilizando diversos utilizadores, permitiram aferir o correto funcionamento e desempenho do sistema de recomendação. Esta abordagem teve especial incidência na implementação de um sistema de agente pessoal que resulta na produção de uma aplicação móvel para dispositivos móveis.

Palavras-chave: extração de informação, extração de palavras-chave, extração de conhecimento, representação de conhecimento, grafos, sistemas de recomendação, agentes pessoais.

Abstract

Nowadays, we experience a huge growth on the amount of information available. This is a result not only from the wide spreading of information technology and desktop Internet access availability, but also from the increasing number of users that access the Internet from mobile devices.

The Social Networking phenomena, collaborative work and online media presence make a huge contribution to information availability. Humans are daily exposed to a huge quantity of information from many different sources that they cannot handle by themselves. As a result, there is an eminent problem of information overload for which humans are not ready to deal with.

In the aforementioned context, personal agents and in particular recommender systems, play an important role on aiding users with the identification and selection of relevant information. In order to perform these tasks efficiently, personal agents must consider users preferences when delivering information. The system must be aware of the users needs and goals in order to provide useful recommendations.

Having this in mind, this work proposes a knowledge-based personal agent system, capable of automatically gathering and selecting news, filtering and delivering them in a selective and personalized fashion to each user. The system uses machine learning in order to identify users models, while combining feedback from the agents network with a knowledge representation created from the extracted news text, in order to recommend more relevant content to users.

During the development of this work, experimental testing using a set of users was carried out, allowing the evaluation the recommender system's behaviour and performance. This approach was designed with personal mobile assistants in mind, resulting in the development of a mobile application prototype for the Google Android platform.

Keywords: information retrieval, keyword extraction, knowledge retrieval, knowledge representation, graphs, recommender systems, personal agents.

Glossário

AS Anotações Sociais

CHC Colaboração Humano-Computador

EM Entidades Mencionadas

GF Grafo de Conhecimento

IA Inteligência Artificial

IHC Interação Humano-Computador

PLN Processamento de Linguagem natural

MU Modelo de Utilizador

RSS Really Simple Syndication

BSD Berkeley Software Distribution

GPL General Public License

OWL Web Ontology Language

RDF Resource Description Framework

XML Extensible Markup Language

Índice

Lista de Figuras

Lista de Tabelas

1 INTRODUÇÃO

Nas últimas décadas temos assistido a uma autentica revolução do conceito de sociedade. A sociedade tal como a conhecemos tem vindo a sofrer enumeras redefinições e transformações com o alargamento da sua abrangência enquanto modelo de organização social. Um dos fatores que mais contribuiu para o desenvolvimento e mutação do conceito de sociedade foi a evolução tecnológica vertiginosa ocorrida no final do século XX. O crescimento e massificação do acesso à tecnologia, sistemas de informação contribuíram para o desenvolvimento de um modelo social focado na Informação. Este modelo é vulgarmente denominado por Sociedade da Informação ou Sociedade do Conhecimento.

Neste contexto assistimos ao aumento vertiginoso no crescimento da informação disponível provocado não só pela massificação do acesso às tecnologias de informação e da globalização do acesso à Internet fixa, mas também como consequência do aumento do número de utilizadores que acedem à Internet a partir de dispositivos móveis. Um estudo recente do *World Bank* revela que cerca de três quartos da população mundial tem agora acesso a um dispositivo móvel (World Bank 2012). Este número torna-se ainda mais relevante se considerarmos um outro estudo da *Morgan Stanley Research* (Meeker, Devitt, and L. Wu 2010) que aponta para que em 2014 o número de acesso à Internet a partir de dispositivos móveis ultrapassará o número de acessos através de dispositivos fixos.

O fenómeno das redes sociais, do trabalho colaborativo e a presença dos media *online* em muito contribuem para o crescimento exponencial da informação criada e disponibilizada a cada instante. Deste modo, o ser humano está hoje exposto diariamente a uma enorme quantidade de informação, proveniente das mais diversas fontes, com a qual não consegue lidar. Por outro lado muita da informação gerada não interessa de igual forma a todas as pessoas.

Assim, torna-se emergente a necessidade de criação de mecanismos que possam auxiliar os utilizadores de sistemas a selecionar a informação mais útil de acordo com os seus interesses.

Com o número de fontes informação a aumentar a cada dia, torna-se crítico um sistema que recomende notícias aos utilizadores e que possua a flexibilidade necessária para se adaptar às características específicas não só dos conteúdos, mas também dos utilizadores. O problema torna-se ainda mais evidente quando um utilizador tem interesse num leque alargado de temas, distribuídas física e logicamente (em termos de disponibilidade e acessibilidade), tornando essencial a criação de sistemas capazes de recolher informação das mais diversas formas e fontes, filtrando conteúdos seletivamente e de acordo com as preferências do utilizador.

Neste sentido, esta tese aborda um conjunto de técnicas de extração de informação, representação de conhecimento e aprendizagem computacional para que, a partir do conhecimento produzido possa ser construído um sistema de agente pessoal capaz de

filtrar informação relevante a cada utilizador. Mais concretamente o sistema proposto será vocacionado para a seleção e entrega personalizada de notícias em aplicações móveis. Os assistentes pessoais são agentes inteligentes capazes de ajudar os humanos a lidar com a tarefa de selecionar informação relevante. Estes sistemas têm em consideração as preferências e necessidades dos utilizadores por forma a selecionar e entregar apenas a informação mais útil a cada utilizador e em cada circunstância.

Por forma a atingir o objetivo proposto foi criado um sistema que permite desenvolver as seguintes tarefas:

- recolher informação a partir de diversas fontes noticiosas;
- categorizar a informação recolhida através de palavras-chave, recolhendo os tópicos mais relevantes por forma a criar uma representação estruturada da informação;
- permitir ao utilizador selecionar os tópicos de interesse de sua preferência por forma a combater a ausência de informação sobre o utilizador no arranque da aplicação;
- aprender dinamicamente os interesses do utilizador à medida que este vai lendo e classificando notícias, utilizando aprendizagem computacional;
- entregar informação personalizada e relevante a cada utilizador, com base nas suas preferências, tirando partido do conhecimento produzido bem como do seu *feedback*.

Para a construção do sistema foram idealizados três componentes modulares que têm como objetivo resolver três questões distintas: a recolha, seleção e categorização da informação, a estruturação da informação por forma a criar uma base de conhecimento e a rede de agentes que partilha conhecimento por forma a entregar informação de forma seletiva aos utilizadores. Uma descrição pormenorizada destes componentes pode ser encontrada no Capítulo 4, mais especificamente na Secção 4.1.

A conceptualização deste sistema consiste numa aplicação multiagente, capaz de entregar notícias de uma forma personalizada para cada utilizador, através de agentes pessoais.

No final deste projeto de investigação são esperadas as seguintes contribuições:

- Avaliar técnicas e ferramentas de extração automática de palavras-chave a partir de texto, estudando a sua utilização na etiquetagem automatizada de notícias.
- Estudar a extração de informação a partir de fontes de informação não estruturadas, com aplicação na produção de conhecimento.
- Representar conhecimento de forma dinâmica e estruturada, avaliando técnicas e ferramentas, na produção de uma representação de conhecimento adequada ao contexto dos Agentes.
- Analisar a performance e aplicação de algoritmos de deteção de comunidades em grafos, na identificação de *clusters* de tópicos de interesse.

- Identificar e avaliar métricas de autoridade em redes, estudando a sua aplicação na etiquetação de *clusters* de tópicos de interesse.
- Testar a adequação da informação extraída e do conhecimento produzido no contexto dos sistemas de recomendação. Mais especificamente a sua aplicação no treino de classificadores para recomendação de notícias.
- Estudar a utilização de classificadores na aprendizagem de modelos de utilizador e recomendação de notícias. Nomeadamente *RBF Networks*, *Naive Bayes* e *J48*.
- Avaliar a performance da combinação do *feedback* dos utilizadores com o conhecimento produzido no treino dos classificadores.

Da estrutura desta tese consistem seis capítulos adicionais:

Capítulo 2 Neste capítulo é abordado o estado de arte no contexto dos Sistemas de Recomendação.

Capítulo 3 Neste capítulo são abordados os conceitos fundamentais que servem de base à realização desta tese e desenvolvimento do sistema de recomendação.

Capítulo 4 Neste capítulo é descrita a abordagem seguida no desenvolvimento do sistema de recomendação, nomeadamente a sua arquitetura, implementação e avaliação experimental.

Capítulo 5 Neste capítulo é feita uma análise crítica a um conjunto de trabalhos relacionados que serviram de inspiração à concepção do sistema proposto.

Capítulo 6 Neste capítulo são apresentadas algumas considerações finais acerca deste trabalho e identificadas algumas questões relacionadas com o trabalho futuro.

2 ESTADO DA ARTE

Nesta secção é efetuada uma revisão do estado da arte no contexto dos Sistemas de Recomendação. Em 2.4 são identificados os diversos métodos de recomendação de conteúdos, identificando as suas características mais relevantes, bem como alguns dos seus problemas em 2.5. A criação de modelos de utilizador é também aqui abordada conjuntamente com técnicas de recolha do modelo de utilizador em 0. Ainda nesta secção são também identificados os métodos mais comuns de avaliação da performance e controlo da qualidade das recomendações 2.7.

2.1 Sistemas de Recomendação

Muitas vezes temos que tomar decisões sem que tenhamos a experiência pessoal suficiente para conhecer as diferentes alternativas. No dia-a-dia contamos muitas vezes com a recomendações de outros para tomar as decisões. Um sistema de recomendação assiste-nos neste processo social de tomada decisões baseado na experiência de outros (Resnick and Varian 1997).

Os autores do primeiro sistema de recomendação que se conhece, o *Tapestry* (Goldberg et al. 1992), referem-se frequentemente a "filtragem colaborativa" para designar "sistema de recomendação", este último é atualmente o termo mais generalizado. O *Tapestry* era um sistema de email experimental assente no pressuposto que a filtragem de informação é mais eficiente se envolver as pessoas no processo de filtragem. O sistema incluía suporte para filtragem baseada em conteúdo e filtragem colaborativa, aperfeiçoando a filtragem a partir das ações que os utilizadores tomavam ao ler as mensagens.

Hoje em dia, denomina-se vulgarmente por sistema de recomendação, todo o sistema que sugere ao utilizador opções ou recursos que se pretendem ser de seu interesse ou utilidade.

Para que um sistema de recomendação possa desempenhar a sua função é necessário que este entenda os problemas do utilizador, as suas necessidades, as suas dificuldades, os seus objetivos e as suas preferências. Deste modo é necessário que estes sistemas possuam uma representação de todas estas características. Mais ainda, é necessário que o sistema possua também algum tipo de conhecimento sobre o domínio em que vai auxiliar o utilizador.

Nas secções seguintes serão abordados uma série de tópicos que pretendem clarificar de que modo um sistema de recomendação pode descobrir as características de cada utilizador, de que modo pode aprender o conhecimento necessário das tarefas para que foi desenhado e de que modo a partir desta informação pode gerar recomendações úteis ao utilizador.

2.2 Modelo de Utilizador

Ainda que não seja fácil especificar este conceito, podemos de forma simples definir o Modelo de Utilizador (MU) como "*o conhecimento que se tem de um determinado utilizador, codificado de forma implícita ou explicita, que é utilizado pelo sistema para melhorar a interação*" (Kass and T Finin 1988).

Este conhecimento das características do utilizador é essencial para que o sistema de recomendação lhe possa fornecer recomendações que vão de encontro às suas espectativas.

A preocupação com a criação de modelos de utilizador surge inicialmente na área da Interação Humano-Computador (IHC), área esta que se preocupa em proporcionar suporte à colaboração entre humanos e computadores. Neste contexto podemos definir colaboração como "*um processo no qual dois ou mais agentes trabalham juntos por forma a atingir objetivos comuns*" (L. Terveen 1995). Aqui, os agentes devem ser entendidos como os elementos participantes na interação, neste caso, um humano e um computador. Da definição anterior podemos extrair algumas questões fundamentais presentes nos sistemas de recomendação, nomeadamente a existência de objetivos partilhados, contexto partilhado, coadaptação, coevolução e aprendizagem.

Existem duas vertentes essenciais na Colaboração Humano-Computador (CHC) que devem ser consideradas no desenvolvimento de sistemas de recomendação: a "emulação" e a "complementação". Na emulação pretende-se implementar "capacidades humanas" nos computadores por forma a melhorar a interação entre ambos, já na *abordagem de complementação* assume-se a natureza não humana dos computadores como forma de explorar esta assimetria na procura de novas possibilidades de interação e colaboração (Suchman 1987).

Deste modo podemos considerar que um sistema de recomendação deve ter a capacidade de se adaptar às mudanças nas necessidades do utilizador à medida que o assiste na realização de tarefas.

2.2.1 Adaptatividade vs. Adaptabilidade de Sistemas

Seguidamente explica-se de que formas podem ser desenhados os sistemas de recomendação para lidar com a dinâmica de adaptação ao utilizador.

No que à capacidade de adaptação dos sistemas diz respeito, existem duas abordagens essenciais, a *abordagem adaptativa* e a *abordagem adaptável*. Apesar de ambos refletirem a capacidade de adaptação de um sistema ao utilizador durante a interação, existem algumas diferenças de perspetiva importantes que interessa realçar. Uma análise critica destas duas abordagens comparando características, forças e fraquezas de ambas as abordagens é apresentada por Fischer (2001).

Na tabela seguinte são apresentados alguns dos aspetos mais relevantes:

	Adaptativa	**Adaptável**
Definição	O sistema adapta-se dinamicamente à tarefa e ao utilizador	O utilizador altera a funcionalidade do sistema (com suporte do sistema)
Conhecimento	Conhecimento contido no sistema	Conhecimento estende-se ao domínio para o qual foi desenhado e sistema
Forças	• Esforço reduzido do utilizador; • Reduzido conhecimento do sistema necessário	• Utilizador no controlo; • O utilizador conhece bem o seu papel
Fraquezas	• Dificuldade do utilizador em conceber um modelo coerente do sistema; • Perda de controlo por parte do utilizador	• Sistema tornam-se incompatíveis; • Utilizador trabalha substancialmente; • Complexidade para o utilizador aumenta (aprender o mecanismo de adaptação)
Requisitos	• Modelos de utilizador, tarefas e diálogos; • Conhecimento de objetivos e planos; • Atualização incremental dos modelos; • Mecanismo de correspondência poderoso	• Arquitetura em camadas; • Design orientado para um domínio; • Modelos do domínio • Sistema responde ao utilizador; • "Design Rationale" (razões para as opções tomadas no design)

Tabela 1 – Comparação entre sistemas adaptativos e adaptáveis (Fischer 2001).

Como podemos verificar pelas características apresentadas anteriormente as duas abordagens tomam perspetivas distintas quanto ao utilizador e sistema, se no caso dos *sistemas adaptativos* o sistema tem o papel ativo de se adaptar às condições e contexto em que utilizador realiza as tarefas utilizador, já no caso de um *sistema adaptável* o sistema "oferece" o controlo ao utilizador, permitindo que este possa alterar substancialmente o seu funcionamento por forma a desempenhar as tarefas.

2.2.2 Preferências do Utilizador

Preferência é uma relação de ordem entre um ou mais itens a caracterizar, que de entre um conjunto de possíveis escolhas, melhor satisfaz ou melhor se enquadra nos gostos do utilizador (Brafman and Domshlak 2009). De uma forma menos formal,

podemos definir preferências como o mecanismo que utilizamos guiar as nossas escolhas, separando o que gostamos do que gostamos pouco, do que não gostamos.

Numa perspectiva de aprendizagem, encontrar as preferências de um utilizador é um problema de pesquiza e optimização com o objetivo de identificar os itens que mais satisfazem o utilizador, dentro de um espaço de possíveis escolhas (Gemmis et al. 2009). Da mesma forma, no contexto dos sistemas de recomendação o objectivo é identificar os itens com maior probabilidade de satisfazer o utilizador, dentro de um espaço de possíveis itens a recomendar.

2.2.3 Recolha de "Feedback" do utilizador

Os sistemas de recomendação dependem em grande parte da qualidade da informação que direta ou indiretamente se consegue obter do utilizador. A importância do *feedback* do utilizador no contexto de sistemas baseados em conhecimento foi abordada por Maué (2008) no qual se conclui que em determinados domínios específicos esta informação é de extremo valor. O autor exemplifica com o caso do sucesso da *Wikipedia* onde fica bem patente o valor da utilização do chamado "conhecimento de massas". É importante recordar que na *Wikipedia* qualquer utilizador pode editar conteúdos e fornecer o *feedback* necessário à correção dos mesmos.

Nos sistemas de recomendação podemos identificar duas formas essenciais de *feedback* proveniente da classificação de itens por parte do utilizador:

Classificações Explicitas: estas classificações são comuns na maioria do sistema e consistem na atribuição de um valor numérico (i.e. {1,2,3,...,10}), qualitativo (i.e. {"bom", "mau"}; {"gosto", "não gosto"}) ou de utilidade (i.e. {"útil", "não útil"}). Esta forma de classificação permite que os dados possam ser processados estatisticamente por forma a obter dimensões, médias e distribuições.

Outra forma de classificação explicita mas que se apresenta mais complexa de processar corresponde à análise de opiniões em texto livre, as denominadas *reviews*, onde o utilizador escrevendo a sua opinião na forma de comentários aos conteúdos.

Alguns exemplos:

- Pedir ao utilizador que classifique qualitativamente um item.
- Pedir ao utilizador que classifique um conjunto de itens numa escala de valores.
- Pedir ao utilizador que escolha o seu item preferido entre dois itens apresentados.
- Pedir ao utilizador que apresente uma lista de itens que mais/menos gosta.

Classificações Implícitas: este tipo de classificação são normalmente utilizadas como complemento das técnicas explícitas como forma de obter mais informação sobre o utilizador de uma forma não obstrutiva. Estas técnicas monitorizam a interação do utilizador com o sistema registando os seus padrões de utilização.

Análise de comportamentos como: padrão de cliques, tempo gasto a ver um item, gestos do rato, escrita no teclado, padrões de navegação etc. são utilizados como forma de obter informação implicitamente.

Alguns exemplos:

- Observar os diferentes itens que o utilizador visita.
- Quantificar o número de vezes que um utilizador observa um item.
- Quantificar o tempo que um utilizador observa um item.
- Analisar a sequência de itens visitados e os itens visitados numa mesma sessão.
- Analisar a rede social do utilizador e descobrir gostos semelhantes.

2.2.4 Tipos de Modelo de Utilizador

Como se pode concluir a partir da alínea anterior a forma como recolhemos informação dos utilizadores, bem como a forma como a informação é disponibilizada, influencia o processo de design do modelo de utilizador.

Relativamente aos padrões de design do modelo de utilizador podemos identificar três grandes grupos:

- **Modelos estáticos:** o modelo permanece inalterado e estático assim que é fornecido ao sistema pelo utilizador. Em muitos dos casos é pedido ao utilizador que selecione itens de um conjunto fornecido de tópicos ou que indique algumas das suas preferências ou interesses.
- **Modelos estereotipados:** o modelo de utilizador é baseado em dados demográficos do utilizador, sendo estes agrupados em grupos estereótipo comuns. A aplicação faz assunções acerca do utilizador com base no modelo mesmo que não existam dados específicos do utilizador numa determinada área.
- **Modelos dinâmicos:** o modelo permite a atualização dinâmica dos dados que representam o utilizador, as mudanças nos interesses, os objetivos à medida que este interage com o sistema. A maioria dos sistemas de recomendação utilizam técnicas dinâmicas de aprendizagem do modelo de utilizador, atualizando e construindo incrementalmente o perfil do utilizador. Esta abordagem é claramente vantajosa num contexto de recomendação em tempo real.

Como se pode aqui perceber, a construção de cada um dos modelos apresentados depende na capacidade de quantificar um conjunto de elementos passíveis de serem recolhidos, a partir das observação das ações dos utilizador. A quantificação destes elementos é discutido na alínea que se segue.

2.2.5 Técnicas de Aprendizagem do Modelo de Utilizador

Existem atualmente diversas abordagens para identificar preferências do utilizador de modo a construir o seu modelo, nomeadamente técnicas de aprendizagem e técnicas

estatísticas. Como se poderá verificar mais adiante este problema é muito semelhante ao problema de classificação de informação, deste modo muitas das técnicas utilizadas para construir modelos de utilizador são também utilizadas para classificar informação, com as devidas adaptações de contexto.

Técnicas Estatísticas de Modelação de Perfis de Utilizador

Outro tipo de abordagem de construção de modelos de utilizador consiste em analisar estatisticamente um conjunto de informação que se conhece relativamente a um utilizador. Nesta abordagem pretende-se quantificar o que o utilizador mais gosta (o que prefere, ou que lhe é útil) e vice-versa. Mais uma vez podemos remeter o problema de classificação de utilizadores através de métodos estatísticos para o problema de classificação de informação descrito na 3.3.3. Os métodos apresentados constituem um conjunto de técnicas estatísticas/probabilísticas de classificação que com o devido enquadramento de contexto podem ser utilizadas para criar o perfil do utilizador.

Depois de aprender o modelo de utilizador, ou seja, depois de obter algum conhecimento sobre as necessidades dos utilizadores, o sistema de recomendação pode iniciar a geração de recomendações. Diversos sistemas de recomendação têm sido criados ao longo dos tempos (alguns aqui anteriormente descritos em tópicos anteriores) com os mais diversos objetivos.

Na secção seguinte são apresentados um conjunto de questões que identificam os objetivos mais comuns para os quais são criados os sistemas de recomendação.

Técnicas Baseadas em Aprendizagem Computacional

Uma abordagem comum consiste na construção de um classificador, ou seja, um modelo capaz de atribuir uma categoria a um determinado conjunto de dados. O classificador é induzido a partir de um conjunto de dados de treino, ou seja, ou conjunto pré selecionado de exemplos e classificados com as categorias a que pertencem.

Nesta abordagem o problema de aprendizagem do modelo de utilizador pode ser reduzido a um processo de categorização binária: classificar um item como interessante ou não para os interesses do utilizador.

Considerando um conjunto de categorias $C = \{c+, c-\}$, onde $c+$ corresponde à classe positiva (o que o utilizador gosta) e $c-$ corresponde à classe negativa (o que o utilizador não gosta), é possível implementar classificadores automáticos utilizando um conjunto de estratégias de aprendizagem computacional descritas na secção 3.3.1. Alguns dos métodos apresentados em 3.3.1 têm origem na classificação de texto, deste modo, na sua descrição é natural a referencia a documentos de texto e aos termos neles contidos por forma a simplificar a sua explicação. Ainda assim, as técnicas são igualmente válidas neste contexto.

2.3 Tarefas de um Sistema de Recomendação

Um trabalho bastante completo foi já realizado por Ricci et al. (2011) na identificação das tarefas mais importantes de um sistema de recomendação moderno e que resultou na lista que é apresentada de seguida. Esta análise foi efetuada essencialmente do ponto de vista dos criadores de sistemas, bastante vocacionada para a sua vertente comercial:

- **Aumentar o número de itens vendidos:** este é o papel mais importante num sistema de recomendação comercial, que tenta provocar a compra de itens ao sugerir mais opções alternativas interessantes ao utilizador.
- **Diversificar as vendas:** aqui o sistema de recomendação tem como função incentivar o utilizador a escolher itens menos populares, recomendando-os aos utilizadores certos.
- **Aumentar a satisfação do utilizador:** um sistema de recomendação eficaz em conjunto com uma interface bem desenhada pode gerar maior satisfação do utilizador, na medida em que o ajuda a encontrar mais facilmente itens que lhe agradam. Esta combinação promove uma experiência de utilizador mais agradável resultando num maior rácio de aceitação das recomendações.
- **Aumentar a fidelização do utilizador:** aqui o sistema de recomendação tem um papel importante do ponto de vista do reconhecimento de antigos utilizadores, por forma a oferecer-lhes um tratamento especial pela sua preferência. Esta é uma característica base destes sistemas, reconhecer os utilizadores através do seu histórico de utilização. Consequentemente, quanto maior é o grau de utilização maior será o conhecimento do perfil do utilizador.
- **Perceber as necessidades do utilizador:** esta função é de extrema importância na tomada de decisões em contextos comerciais. Perceber o que os utilizadores preferem ou mais necessitam pode ajudar a estabelecer por exemplo necessidades de stock de produto ou à criação de serviços potencialmente interessantes para os clientes.

Herlocker et al. (2004) propõem uma lista de tarefas que um sistema de recomendação pode ajudar a implementar:

- **Encontrar alguns itens adequados:** recomendar ao utilizador alguns itens ordenados por relevância.
- **Encontrar todos os itens adequados:** recomendar ao utilizador todos os itens que satisfazem as suas necessidades.
- **Destacar itens num contexto:** dado um determinado contexto o sistema deve destacar uma lista de itens baseando-se no histórico de preferências do utilizador.
- **Encontrar uma sequencia adequada:** recomendar uma sequencia de itens que no seu todo interessam ao utilizador.
- **Encontrar um pacote de itens adequado:** encontrar um conjunto de itens que agregados interessam ao utilizador.

- **Auxiliar a navegação:** ajudar o utilizador a navegar pelos diversos itens que de algum modo possam satisfazer as suas intenções naquela sessão.
- **Criar confiança:** disponibilizar ao utilizador mais séptico um conjunto de funcionalidades que o ajudem a testar a qualidade das recomendações.
- **Melhorar o perfil:** ajudar e incentivar o utilizador a revelar os seus gostos e preferências.
- **Auxiliar a expressão:** proporcionar satisfação ao utilizador que se sente realizado por expressar o seu conhecimento.
- **Ajudar terceiros:** proporcionar satisfação ao utilizador que se sente realizado por contribuir para o bem comum, sabendo que à partida estará a ajudar terceiros e não a si mesmo.
- **Influenciar terceiros:** existem utilizadores de sistemas de recomendação que têm como única função influenciar outros a adquirir determinados itens. Esta ação pode não ser de todo bem intencionada mas está contemplada neste tipo de sistema.

Após terem sido apresentados os objetivos gerais de um Sistema de Recomendação, em seguida são apresentadas as suas abordagens e tipos mais tradicionais.

2.4 Tipos de Sistemas de Recomendação

Nesta secção é apresentada uma síntese das técnicas de recomendação mais comuns, que podem ser enquadradas em seis grupos essenciais, descritos nas alíneas seguintes.

2.4.1 Filtragem Colaborativa

Esta técnica é uma das mais populares e mais implementadas nos sistemas de recomendação. A ideia passa por recomendar ao utilizador itens que outros utilizadores com gostos semelhantes, consideraram interessantes no passado.

Nesta abordagem, a proximidade nas preferências de dois utilizadores é calculada com base na semelhança entre o histórico de classificações que os utilizadores atribuíram a cada item.

2.4.2 Baseados em Conteúdo

Esta abordagem deriva do trabalho de investigação na área da filtragem de informação. As preferências dos utilizadores são aprendidas com base em características específicas dos itens que este classificou ou que simplesmente visitou. Como exemplos, podemos considerar a utilização das palavras-chave presentes num texto que um utilizador classificou ou visitou ou os nomes dos atores que constituem o elenco de um filme que o utilizador comprou. As palavras-chave são aqui utilizadas como propriedade (características) do objecto de texto.

Os métodos baseados em conteúdo utilizam perfis de itens (conjunto de propriedades discretas dos itens) que caracterizam os itens no sistema. O sistema cria então um perfil de conteúdos construindo um vetor de pesos associados às características dos itens. Estes pesos refletem a importância de cada uma dessas características para o

utilizador e podem ser calculados a partir dos vetores associados aos itens que o utilizador classificou.

Com base do perfil de utilizador e com base nos perfis associados aos itens o sistema de recomendação utiliza mecanismos de correspondência estatísticos ou de aprendizagem computacional para selecionar os itens a recomendar.

2.4.3 Baseados em Conhecimento

Este método tenta entregar sugestões ao utilizador com base no conhecimento acerca da necessidade de um utilizador para com um determinado item. Por este motivo alguns autores chamaram-lhe "Recomendação Utilitária" (Guttman, Moukas, and Maes 1998). As recomendações são efetuadas através do estabelecimento de medidas de utilidade, derivadas a partir do conhecimento que se possui das relações de um item para com um determinado utilizador. A recomendação baseada em conhecimento pressupõe uma estrutura que guarde estas relações e que permita a sua consulta de modo a determinar a utilidade para o utilizador inferindo novas recomendações. O domínio de conhecimento considerado está associado não só às preferências do utilizador mas também ao tipo de item a recomendar. Por exemplo, um sistema que recomenda viagens baseado em conhecimento pode tirar partido não só do que se conhece acerca da experiência do utilizador em viagens anteriores, mas também do que se sabe sobre as características dos locais que visitou e dos locais disponíveis para recomendar.

2.4.4 Comunitários

A recomendação comunitária é uma das abordagem mais populares atualmente devido aumento crescente de popularidade que se tem verificado nos últimos anos nas redes sociais. Um exemplo interessante é o caso do Bling[1], uma rede social recente, que convida os seus utilizadores a partilhar as suas compras. A aplicação sugere artigos aos utilizadores com base nas compras dos "amigos" dos utilizadores.

Este tipo de sistema de recomendação, agrega dados relativos às classificações e recomendações de objetos por parte de utilizadores que fazem parte de uma mesma rede ou comunidade, reconhecendo semelhanças entre utilizadores através da comparação das suas classificações. Novas recomendações são geradas por comparação de perfis de utilizadores dentro da mesma rede. É importante não confundir Recomendação Comunitária com Filtragem Colaborativa, ainda que o tipo de abordagem seja semelhante, a diferença reside na abrangência da "vizinhança" de utilizadores utilizada para calcular as recomendações. No caso da Recomendação Colaborativa segue-se o epigrama "diz-me com quem andas, dir-te-ei quem és", ou seja, são tidos em conta apenas as classificações de itens de utilizadores próximos ("amigos") do utilizador em questão.

[1] http://bling.io/
[2] http://codex.wordpress.org/Taxonomies

Diversos trabalhos têm surgido na comunidade científica que procuram melhorar as técnicas tradicionais de Filtragem Colaborativa, integrando aspectos sociais nas recomendações. Veja-se por exemplo (Pham et al. 2010; Golbeck 2006).

2.4.5 Demográficos

Um aspeto importante considerado por alguns autores passa por considerar o fator tempo no peso das classificações, isto porque, é provável que os interesses dos utilizadores variem no tempo e no espaço (N.-han Liu et al. 2009). Tendo este fator em consideração, a abordagem demográfica para recomendação caracteriza os utilizadores com base nos seus atributos pessoais. A partir de um conjunto de questões colocadas pelo sistema aos utilizadores, estes são encaixados em modelos estereótipo de utilizador. As recomendações são depois criadas utilizando um mecanismo de seleção que tem em conta o perfil demográfico do utilizador. A utilização de dados demográficos mostra-se eficaz como meio para aumentar a performance dos sistemas de recomendação (Yun, Yang, and Wang 2011).

Os sistemas aqui apresentados apresentam níveis de performance diferentes, quando colocados em condições distintas, porque têm em conta critérios distintos para gerar recomendações. Cada um dos métodos possui vantagens e desvantagens dependendo da informação que têm disponível, mas também do tipo de tarefa que têm que desempenhar. De seguida são apresentados alguns dos problemas mais comuns dos sistemas de recomendação.

2.4.6 Híbridos

Na sua maioria os modelos híbridos de sistemas de recomendação são combinações dos diversos tipos mencionados nas alíneas anteriores. A combinação de várias abordagens é utilizada frequentemente como forma de colmatar os pontos fracos de uma abordagem, com os pontos fortes de outras.

Em seguida são apresentadas algumas estratégias comuns utilizadas em sistemas de recomendação híbridos (Burke 2007):

- **Combinação de Pesos:** os resultados de diferentes medidas de recomendação são combinados numericamente.
- **Alternância de Técnicas:** o sistema escolhe entre diversos métodos de recomendação o que mais se adequa em cada situação.
- **Mistura de Resultados:** recomendações de diferentes tipos são entregues conjuntamente ao utilizador.
- **Combinação de Características:** características obtidas de diferentes fontes de informação e conhecimento são combinadas e entregues a um único sistema de recomendação.
- **Ampliação de Características:** uma técnica de recomendação é utilizada para calcular um conjunto de características que de seguida são utilizadas como entrada para uma próxima técnica.

- **Técnicas em Cascata:** diversos sistemas de recomendação são utilizados hierarquicamente, atribuindo uma ordem de prioridade a cada um, os sistemas com prioridades mais baixas são utilizados para quebrar empates nos resultados dos mais prioritários.
- **Meta-Níveis:** uma técnica de recomendação é utilizada para produzir um determinado tipo de modelo que será depois entregue a uma técnica seguinte.

Para uma descrição mais detalhada de cada uma das técnicas apresentadas ver por exemplo Burke (2002).

2.5 Problemas dos Sistemas de Recomendação

Anteriormente foram descritos alguns métodos de recomendação que resumidamente podemos enquadrar em dois grandes grupos: as técnicas com maior foco nas características dos objetos (Conteúdo) e as técnicas com maior foco nos utilizadores (Colaborativa, Comunitária, Demográfica, Conhecimento).

Seguidamente são apresentadas algumas vantagens e desvantagens de cada um dos dois grupos essenciais de técnicas de recomendação.

Um problema geral dos sistemas de recomendação é o chamado "arranque a frio" (*Cold Start*) que se torna mais evidente nos sistemas baseados em aprendizagem. O problema reside na incapacidade dos sistemas que aprendem as preferências do utilizador ao longo da sua interação com o sistema, de lidar com a chegada de um novo utilizador ou item, ou seja, utilizadores para os quais ainda não existiu qualquer tipo de registo de atividade e itens para os quais ainda não foi registada qualquer classificação.

Uma visão mais detalhada dos problemas de cada um dos critérios de recomendação é apresentada de seguida.

Técnicas focadas no conteúdo

- **Análise Limitada de Conteúdos:** as técnicas estão limitadas às características que manual ou automaticamente são associadas aos objetos. Se o objeto possuir poucos elementos descritivos associados a recomendação torna-se difícil.
- **Conteúdo Inacessível:** contrariamente aos objetos de texto, muitos tipos de objeto não possuem conteúdos passíveis de serem analisados pelo sistema de informação, dependendo totalmente da informação fornecida pelos utilizadores.
- **Ultra Especialização (*Serendipity*):** as técnicas não têm nenhum processo de identificar algo inesperado. Os sistemas recomendam apenas os objetos que têm grande correspondência com o perfil do utilizador.

Técnicas focadas no utilizador

- **Novo Item:** não é possível efetuar recomendações de itens recentes para os quais nenhum utilizador tenha atribuído classificação.
- **Novo Utilizador:** para efetuar recomendações precisas o sistema tem que primeiro aprender as preferências do utilizador, o que pode levar algum tempo.
- **Utilizador Incomum:** utilizadores com perfis e classificações pouco comuns não recebem recomendações muito precisas pois não têm muitas características em comum com a maioria dos utilizadores.
- **Esparcidade (*Sparsity*):** normalmente o número de classificações necessárias é muito inferior ao número de classificações realmente realizadas pelos utilizadores.
- **Escalabilidade:** os sistemas necessitam de dados de uma grande quantidade de utilizadores antes de se tornarem eficientes.
- **Transparência e Privacidade:** há utilizadores muito céticos em fornecer informações ao sistema com receio de questões de privacidade. Muitos sistemas de recomendação são completamente fechados, sem que o utilizador consiga perceber de que modo os seus dados estão a ser utilizados.

Para além das questões identificadas anteriormente, Resnick et al. (1997) identificaram dois problemas sociais associados aos sistemas de recomendação:

O primeiro é o possível egoísmo dos utilizadores, que assim que estabelecem o seu perfil de interesses se colocam ao sabor das recomendações de terceiros sem contribuir para o sistema com as suas recomendações. O segundo pela seriedade dos fornecedores de conteúdos, que por também poderem eles mesmos utilizar o sistema, podem simplesmente "inunda-lo" com recomendações positivas aos seus produtos e negativas aos produtos da concorrência. O autor apresenta ainda um terceiro problema crítico nestes tipo de sistema, a questão da privacidade.

Genericamente quanto mais informação for disponibilizada pelos utilizadores melhor serão as bases para a recomendação, contudo muitas pessoas não quererão os seus hábitos e preferências expostas ao conhecimento de terceiros.

Como resposta a muitos dos problemas aqui abordados são utilizadas abordagens híbridas, como as descritas em 2.4.6, que tiram partido do melhor que cada uma das técnicas oferece.

2.6 Comparação de Técnicas de Recomendação

Na tabela seguinte são comparadas as diferentes técnicas de recomendação descritas nos tópicos anteriores, destacando algumas das suas características funcionais.

Técnica	**Conhecimento**	**Dados de Entrada**	**Processo**
Colaborativa	Classificações de $\boldsymbol{U}$ itens em $\boldsymbol{I}$.	Classificações de $\boldsymbol{u}$ de itens em $\boldsymbol{I}$.	Identificar utilizadores em $\boldsymbol{U}$ semelhantes a $\boldsymbol{u}$ e extrapolar a partir das suas classificações de $\boldsymbol{i}$.
Comunitária	Classificações de $\boldsymbol{U}$ itens em $\boldsymbol{I}$.	Classificações de $\boldsymbol{u}$ de itens em $\boldsymbol{I}$.	Identificar utilizadores em $\boldsymbol{U}$ "amigos" de $\boldsymbol{u}$ e extrapolar a partir das suas classificações de $\boldsymbol{i}$.
Conteúdo	Características dos itens em $\boldsymbol{I}$.	Classificações de $\boldsymbol{u}$ de itens em $\boldsymbol{I}$.	Gerar um classificador que enquadre as classificações de $\boldsymbol{u}$ e as use em $\boldsymbol{i}$.
Demográfica	Informação demográfica de $\boldsymbol{U}$ e as suas classificações em I.	Dados demográficos sobre $\boldsymbol{u}$.	Identificar utilizadores demograficamente semelhantes a $\boldsymbol{u}$, extrapolando a partir das suas classificações de $\boldsymbol{i}$.
Conhecimento	Características dos itens em $\boldsymbol{I}$. Conhecimento da utilidade dos itens para os utilizadores em U.	Uma descrição dos interesses e necessidades de $\boldsymbol{u}$.	Inferir uma correspondência entre $\boldsymbol{i}$ e as necessidades e interesses de $\boldsymbol{u}$.

Tabela 2 – Tabela comparativa de técnicas de recomendação.

Para interpretar a tabela acima (Tabela 2) considere a seguinte notação:

- $\boldsymbol{U}$ – Conjunto de Utilizadores para os quais se conhecem as preferências.
- $\boldsymbol{I}$ – Conjunto de Itens que podem ser objeto de recomendação.
- $\boldsymbol{u}$ – Utilizador para o qual é necessário gerar recomendações.
- $\boldsymbol{i}$ – Item para o qual se pretende calcular a preferência de u.

2.7 Avaliação de Sistemas de Recomendação

Avaliar a performance de algoritmos de recomendação consiste essencialmente em avaliar a satisfação do utilizador. Tipicamente estamos interessados em perceber o grau de aceitação das recomendações, ou seja, quantificar o número de vezes que os utilizadores aceitam ou rejeitam itens recomendados.

A qualidade de um sistema de recomendação pode ser avaliada comparando as recomendações a um conjunto de teste de classificações do utilizador conhecidas. Esta avaliação é efetuada recorrendo a um conjunto de métricas conhecidas como *predictive accuracy metrics* (Herlocker et al. 2004). Estas métricas permitem comparar as classificações previstas com as classificações reais dos utilizadores.

Uma das métricas mais comuns é a *Mean Absolute Error* (MAE), expressa pela Equação 2.1, que mede a diferença média entre classificações previstas e as classificações reais.

Equação 2.1

$$MAE = \frac{\sum_{\{u,i\}} |p_{u,i} - r_{u,i}|}{N}$$

Outra métrica relacionada com a anterior é a *Root Mean Square Error* (RMSE), expressa pela Equação 2.2, que coloca maior importância em erros absolutos maiores.

Equação 2.2

$$\text{RMSE} = \sqrt{\frac{\sum_{\{u,i\}} (p_{u,i} - r_{u,i})^2}{N}}$$

Nas equações anteriores $p_{u,i}$ representa a classificação prevista do utilizador u pata o item i, $r_{u,i}$ representa a classificação real e N o número de classificações contidas no caso de teste.

As métricas preditivas consideram todos os itens da mesma forma, ou seja, atribuem-lhe a mesma relevância. Na maioria dos casos existe maior interesse em avaliar os itens que são do interesse do utilizador. Deste modo interessa discriminar as boas das más recomendações, na medida em que o processo de recomendação consiste em entregar boas recomendações.

Neste sentido, Gunawardana et al. (2009) apresenta uma discussão sobre os métodos tradicionais de avaliação da recomendação, resumindo de forma simplificada três métricas mais comuns que são de seguida apresentadas.

Considerando um universo de itens a recomendar, podemos considerar as seguintes possibilidades:

	Recomendados	**Não Recomendados**
Preferidos	Verdadeiro-Positivo (VP)	Falso-Negativo (FN)
Não Preferidos	Falso-Positivo (FP)	Verdadeiro-Negativo (VN)

Tabela 3 – Resultados possíveis de uma recomendação.

Podemos então contar o número de ocorrências que se enquadram em cada uma das categorias e calcular as seguintes métricas:

Métrica	**Expressão**	**Interpretação**
Precision	$\frac{Nº\ de\ VP}{Nº\ de\ VP + Nº\ de\ FP}$	Mede a probabilidade de um item recomendado ser relevante.
Recall	$\frac{Nº\ de\ VP}{Nº\ de\ VP + Nº\ de\ FN}$	Mede a probabilidade de um item relevante ser recomendado.
False Positive Rate	$\frac{Nº\ de\ FP}{Nº\ de\ FP + Nº\ de\ VN}$	Mede a probabilidade de um item ser mal recomendado.
F-measure (F_1)	$2 \times \frac{Precision + Recall}{Precision \times Recall}$	Combina as características do *Precision* e do *Recall* numa métrica conjunta.

Tabela 4 – Métricas de performance de um sistema de recomendação.

Uma descrição mais detalhada destas e outras métricas como, *Pearson's product-moment correlation*, *Kendall's* τ, *Mean average precision*, *Half-life utility* e *Normalized distance-based performance measure* podem ser encontradas em Herlocker et al. (2004).

3 CONCEITOS FUNDAMENTAIS

Neste capítulo são abordadas as temáticas relevantes para a elaboração e entendimento do sistema proposto que constitui o objeto de estudo desta tese.

Inicialmente são descritas algumas questões gerais que introduzem o conceito de conhecimento no contexto dos sistemas de recomendação (Secção 3.1), nos quais se inclui uma descrição das formas de representar conhecimento, em sistemas baseados em conhecimento. São também analisadas neste capítulo um conjunto de ferramentas que permitem a criação de estruturas de conhecimento em sistemas de informação.

De seguida na Secção 0 aborda-se o problema da classificação da informação, onde são identificadas métricas e métodos que permitem identificar elementos relevantes em corpos de texto, nomeadamente palavras chave e categorias. Esta secção apresenta ainda um conjunto de ferramentas de processamento de informação que permitem levar a cabo os processos de classificação da informação.

Ainda na Secção 0 são identificadas algumas técnicas de identificação de Tópicos de Interesse e categorização da informação, nomeadamente o particionamento de grafos e a identificação de comunidades.

Por último na Secção 0 são enquadrados os Agentes Inteligentes no contexto dos sistemas de recomendação, onde são apontados os elementos que os tornam adequados a este tipo de sistema baseado em conhecimento.

3.1 Conhecimento

A maioria das questões sobre as quais se debruça a Inteligência Artificial (IA) estão relacionados com o conhecimento. São várias as áreas onde a IA intervém para resolver problemas onde o conhecimento é o elemento fulcral, nomeadamente: a representação de conhecimento, a aquisição de conhecimento, as bases de conhecimento e os sistemas baseados em conhecimento.

De seguida são abordadas algumas questões que se prendem com o conhecimento em IA do ponto de vista da engenharia. As questões do conhecimento são de extrema importância no contexto dos Agentes Inteligentes como vamos poder verificar em 3.5. Nesse contexto o conhecimento representa o elemento essencial dos sistemas baseados em conhecimento na medida em que representam a matéria a partir da qual os agentes realizam o seu trabalho (ver 3.5.1).

3.1.1 O que é o Conhecimento?

(Levesque and Lakemeyer 2000) definem o conhecimento como uma relação entre um agente (aquele que sabe) e uma proposição (o que se sabe). Esta noção vem da análise da forma como formalmente falamos e da forma como a nossa linguagem representa o saber.

Por exemplo, na frase "O João sabe que a Maria vem à festa" podemos claramente identificar o agente (João) e a frase declarativa "a Maria vem à festa" a proposição. O conhecimento é neste caso uma relação entre o "conhecedor" e o "o que se conhece".

Proposições

Ainda que de forma abstrata as proposições representam os fatos, ou seja, o conteúdo do que se conhece acerca de um determinado tema. Apesar do carácter abstrato das proposições o que realmente interessa saber do ponto de vista da IA é o seu valor "verdadeiro" ou "falso", "correto" ou "incorreto". As proposições representam então o julgamento que alguém fez sobre o estado do mundo que o rodeia e que se aceitam ser verdadeiras. Em 2.4.1 quando se fala em Agentes Baseados em Conhecimento, estamos a considerar agentes que possuem ou têm acesso a um conjunto de proposições verdadeiras sobre o mundo que os rodeia.

O interesse da IA é que os sistemas saibam o mais possível acerca do seu mundo. Quando se fala em sistemas baseados em conhecimento, considera-se um sistema que possui muito conhecimento tal como foi descrito anteriormente mas também um sistema que faz o que faz porque possui uma representação desse conhecimento.

Raciocínio

Quando consideramos a representação do mundo que rodeia um agente podemos pensar que existem uma infinidade de proposições que o podem representar, o que não implica que todas elas tenham que ser representadas. Compete ao raciocínio fazer a ponte entre o que foi de fato representado e o conjunto total de proposições que se aceitam ser verdade.

O raciocínio é o processo de manipulação dos símbolos que representam as proposições nas quais se acreditam, por forma a produzir novas representações e consequentemente novas proposições.

3.2 Representação de Conhecimento

A Representação de Conhecimento é então o campo da IA preocupado em estudar a utilização formal de símbolos para representar um conjunto de proposições nas quais um determinado agente acredita. Podemos afirmar que um sistema sabe p se possuir uma representação simbólica de p. Em IA pretendemos construir sistemas que contenham representações simbólicas que possam ser entendidas como proposições e que se comporta de determinada maneira à custa dessa mesma representação simbólica.

O que identifica um sistema baseado em conhecimento é a presença de uma "Base de Conhecimento". Uma "Base de Conhecimento" é uma representação simbólica que contém um conjunto de proposições em que um agente acredita e sobre as quais opera raciocínio.

Nas alíneas seguintes serão analisadas algumas das formas mais utilizadas para representar e estruturar conhecimento do ponto de vista da engenharia. São também abordados alguns mecanismos importantes relacionados com a integração do conhecimento em sistemas baseados em conhecimento.

3.2.1 Taxonomias

Uma taxonomia é um vocabulário controlado em que cada termo possui relações de descendência hierárquica com outros termos equivalente ou semelhantes (Whittaker and Breininger 2008). A estrutura de uma taxonomia pressupõe a utilização de relações de subclasse, ou seja, cada instancia de uma classe pode assumir-se ser uma instancia das suas categorias pai. As taxonomias são criadas e atualizadas por grupos de especialistas em cada uma das matérias que estas representam.

As taxonomias são de extrema importância nos sistemas de informação pela forma como permitem consistentemente categorizar a informação. Por exemplo, o uso de subcategorias taxionómicas em pesquisas permite simplificar o processo de construção da procura, na medida em que o utilizador não precisa de conhecer todo o vocabulário de um domínio.

As taxonomias tiveram a sua origem no campo das ciências naturais mas têm vindo a ser cada vez mais utilizadas e implementadas em sistemas de informação. O Wordpress[2] é um sistema de gestão de conteúdos que utiliza o conceito de taxonomia para organizar os artigos, páginas e ficheiros que os utilizadores criam e disponibilizam.

As taxonomias representam estruturas muito interessantes do ponto de vista da classificação de informação por permitirem obter uma representação das relações entre os termos pertencentes a um determinado domínio ou tópico, tornando-as ideais para a sua utilização em tarefas de agregação e categorização de conteúdos.

3.2.2 Bases de Dados Relacionais

Uma Base de Dados Relacional consiste numa coleção de tabelas relacionadas entre si que guardam um conjunto de dados.

Codd (1970) estabeleceu um conjunto de princípios que hoje servem de base ao conceito de Base de Dados Relacional. Na sua proposta defende que os dados devem ser independentes do "hardware" e do sistema de armazenamento utilizado, devendo o sistema permitir que se possa "navegar" automaticamente pelos dados. Na prática, isto significa que os dados deveriam ser guardados em tabelas e que deveriam existir relações lógicas entre as mesmas. Fox and McDermott (1986) testaram a adequação de alguns motores de base de dados relacionais da época a sistemas baseados em conhecimento e concluíram a sua utilidade à medida que os problemas da Inteligência Artificial se tornavam mais complexos e mais dependentes de grandes volumes de dados.

[2] http://codex.wordpress.org/Taxonomies

3.2.3 Ontologias

As Ontologias têm sido utilizadas amplamente em Engenharia do Conhecimento e Inteligência Artificial em aplicações de processamento de linguagem natural, gestão de conhecimento, extração de informação e internet semântica.

"Uma Ontologia define não só os termos básicos e as relações de vocabulário de um determinado tópico mas também as regras para combinar esses termos e relações por forma a criar extensões desse vocabulário" (Neches et al. 1991).

Os autores referidos anteriormente propuseram uma nova forma de construir sistemas inteligentes, rompendo com o paradigma utilizado até então, a construção das bases de conhecimento a partir do zero cada vez que era necessário utilizar conhecimento num sistema. Na sua proposta imaginaram um formato novo no qual estas seriam construídas utilizando componentes reutilizáveis. Este novo ideal de sistema promove a interação com outros sistemas, permitindo sua a interoperabilidade na utilização do conhecimento e dos mecanismos de resolução de problemas. Estas ideias resultaram num conjunto de projetos que culminaram na conceptualização do que hoje podemos chamar de Ontologia. Uma estrutura modular, que permite guardar conhecimento declarativo e resolver problemas operando sobre o conhecimento através de mecanismos de raciocínio.

Mais recentemente e numa perspectiva do ponto de vista da Engenharia de Sistemas Fensel (2002) define uma Ontologia como:

"um entendimento partilhado e comum de um domínio que pode ser comunicado entre pessoas e sistemas de aplicações heterogéneos e distribuídos"

Uma Ontologia descreve conceitos, instâncias (dos conceitos) e propriedades (associadas aos conceitos). Estes elementos são expressos formalmente ou através de uma linguagem que um computador possa entender, nomeadamente através dos formatos *Web Ontology Language* (OWL) e *Resource Description Framework* (RDF).

Os dados descritos por uma Ontologia no formato OWL são interpretados como um conjunto de "indivíduos" e um conjunto de "afirmações sobre os indivíduos" que os relacionam entre si. O OWL foi criado para disponibilizar uma forma sistematizada de processar informação Web, permitindo que seja lida facilmente por uma aplicações e sistemas informáticos. O RDF segue basicamente o mesmo conceito que o OWL ainda que menos potente como linguagem. O OWL possui um vocabulário maior e uma sintaxe mais potente que o RDF, tornando-o mais interpretável por máquinas. Em ambos os casos é utilizado *eXtensible Markup Language* (XML) para representar os dados produzidos.

Algumas de Ontologias Importantes

- **Wordnet**[3] (G. A. Miller et al. 1990) é uma base de dados lexical, que contém grupos de palavras da Língua Inglesa, constituídos por conjuntos de sinónimos denominados *synsets*. Contém também descrições genéricas e relações semânticas entre estas conjuntos. O seu objetivo é constituir um recurso estruturado que possa dar suporte a aplicações de análise de texto.
- **OpenCyc**[4] (Matuszek et al. 2006) é um projeto dedicado à construção da maior Ontologia generalista até à data, com o objetivo de proporcionar conhecimento "humano" a aplicações de Inteligência artificial.

3.2.4 Grafos de Conhecimento

No contexto da representação de conhecimento introduzimos aqui o modelo de "Grafo Conceptual" proposto por Sowa (1976) que tem vindo a ser desenvolvido ao longo de mais de três décadas e que deu origem a diversas modificações, adaptações e extensões nos mais diversos contextos da representação de conhecimento.

Um grafo conceptual é também conhecido por "Grafo de Conhecimento" (GC) no contexto dos sistemas baseados em conhecimento. Um Grafo de Conhecimento é um método de representação pertencente à categoria das redes semânticas.

A composição básica de um Grafo de Conhecimento compreende dois tipos de nós, nós que representam *tokens* (1) e nós que representam *relações* (2). Os *tokens* podem representar termos (entidades) ou tipos (grupos/classes de termos), as *relações* representam associações entre os *tokens* e/ou *tipos*.

1. Tokens: tudo aquilo que um ser humano consegue percepcionar no mundo real dá origem a um token na mente, ou seja, é toda a coisa do mundo real que pode ser representada simbolicamente. Neste sentido, num grafo de conhecimento tudo o que percepcionamos no mundo real pode ser representado por tokens e pode aqui ser interpretado como uma simplificação de um conceito.
2. Tipos: subjetivamente podemos afirmar que pessoas diferentes podem descrever experiencias do mundo real através de diferentes tokens. Por outro lado se muitas pessoas (ou todas) tiverem uma percepção semelhante de uma experiencia podemos afirmar que existe uma figura objectiva. Deste modo observamos que existem tipos idênticos de tokens que pertencem a uma mesma classe, permitindo introduzir tipos para expressar estes tokens.
3. Relações: estabelecem associações entre os conceitos, aqui representados por tokens. Eventualmente estes nós podem conter apenas afirmações sobre os tokens através de relações unárias.

Como foi referido anteriormente um GC é constituído por dois tipos de nós, os nós que representam conceitos (*tokens*) e os nós que representam relações entre *tokens*.

[3] http://wordnet.princeton.edu/
[4] http://opencyc.org/

Cada nó pode ser etiquetado com um determinado *tipo* adicionando um marcador especifico à sua etiqueta. De outro modo consideramos que o nó representa um conceito não especificado. Um aspecto importante a referir é a existência de números nas ligações entre nós que indicam a ordem pela qual devem ser interpretadas as relações.

Na imagem seguinte podemos observar um pequeno exemplo da representação de conceitos utilizando um Grafo Conceptual.

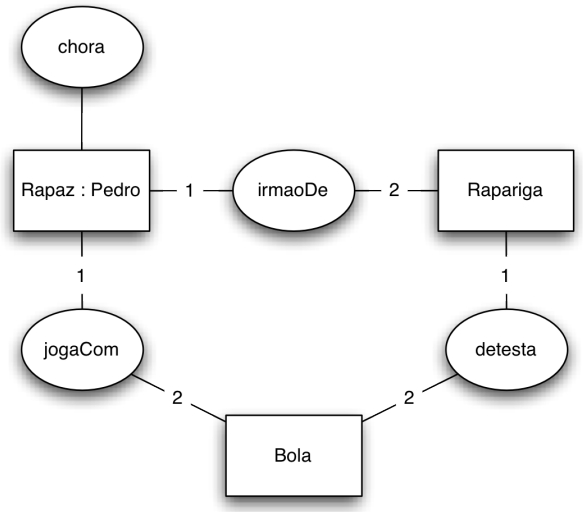


Imagem 1 – Um grafo conceptual básico.

A partir do grafo representado na imagem acima podemos afirmar o seguinte:

"O Pedro que é um Rapaz que joga à Bola, é irmão de uma Rapariga que detesta a Bola; O Pedro chora."

3.2.5 Ferramentas e Bibliotecas de Representação

De seguida são apresentadas algumas bibliotecas e ferramentas de representação de conhecimento.

- **Neo4j:** uma base de dados de alta performance desenhada e construída sobre grafos. O facto de ser construída com base em grafos permite que o programador trabalhe com uma estrutura em rede, orientada a objetos, flexível e dinâmica. A ferramenta é altamente escalável, lidando facilmente com milhares de nós. Esta biblioteca possui um conjunto de algoritmos de procura ideais para pesquisar nós da estruturas em rede.
    - **Licença:** Open Source (GPLv3)
    - **Linguagem:** Java
    - **Website:** http://neo4j.org

- **JGraphT:** é uma biblioteca que disponibiliza um conjunto de estruturas e objetos para construção de vários tipos de grafos. A biblioteca possui também

algoritmos para iterar sobre o grafo construído e efetuar pesquisas. O seu design é simples e permite construir grafos a partir de qualquer tipo de objeto Java, inclusivamente grafos de grafos.
    - **Licença:** Open Source (GPL)
    - **Linguagem:** Java
    - **Website:** http://www.jgrapht.org

- **QuickGraph:** é uma biblioteca que disponibiliza um conjunto de estruturas e objetos para construção de vários tipos de grafos. A biblioteca possui também algoritmos para iterar sobre o grafo construído e efetuar pesquisas. A biblioteca foi desenhada com a portabilidade em mente incluindo suporte para .Net 4.0, Silverlight 4.0, Windows Phone 7, XBox 360 e Windows 8 Metro.
    - **Licença:** Open Source (GPL)
    - **Linguagem:** .Net
    - **Website:** http://quickgraph.codeplex.com

- **NetworkX:** é uma framework que permite criar, manipular a estrutura, dinâmica e funções de redes complexas. A framework disponibiliza um conjunto de algoritmos comuns utilizados em grafos. O seu design prevê a criação de nós de vários tipos que podem conter virtualmente qualquer tipo de dados associados aos nós.
    - **Licença:** Open Source (BSD)
    - **Linguagem:** Python
    - **Website:** http://quickgraph.codeplex.com

- **SNAP:** é uma biblioteca genérica para análise de redes e procura em grafos, facilmente escalável para redes de grande escala. O seu design suporta a utilização de tipos complexos de dados.
    - **Licença:** Open Source (BSD)
    - **Linguagem:** C++
    - **Website:** http://snap.stanford.edu/snap/index.html

- **Gephi:** é uma ferramenta de visualização, manipulação e análise de redes e grafos. O Gephi implementa um conjunto de algoritmos de análise, representação, visualização e *clustering* de nós em grafos. Para além de uma API Java que permite a sua implementação em código próprio este disponibiliza também um editor gráfico.
    - **Licença:** Open Source (GPL)
    - **Linguagem:** Java
    - **Website:** http://gephi.org

3.3 Classificação de Informação

Em meados da década de 60 Luhn (1957) propõe a utilização de palavras como unidade de indexação de documentos e medidas de coocorrência de palavras entre documentos como critério de seleção na pesquiza de informação. O seu trabalho foi importante na medida em que para além de abordar o problema do armazenamento e indexação de informação, abre portas à possibilidade de se poder recolher a informação de forma automática e com base em critérios definidos pelo utilizador.

Salton e Buckley (1988) desenvolveram alguns dos mais importantes trabalhos de base relativamente à utilização de métricas e pesos associados às palavras para na extração de informação. No seu trabalho fica evidente a relação entre os pesos (utilizado medida de importância) atribuídos aos termos e a eficiência na extração de informação. Segundo estes, existem dois pontos fulcrais na extração de informação: a capacidade de extrair elementos potencialmente relevantes para o utilizador e a capacidade de rejeitar os elementos não relevantes.

3.3.1 Classificação e Categorização de Informação

No âmbito deste projeto pretende-se extrair elementos relevantes de cada notícia que as possam caracterizar relativamente ao seu conteúdo. Este processo consiste na extração de tópicos e palavras-chave que possam constituir elementos diferenciadores por forma a enquadrar cada notícia numa determinada categoria. Para realizar deste processo é necessário a realizar um conjunto de tarefas de processamento de linguagem natural, nomeadamente: a extração de tópicos, a extração de palavras-chave e o reconhecimento de entidades mencionadas.

O processo de classificação é um elemento crítico de sistemas baseados em conhecimento, como é o caso dos sistemas de recomendação. Por esse motivo, de seguida são abordadas as metodologias mais importantes de classificação e processamento de informação. Passando por um conjunto de processos de processamento de linguagem natural, essenciais à extração de conhecimento a partir de texto. Existem atualmente dois tipos principais de abordagem na classificação e categorização de informação, as abordagens baseadas em métodos probabilísticos (estatísticos) e abordagens baseadas em aprendizagem computacional, que serão apresentadas em detalhe nas secções seguintes.

3.3.2 Classificação baseada em Aprendizagem

No âmbito desta tese consideramos a classificação e categorização de informação um elemento muito importante pois esta é o elemento base sobre o qual o sistema atua, quer no caso da informação contida nas notícias, quer no caso da informação que pretendemos obter dos utilizadores. Uma das formas de realizar estes processos de classificação é utilizando técnicas de aprendizagem, ou seja construir algoritmos que possam automaticamente e a partir de exemplos, reproduzir os mesmos efeitos de classificação a novos conteúdos.

Em aprendizagem computacional pretende-se construir classificadores que consigam efetuar a categorização da informação automaticamente, criando um modelo a partir de exemplos. A categorização não é mais do que o processo de atribuição de uma classe a um determinado objecto.

De seguida são descritas algumas das técnicas de aprendizagem mais utilizadas neste contexto.

Support Vector Machines (Vapnik 1999) é um método baseado em aprendizagem utilizada para classificar e categorizar dados de forma supervisionada. Tecnicamente as *Support Vector Machines* (SVM) funcionam através da criação de híper-planos num espaço multidimensional que separam objetos pertencentes a classes distintas. O objectivo central das SVM é o ajustamento, a partir de exemplos, de uma função discriminante (vector de suporte) que maximiza a margem de separação entre as classes consideradas e a fronteira de decisão, minimizando desta forma o erro na classificação.

A utilização de SVM na aprendizagem de classificadores a partir de texto para automaticamente classificar e categorizar informação foi abordada por Joachims (1998). A sua investigação demonstrou a boa performance desta abordagem na classificação de informação mesmo em cenários onde são considerados um número elevado de características, identificando um conjunto de vantagens relativamente a outros métodos tradicionais de classificação.

Naive Bayes (Sahami et al. 1998) consiste num algoritmo probabilístico pertencente à classe dos "Classificadores Bayesianos". As *Naive Bayes* (NB) geram um modelo probabilístico baseado em observações anteriores dos dados, permitindo obter uma probabilidade *a posteriori* de um determinado item pertencer a uma determinada classe. Este conceito é baseado na teoria de probabilidade Bayesiana (Bayes 1763).

De uma forma simplificada podemos descrever o Teorema de Bayes na seguinte expressão representada pela Equação 3.1.

Equação 3.1

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- $P(c|d)$ é a probabilidade à posteriori de um item d pertencer à classe c.
- $P(c)$ é a probabilidade de observar um item na classe c.
- $P(d|c)$ é a probabilidade de observar o item d dada a classe c.
- $P(d)$ é a probabilidade de observar o item d.

Para classificar o item d, escolhe-se a classe com maior probabilidade:

$$c = argmax_{c_j} \frac{P(c_j)P(d|c_j)}{P(d)}$$

$P(d)$ é normalmente removido por ser igual para todo o c_j. Como não conhecemos os valores de $P(d|c)$ e de $P(c)$, estes são estimados a partir da observação dos dados de treino.

Decision Trees (Von Neumann and Morgenstern 1944) são constituídos por grafos em árvore nos quais os nós internos estão etiquetados com termos, sendo que os ramos que deles partem estão etiquetados com o peso que o termo tem no documento de teste e os nós folhas estão etiquetadas com categorias. As árvores de decisão são construídas dividindo sucessivamente os documentos de teste em grupos, até que estes contenham apenas documentos pertencentes a uma determinada categoria. O teste utilizado para dividir os documentos em grupos é feito com base nos pesos que os termos contidos nos nós têm no documento. A escolha dos termos utilizados é usualmente efetuada com base num função de ganho ou entropia de informação.

A partir das árvores de decisão é também possível construir um tipo de classificador denominado *Decision Rule Classifier,* esta técnica é semelhante às árvores de decisão, na medida em que segue a mesma abordagem de divisão de dados em grupos descrita anteriormente. As regras de decisão podem ser extraídas a partir de árvores de decisão existentes. Esta abordagem tende a gerar classificadores mais compactos na medida em seleciona a melhor de entre as regras mais abrangentes (as que cobrem corretamente todos os exemplos de treino) de acordo com uma função minimizante.

Entre os decisores deste tipo mais utilizados podemos encontrar o ID3 (J R Quinlan 1986) e o C4.5 (J. Ross Quinlan 1993) que é uma extensão da versão original do ID3. Uma implementação Java em código aberto do C4.5 pode ser encontrada ferramenta J48[5] que foi construída sobre o Weka[6]. Paralelemente e no caso das *Decision Rule Classifiers* podemos encontrar o C4.5rules (J.R. Quinlan, Cohen, and Hirsh 1994) (que cria regras a partir de árvores geradas pelo C4.5) e o CN2 (Clark and Niblett 1989).

k-Nearest Neighbour (Fix and Hodges 1951) é um método de aprendizagem baseado também conhecido por *Lazy Learning*, que aborda o problema de classificar itens novos comparando-os com a informação que têm guardada em memória (exemplos de treino) utilizando uma função de similaridade (i.e. Distância Euclidiana).

Neste processo, determinam-se os *k-nearest neighbors* (kNN), ou seja, encontram-se os *k-items* mais próximos do novo item a classificar e é-lhe atribuída uma categoria extrapolada a partir das categorias dos seus *nearest neighbors*.

Para encontrar os kNN é necessário utilizar uma métrica de semelhança, nomeadamente e caso os itens estejam representados por *Vector Space Models* (VSM) pode ser utilizada *Cosine similatity* (CS), uma métrica que mede a semelhança entre dois vetores a partir do ângulo que formam entre si. A CS tenta verificar se dois vetores apontam para uma mesma direção. Clarificando, um *Vector Space Model*

[5] http://www.opentox.org/dev/documentation/components/j48
[6] http://www.cs.waikato.ac.nz/ml/weka/

consiste num conjunto de vetores que contêm informação sobre as características dos itens.

Neural Networks (Bain 1873) são compostas uma cadeia de neurónios artificiais que imitam o comportamento dos neurónios biológicos, permitindo resolver problemas de processamento de dados com base em informação que circula dentro e fora da rede. De uma forma simples, as redes neuronais processam os dados de entrada com base em modelos matemáticos que ajustam a "força" das ligações entre os neurónios (que representam aqui unidades de processamento) para produzir um determinado resultado de saída esperado.

Neste caso os interesses do utilizador são representados pelos neurónios de saída, que são atingidos pro um padrão específico na rede. As *Neural Networks* (NN) possuem mecanismos de retro propagação que em caso de erro identificam ao neurónio responsável ajustando os seus parâmetros.

O conceito de rede neuronal foi aplicado a modelos computacionais por McCulloch e Pitts (1943), modelos a que denominaram por *Threshold Logic*.

RBFNetworks (Radial Basis Function Networks) (Orr 1996) são um tipo de redes neuronais artificiais com aplicação em problemas de aprendizagem supervisionada, nomeadamente regressão e classificação. Este tipo de rede neuronal utiliza funções de base radial, funções não-lineares que podem ser utilizadas em modelos de regressão linear com parâmetros lineares ou não-lineares. Tipicamente as *RBF Networks* possuem três camadas, uma camada de entrada, uma camada escondida com uma função de ativação não-linear e uma camada de saída linear.

A saída $\varphi : \mathbb{R}^n \to \mathbb{R}$ da rede pode ser definida na seguinte expressão representada pela Equação 3.2:

Equação 3.2

$$\varphi(\mathrm{X}) = \sum_{i=1}^{N} a_i \rho(\|\mathrm{X} - c_i\|)$$

onde N representa o número de neurónios na camada escondida, c_i corresponde ao vetor central para o neurónio i e a_i os pesos do neurónio de saída linear.

Relativamente à norma, tipicamente é utilizada a Distância Euclidiana e a função de base é normalmente uma função Gaussiana:

Equação 3.3

$$\rho(\|\mathrm{X} - c_i\|) = exp(-\beta\|\mathrm{X} - c_i\|^2)$$

As *RBF Networks* são estimadores universais em um subconjunto de $\mathbb{R}^n$, deste modo uma RBF composta por um número suficiente de neurónios na camada escondida consegue aproximar qualquer função contínua com precisão arbitrária. Os pesos a_i, c_i e β são determinados de maneira a otimizar a correspondência entre φ e os dados.

3.3.3 Métodos Estatísticos de Classificação

Não só através de aprendizagem é possível criar algoritmos de classificação, abordagens utilizando processos estatísticos têm vindo a ser propostos recorrentemente. Este tipo de abordagem procura tirar partido de aspectos estatísticos dos elementos a classificar, mais concretamente, e no caso da classificação de documentos de texto, de aspectos semânticos dos elementos que constituem o texto. Deste modo as técnicas probabilísticas de classificação recorrem a uma série de métricas auxiliares que permitem aferir e quantificar relações entre os objetos em estudo.

De seguida, são apresentados alguns dos métodos probabilísticos mais relevantes, bem como um conjunto de métricas bastantes utilizadas para classificar objetos e/ou conjuntos de informação.

Conditional Random Fields (Lafferty, McCallum, and F. Pereira 2001) são uma abordagem estatística utilizada para identificar ou etiquetar uma sequência de dados das quais se conhecem determinadas características estruturais. Este método consiste na construção um modelo com base nas relações estatísticas existentes entre diversas observações de um determinado padrão, procurando construir interpretações consistentes dessas relações que permitam identificar novos elementos que se enquadrem no modelo criado.

Considerando que estamos a processar texto, este processo funciona tirando partido das relações sintáticas entre os elementos constituintes de o texto escrito. A sintaxe é específica de cada tipo de linguagem e define a forma como as frases são construídas. É esta estrutura sistemática da linguagem que permite criar modelos para identificar os seus constituintes básicos (nomes, verbos, adjetivos, etc.).

Os *Conditional Random Fields* (CRF) podem ser utilizados, por exemplo, para extrair Entidades Mencionadas (EM) de documentos. Uma abordagem propôs um método para extrair e categorizar entidades mencionadas a partir da Wikipedia[7] (Watanabe, Asahara, and Matsumoto 2007). Considerando que na Wikipedia cada assunto corresponde a um artigo (Documento Web), o problema de extração de entidades mencionadas foi reduzido a um problema de classificação de documentos. Na sua abordagem Watanabe et al. (2007) propõem a construção de um grafo a partir das hiperligações entre documentos disponíveis na Wikipedia e a utilização de CRF para classificar os nós do grafo.

Term Frequency - Inverse Document Frequency (Salton and Buckley 1988) é uma medida estatística utilizada para avaliar a importância (w) de um determinado termo (t_i) para um determinado documento (d_j) num determinado universo (D) de documentos representada na Equação 3.4.

A métrica *Term Frequency - Inverse Document Frequency* (TF-IDF) tem como base o princípio de que a importância de um termo para um texto, de um mesmo

[7] http://wikipedia.org

conjunto, aumenta proporcionalmente ao número de vezes que nele ocorre, sendo diluído pelo aumento do número de vezes que o termo ocorre em outros textos do mesmo conjunto.

Podemos então descrever esta métrica do seguinte modo,

Equação 3.4

$$w(t_i, d_j) = \left(1 + log_2\, tf(t_i, d_j)\right) * log_2 \left(\frac{|D|}{F(t_i)}\right)$$

- D = $(d_1, d_2, d_3, \dots, d_n)$ representa uma coleção contendo n documentos.
- $|D|$ número de documentos contidos numa coleção D.
- $tf(t_i, d_j)$ "*term frequency*" corresponde ao número de ocorrências de um termo num dado documento.
- $F(t_i)$ corresponde ao número de documentos onde t_i ocorre.

Nas expressões anteriores, $\{n, i, j\} \in \mathbb{N}$.

Cocitation (Small 1973), representada pela Equação 3.5 é uma métrica de correlação utilizada para medir a semelhança entre documentos proposta por inicialmente utilizada para identificar semelhança em documentos científicos através da análise das referencias. Esta medida foi também adaptada para medir semelhança entre páginas Web (Cristo et al. 2003).

Podemos então descrever esta métrica do seguinte modo,

Equação 3.5

$$c(t_i, t_j) = \frac{R(t_i \cap t_j)}{R(t_i \cup t_j)}, \{i, j\} \in \mathbb{N}.$$

- $R(t_i \cap t_j)$ corresponde ao número de documentos que contêm os dois termos.
- $R(t_i \cup t_j)$ corresponde ao número de documentos que contêm pelo menos um dos termos.
- $t_i\ e\ t_j$ correspondem a dois termos (palavras ou conjunto de palavras) de um texto.

Term Coocurrence (coocorrência) corresponde genericamente à ocorrência de dois termos num mesmo corpus de texto, tendo como base a *Distributional Hypothesis* (Harris 1954) que propõe que palavras que ocorrem num mesmo contexto revelam tendência para possuir o mesmo significado. Esta abordagem tem sido frequentemente utilizada no campo da linguística computacional, por exemplo na desambiguação de sentido de termos/palavras (Dagan, Lee, and F. C. N. Pereira 1998) e (Schütze 1998), correção ortográfica em contexto (Jones and Martin 1997) e mais recentemente no campo da Ciência Cognitiva (McDonald and Ramscar 2001) como forma de influenciar a aprendizagem do sentido semântico de palavras através da sua colocação em contexto.

Pode-se considerar que o significado de uma palavra pode variar ao longo de várias dimensões. Aos modelos que procuram representar esta variação de forma coerente, distribuindo termos num espaço geométrico dá-se o nome de *Semantic Space Models* (SSM). Extraindo as frequências de coocorrência de uma palavra num corpo alargado de texto em linguagem natural, é possível posicionar essa palavra ao longo do espaço de cordo com o grau de coocorrência com outras palavras que compõem a dimensão do espaço. Deste modo, duas palavras que tendem a coocorrer em contextos linguísticos semelhantes, ou seja, que apresentam distribuições semelhantes, serão colocadas com maior proximidade no espaço semântico.

A coocorrência entre termos pode então ser definida na sua forma mais simples como:

Equação 3.6

$$C(t_i, t_j) = R(t_i \cap t_j), \{i, j\} \in \mathbb{N}.$$

- $R(t_i \cap t_j)$ corresponde ao número de documentos, de um determinado conjunto, que contêm os dois termos.
- $t_i \; e \; t_j$ correspondem a dois termos (palavras ou conjunto de palavras) de um documento.

Ao espaço no qual se considera a coocorrência de dois termos designa-se por "janela de contexto". Esta definição é útil quando são considerados corpos de texto não delimitados entre si, sendo que se pode definir coocorrência para uma determinada área (janela) do corpo de texto. Sendo que a frequência de coocorrência do termo t_i com o termo t_j considerando uma "janela de contexto" pode ser definida como: o número de vezes que t_j ocorre numa janela de n palavras em torno de t_i, para todas as ocorrências de t_i.

A partir do conceito de coocorrência é possível construir uma estrutura denominada *Rede de Coocorrência*. Esta rede é gerada através da agregação na forma de grafo de cada um dos pares de termos seguindo o critério da coocorrência. Os pares coocorrentes podem ser designados por "vizinhos" que frequentemente formam "bairros" como vai poder ser observado mais à frente no desenvolvimento desta tese (ver 3.4.3).

3.3.4 Bibliotecas e Ferramentas de Classificação

Existem um conjunto de ferramentas disponíveis potencialmente úteis no contexto deste trabalho, mais concretamente nos processos de classificação e categorização de informação.

De seguida são apresentadas algumas dessas ferramentas que foram analisadas no âmbito deste trabalho:

- ***Weka***[8] (Hall et al. 2009) é uma ferramenta que implementa um conjunto de algoritmos de aprendizagem computacional (alguns descritos em 3.3.1), destinada ao processamento e prospecção de informação.

  O *Weka* permite a sua utilização como aplicação (*framework*), permitindo aplicar os seus algoritmos diretamente a um corpo de dados bem como a sua utilização na forma de biblioteca que pode ser integrada e utilizada em código escrito na linguagem Java.

  De entre as aplicações que o *Weka* pode ter, para além dos já referidos, podemos ainda destacar, o *clustering* e a visualização de dados.

  Linguagem: Java

- ***Keyword Extraction Algorithm***[9] (KEA) (Witten et al. 1999) é um algoritmo baseado em aprendizagem implementado sobre a base do *Weka*, que permite extrair palavras-chave e frases-chave a partir de documentos de texto. O KEA utiliza alguns dos métodos descritos em 3.3.1 para classificar a informação, nomeadamente a métrica TF-IDF em conjunto com Naive Bayes.

  Linguagem: Java (baseado no *Weka*)

- ***General Architecture for Text Engineering***[10] (GATE) (Cunningham et al. 2011) é uma ferramenta capaz de realizar uma vasta gama de tarefas de processamento de linguagem natural, incluindo extração de informação em diversas línguas. O GATE consiste basicamente numa extensão do *Weka*, integrando um conjunto vastíssimo de extras, que permitem por exemplo editar Ontologias. Outra das funcionalidades interessantes do GATE é a integração com motores de pesquisa, permitindo obter resultados de pesquisa através de funções integradas.

  Tal como o *Weka*, o GATE pode ser utilizado como biblioteca ou na forma de aplicação. O GATE vai ainda mais longe, permitindo integrar outras bibliotecas como o OpenNLP descrito de seguida.

  Linguagem: Java (baseado no Weka)

- ***OpenNLP***[11] (Hockenmaier, Bierner, and Baldridge 2000) é uma biblioteca que permite realizar operações de processamento de linguagem natural. Entre as várias ferramentas incluídas podemos encontrar um extrator de entidades mencionadas, um etiquetador gramatical e um identificador de coreferências.

  Linguagem: Java

[8] http://www.cs.waikato.ac.nz/ml/weka/
[9] http://www.nzdl.org/Kea/
[10] http://gate.ac.uk/
[11] http://incubator.apache.org/opennlp/

- ***Natural Language Toolkit*[12] *(NLTK)*** (Loper and Bird 2002) é uma ferramenta composta por um conjunto de bibliotecas capazes de efetuar tarefas de Processamento de Linguagem Natural (PLN) e aprendizagem computacional. Este ferramenta inclui ainda um conjunto de dados de teste e modelos de aprendizagem pré-treinados para testar a aplicação.

  Linguagem: Python

3.3.5 Serviços de Classificação de Informação

Para além das bibliotecas e ferramentas descritas anteriormente, foram também analisadas algumas aplicações *Web* que disponibilizam na forma de *API* um conjunto de serviços interessantes no contesto deste trabalho. Estas ferramentas funcionam com base em conhecimento obtido em motores de pesquisa, combinados com técnicas de aprendizagem estado da arte, que foram também já abordadas nas secções anteriores.

De seguida são apresentados dois exemplos que no contexto deste trabalho se revelam potencialmente interessantes de explorar.

- ***AlchemyAPI*[13]** (Orchestr8 LLC 2012) é um serviço que disponibiliza um conjunto de funções para realizar tarefas de processamento de linguagem natural e prospecção de dados.
  A utilização do serviço está disponível comercialmente e para efeitos de investigação. O acesso ao serviço é efetuado através de uma API.

  Funções disponibilizadas: extração de entidades mencionadas, extração de palavras-chave, análise sentimental, extração de fatos e relações semânticas, categorização de documentos e detecção de língua.

- ***Yahoo! Content Analysis API*[14]** é um serviço disponibilizado pela Yahoo![15] para efetuar análise de conteúdos. Este serviço disponibiliza um conjunto de funcionalidades que podem ser acedidas através de uma API.

  Funções disponibilizadas: extração de entidades, extração de categorias, extração de relações, medição de relevância e associação de entidades com entradas na Wikipedia.

[12] http://nltk.org
[13] http://www.alchemyapi.com/
[14] http://developer.yahoo.com/contentanalysis/
[15] http://yahoo.com

3.4 Identificação de Tópicos de Interesse

Como já foi referido no tópico anterior, as etiquetas (palavras-chave) associadas a documentos, pelos utilizadores ou extraídas computacionalmente, a partir do texto representam uma ferramenta valiosa no contexto dos sistemas de recomendação.

A partir destas etiquetas e dos corpos de texto a que estão associadas, é possível extrapolar tópicos de interesse genéricos que permitem identificar utilizadores potencialmente interessados nesses tópicos.

Um dos problemas dos sistemas de recomendação, prende-se com a vastidão de tópicos de interesse que um sistema pode conter, por outro lado, sabe-se que existem tópicos sinónimos (i.e. carros e automóveis) ou que representam abreviações (i.e. NY e Nova York). Posto isto, para que os tópicos de interesse possam ser corretamente identificados, torna-se necessário realizar um processo de desambiguação e eliminação desta redundância, agregando os tópicos em grupos coerentes.

Ao processo de agregação de tópicos em categorias mais abrangestes dá-se o nome de *Clustering*. Esta técnica tem vindo a ser utilizada nos mais diversos domínios, desde a medicina (Kande, Savithri, and Subbaiah 2007; Mohamed and Salama 2007; Reich and Bondell 2011) ao processamento de imagem (Ilea and Whelan 2006; Silakari, Motwani, and Maheshwari 2009) passando pelo processamento de linguagem natural (Matsuzaki, Miyao, and Tsujii 2003).

A coocorrência de tópicos num mesmo contexto tende a gerar padrões de proximidade entre palavras do mesmo campo lexical (ver 3.3.3), ou seja, palavras de campos lexicais próximo tendem a coocorrer. Utilizando métricas de semelhança e proximidade é possível identificar *clusters* de palavras que são semanticamente próximas. Da mesma forma, é possível identificar palavras que pela sua maior coocorrência com outras do mesmo campo lexical podem representar o conceito geral em torno do qual todos os termos próximos orbitam (categorias).

A figura seguinte ilustra este efeito esperado de *clustering* da coocorrência de palavras do mesmo léxico sob a forma de grafo:

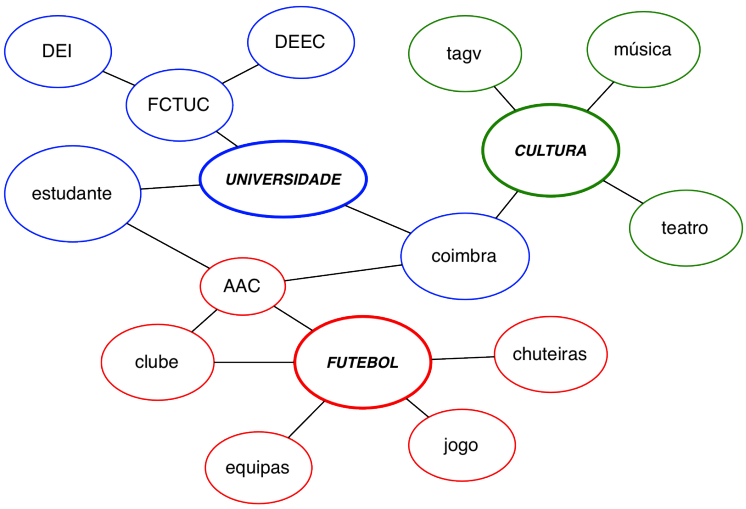


Imagem 2 – *Clustering* de tópicos em grafos.

3.4.1 Particionamento de Grafos

O problema da identificação de tópicos de interesse pode ser reduzido à tarefa de particionar um grafo, que contém a rede de associações entre os termos que representam os tópicos de interesse (Dhillon, Guan, and Kulis 2005).

Se considerarmos um grafo $G = (V, E, S)$ constituído por um conjunto de vértices V e um conjunto de relações E de modo a que uma relação representa a semelhança entre dois vértices, podemos representar a matriz de semelhança S, como a matriz $|V| \times |V|$ cujas entradas representam os pesos das relações de semelhança (uma entrada em S é zero se não existir qualquer relação entre os dois vértices).

Podemos assim definir o problema do particionamento como o processo de encontrar k partições distintas $\{V_1, \dots, V_k\}$ de modo a que a sua união é V, ou seja, V é a união dos k *clusters*.

As abordagens mais comuns (Dhillon et al. 2005) para resolver o problema de corte de grafos são:

Considerando $links(A, B)$ como a soma dos pesos das relações entre os nós em A e B, temos:

Equação 3.7

$$links(A, B) = \sum_{i \in A, j \in B} S_{ij}$$

Considerando ainda que o grau de A ($degree(A)$), representa o número de ligações dos nós em A para todos os outros vértices, temos:

Equação 3.8

$$degree(A) = links(A, V)$$

Podemos então definir as seguintes métricas de corte:

***Ratio Association* (RA):** esta abordagem, expressa pela Equação 3.9, que é também conhecida como *Average Association* tem como objetivo a maximizar as ligações no interior do *cluster* relativamente ao tamanho do *cluster*.

Equação 3.9

$$RA(G) = \max_{V_1,\dots,V_k} \sum_{c=1}^{k} \frac{links(V_c, V_c)}{|V_c|}$$

***Ratio Cut* (RC):** esta abordagem, expressa pela Equação 3.10, difere da abordagem anterior (RA), na medida em que procura minimizar o corte entre *clusters* e os restantes nós. Deste modo temos,

Equação 3.10

$$RC(G) = \min_{V_1,\dots,V_k} \sum_{c=1}^{k} \frac{links(V_c, V \setminus V_c)}{|V_c|}$$

***Kernighan-Lin Objective* (KLO):** esta abordagem, expressa pela Equação 3.11 é praticamente igual à abordagem anterior, exceto que é necessário que as partições tenham tamanhos iguais. É vulgar a utilização desta abordagem para $k = 2$ partições, ainda assim podemos generalizar para um número arbitrário k de partições.

Por uma questão de simplicidade assume-se que $|V|$ é divisível por k.

Equação 3.11

$$KLO(G) = \min_{V_1,\dots,V_k} \sum_{c=1}^{k} \frac{links(V_c, V \setminus V_c)}{|V_c|},$$

$$com\ |V_c| = \frac{|V|}{k}\ \forall c = 1, \dots, k.$$

***Normalized Cut* (NC):** esta abordagem, expressa pela é provavelmente a mais popular das técnicas de particionamento de grafos e tem como objetivo minimizar o corte relativamente ao *degree* (grau) de um *cluster* e não ao seu tamanho.

Equação 3.12

$$NC(G) = \min_{V_1,\dots,V_k} \sum_{c=1}^{k} \frac{links(V_c, V \setminus V_c)}{degree(V_c)}$$

3.4.2 Clustering Aglomerativo Hierárquico

Hierarchical Agglomerative Clustering: Este método de *clustering* proposto por Gower et al. (1969), consiste numa abordagem "*bottom up*" de construção de *clusters*.

Inicialmente cada item constitui um *cluster* e a cada iteração do algoritmo, pares de *clusters* são fundidos para construir grupos mais largos. Uma versão inversa ("top down") consiste numa abordagem divisiva, onde os itens formam um *cluster* único que a cada iteração do algoritmo vai sendo dividido em grupos mais específicos.

Ambos os métodos dependem da utilização de métricas de similaridade ou dissimilaridade entre os pares de itens para realizar a aglomeração ou divisão dos "clusters". Para itens de texto são utilizados frequentemente distancias de Hamming (Hamming 1950) ou Levenshtein (Levenshtein 1966).

Existem diversas versões de *hierarchical clustering*, das quais se destacam o *Complete-link Clustering* ou o *Single-link Clustering*.

O caso *Single-link* define a distância entre dois *clusters* como a mínima distância entre os seus membros (ver Equação 3.13).

Equação 3.13

$$single\_dist(A,B) \equiv \min_{\vec{x}\in A,\vec{y}\in B} \|\vec{x}-\vec{y}\|$$

é chamado de *Single-link* porque afirmar que os *clusters* são próximo mesmo que sejam apenas um par de pontos.

O caso do *Complete-link* define a distância entre dois *clusters* como a máxima distância entre os seus membros (ver Equação 3.14).

Equação 3.14

$$complete\_dist(A,B) \equiv \max_{\vec{x}\in A,\vec{y}\in B} \|\vec{x}-\vec{y}\|$$

3.4.3 Deteção de Comunidades

Community Detection: Um dos problemas dos métodos tradicionais de *clustering* prende-se com o facto de ser necessário definir à partida o número de partições a considerar. Este fator torna-se relevante num contexto de sistemas que integram um elevado número de palavras-chave, sendo bastante difícil estimar previamente o valor do número de partições. Por outro lado a maioria das abordagens não são facilmente escaláveis.

Recentemente um conjunto de métodos de *clustering* que abordam o problema numa vertente de deteção de comunidades têm surgido como forma de contornar os problemas dos métodos tradicionais. Genericamente os métodos de deteção de comunidades procuram identificar grupos de nós de um grafo que se encontram mais conetados entre si do que ao resto do grafo. O fato de neste tipo de abordagem não

ser definido à partida o número de partições a produzir implica que se torne necessário a definição de uma condição de paragem dos algoritmos, mais concretamente torne-se necessário a definição de uma função de qualidade das partições.

Neste contexto a função de qualidade mais popular e amplamente utilizada é a função de modularidade Q proposta por Newman-Girvan (Girvan and M. E. J. Newman 2002). O conceito de modularidade é baseado na ideia que para um qualquer grafo não é esperado que este possua estruturas que correspondam a *clusters*, então a possibilidade de existência de *clusters* é revelada através da comparação entre a real densidade das ligações de um subgrafo e a densidade que seria esperada para esse mesmo subgrafo independentemente da sua estrutura comunitária. A densidade esperada depende da escolha de um *null model*, ou seja, uma cópia do grafo original que mantém algumas das propriedades estruturais, mas excluindo a estrutura comunitária. O *null model* consiste num modelo aleatório que assume a possibilidade de cada nó do grafo poder estar ligado a qualquer outro nó do grafo. A escolha deste modelo é arbitrária, existindo diversas possibilidades de escolha no que à distribuição probabilística diz respeito. A escolha do modelo é muitas vezes condicionada pelo tipo de rede a analisar.

A modularidade de uma partição é então um escalar com valores entre -1 e 1 que mede a densidade das ligações dentro das comunidades, comparada com as ligações entre comunidades. Segundo o conceito de *null model* descrito anteriormente, se preservarmos o grau dos vértices na rede mas os ligarmos aleatoriamente, então a probabilidade de existir um nó entre os vértices i e j é $P_{ij} = \frac{k_i k_j}{2m}$ com k_i o grau do nó i. Então no caso de redes com pesos associados às ligações (pesos das arestas) podemos definir a Modularidade Q da seguinte forma (M. Newman 2004)

Equação 3.15

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - P_{ij}\right)\delta\left(c_i, c_j\right),$$

onde o somatório considera todos os pares de vértices, A_{ij} corresponde à matriz de adjacência (pesos das arestas entre os nós i e j). c_i é a comunidade ao qual o vértice está associado, sendo que $\delta(u, v)$ devolve o valor 1 caso $u = v$ e zero caso contrário e $m = \frac{1}{2}\sum_{ij} A_{ij}$ o número de arestas no grafo.

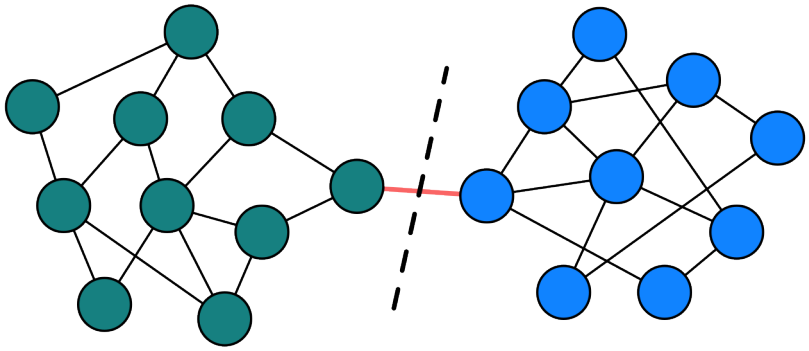
O Algoritmo de Newman-Girvan

Os métodos tradicionais de deteção de comunidades envolvem quase sempre alguma forma de *clustering* hierárquico, como o descrito em 0, envolvendo uma medida de proximidade entre os elementos da rede. As medidas de proximidade consistem em muitos casos no cálculo da distância ou do numero de caminhos que ligam ou passam por cada par de nós, evidenciando os nós mais centrais da rede. Estes métodos apesar de oferecerem resultados razoáveis tendem a evidenciar algumas falhas em algumas situações pois tendem a isolar nós periféricos e pouco ligados ao resto da rede. Por

exemplo, se um nó estiver ligado apenas por uma aresta ao resto da rede este deverá pertencer à comunidade presente no outro lado dessa aresta.

De modo a combater este efeito, o algoritmo de Newman-Girvan (Girvan and M. E. J. Newman 2002) propõe uma inversão no processo, procurando encontrar as arestas menos centrais, ou seja, arestas que se encontram "entre" comunidades. Adicionalmente em vez de construir as comunidades agregando nós, procuram-se as comunidades retirando arestas ao grafo original. Por forma a identificar as arestas a remover, é utilizado uma métrica de Edge Betweeness, uma generalização da *Betweenness Centrality,* medida de centralidade de nós (ver 0) aqui aplicada a arestas. Partindo do principio que se existem comunidades altamente ligadas entre si, existiram arestas que ligam essas comunidades, de forma a que a maioria dos caminhos entre comunidades passam por essas arestas. Estas arestas são então aquelas com maior valor de *Edge Betweeness*.

Removendo as arestas de maior *Edge Betweeness* é possível separar as comunidades presentes no grafo.



O algoritmo segue então o seguinte processo:

1. Calcula-se a *Edge Betweeness* para todas as arestas da rede.
2. Remove-se a aresta com maior *Edge Betweeness*.
3. Recalcula-se a *Edge Betweeness* para todas as arestas afetadas pela remoção anterior.
4. Repete-se a partir do passo 2 até não existirem mais arestas ou a função de Modularidade (ver Equação 3.15) ser óptima.

Deteção eficiente de comunidades

Infelizmente a maximização da Modularidade é um problema de procura computacionalmente pesado, em que o espaço de procura aumenta exponencialmente relativamente ao número de nós. Uma solução eficiente consiste no algoritmo *"Fast unfolding of communities in large networks"* (Blondel et al. 2008). Este algoritmo é composto por duas fases:

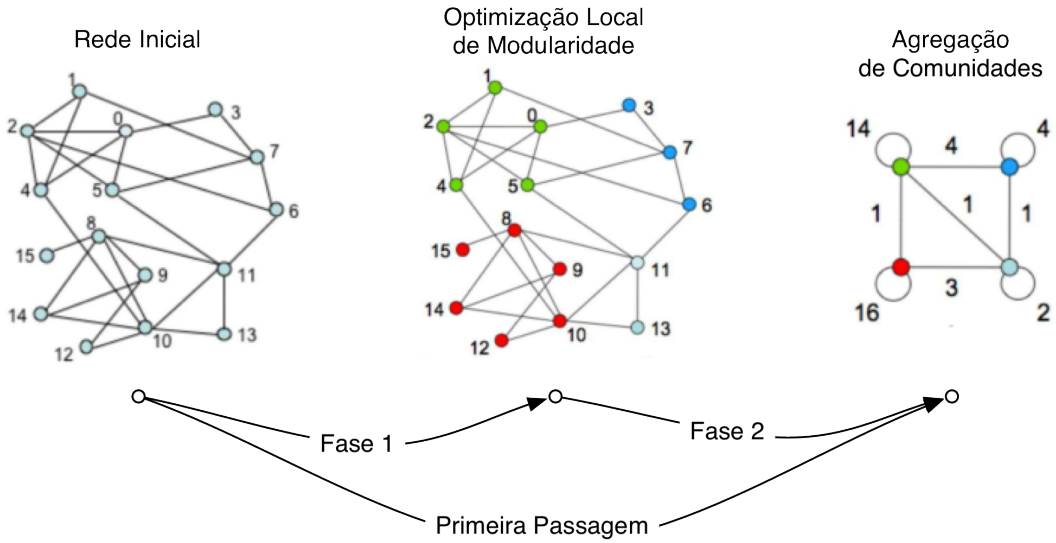
Fase 1 – Otimização local de Modularidade;

- Inicialmente cada nó corresponde a uma comunidade
- Para cada nó i, consideram-se os seus vizinhos j, avaliando o ganho de Modularidade ao retirar i da sua comunidade e colocando-o na comunidade de j. O nó i é colocado na comunidade de j se o ganho é positivo, permanece na sua comunidade caso contrário.
- O processo anterior é aplicado repetidamente até não existir melhoria no ganho, atingido um máximo para a Modularidade local (nenhum movimento individual melhora a modularidade).

Fase 2 – Construção uma nova rede de "meta-comunidades";

- Uma nova rede é construída, sendo os seus nós compostos pelas comunidades identificadas na Fase 1, em que os pesos entre os novos nós são calculados através da soma dos pesos das ligações entre os nós das duas comunidades.
- Efetuado este processo é aplicado de novo o processo da Fase 1.

Uma combinação da Fase 1 e Fase 2 designa-se por "passagem". O método aqui utilizado, reduz por construção, o número de comunidades a cada passagem, sendo grande parte do tempo consumido na computação do algoritmo gasto na primeira passagem.



As passagens decorrem então iterativamente, até que não ocorra qualquer modificação e um máximo de Modularidade é atingido. Este método apresenta algumas características típicas do *clustering* hierárquico (fase de agregação), evidenciando as estruturas hierárquicas presentes na rede.

3.4.4 Medidas de Autoridade em Redes

A utilidade das métricas de centralidade em redes tem sido estudada ao longo das últimas décadas como forma de estudar a importância de cada elemento que as constitui. As métricas de seguida apresentadas consistem em generalizações de três métricas de centralidade em redes, nomeadamente *Degree, Closeness e Betweenness Centrality* originalmente introduzidas por Linton Freeman (Freeman 1978).

A informação que cada uma destas métricas fornece para o estudo de redes pode ser resumida da seguinte forma:

- *Degree Centrality*: indica quão conetado se encontra um nó da rede, ou seja, apresenta informação relativamente à sua influencia direta para com os nós adjacentes.
- *Closeness Centrality*: indica quão distante um nó se encontra de todos os outros nós da rede, ou seja, quanto tempo demora a informação a chegar até si.
- *Betweenness Centrality*: indica a importância de um nó na distribuição de informação na rede, ou seja, nós com elevada *Betweness Centralicty* tendem a ser mediadores de informação na rede pela posição topologicamente centrada que ocupam na rede.

A generalização das métricas referidas anteriormente podem descritas da seguinte forma:

Degree Centrality: Genericamente podemos definir *Closeness Centrality* de um nó através da expressão presente na Equação 3.16:

Equação 3.16

$$D_c(i) = \sum_i x_{ij}$$

Closeness Centrality: Genericamente podemos definir *Closeness Centrality* de um nó através da expressão presente na Equação 3.17:

Equação 3.17

$$C_c(i) = \left[\sum_{j=1}^{N} d(i,j)\right]^{-1}$$

Betweenness Centrality: Genericamente podemos definir *Betweenness Centrality* de um nó através da expressão presente na Equação 3.18:

Equação 3.18

$$B_c(i) = \frac{g_{jk}(i)}{g_{jk}}$$

onde:

- $\boldsymbol{i}$ representa o nó em estudo;
- $\boldsymbol{j}$ e $\boldsymbol{k}$ correspondem a todos os outros da rede;
- $\boldsymbol{x}$ é a matriz de adjacência com $x_{ij} = 1$ se o nó i possui ligação com o nó j e $x_{ij} = 0$ caso contrário.
- $d(i,j)$ corresponde à distância mais curta entre o nó $\boldsymbol{i}$ (em estudo) e o nó $\boldsymbol{j}$.
- g_{jk} é o número de caminhos mais curtos entre o nó $\boldsymbol{j}$ (em estudo) e o nó $\boldsymbol{k}$, sendo $g_{jk}(i)$ o número destes caminhos aos quais pertence o nó $\boldsymbol{j}$.

A extensão das métricas anteriores para redes com pesos entre nós, implica as seguintes adaptações.

No caso da *Degree Centrality* a matriz de adjacência $\boldsymbol{x}$ é substituída por uma matriz $\boldsymbol{w}$ com $w_{ij} =$ "*peso da ligação entre os nós*" se o nó i possui ligação com o nó j e $w_{ij} = 0$ caso contrário.

Para as métricas *Closeness Centrality* e *Betweeness Centrality* torna-se necessário calcular as distâncias mais curtas entre os nós da rede (Wasserman and Katherine 1994). No caso da *Closeness Centrality* corresponde ao inverso da soma de todas as distâncias mais curtas do nó em estudo, para todos os outros nós atingíveis da rede, ao passo que para a *Betweeness Centrality* é calculado o número de caminhos mais curtos entre todos os outros nós da rede e dos quais o nó em estudo faz parte.

3.5 Agentes

De uma forma genérica podemos definir um agente como um programa de computador concebido para efetuar ações autónomas quando colocado num determinado ambiente (Wooldridge 2002). Estas ações podem ou não ter um objectivo maior, acima do simples desempenho das tarefas para que foi desenhado.

Alguns tipos de agentes podem mesmo possuir algum tipo de corpo físico como é o caso dos agentes robóticos (J. Liu 1999). No contexto deste trabalho vamos considerar agentes puramente baseados em software.

Se considerarmos que o comportamento do agente (conjunto de tarefas que este consegue desempenhar) é orientado para um determinado objectivo, podemos definilo como sendo um agente racional se este desempenhar "bons comportamentos" no sentido de atingir o objectivo desejado (Russell and Norvig 2010).

Tal como abordado em 2.2.1 espera-se que um agente seja capaz de se adaptar ao utilizador e às tarefas que este pretende realizar por forma a fornecer-lhe recomendações. Deste modo é necessário que o agente possua o conhecimento necessário tanto acerca dos objetivos como das preferências do utilizador. Neste contexto o processo de criação do modelo de utilizador descrito em 2.2.3 é de extrema importância, pois fornece dados essenciais ao conhecimento do agente acerca do utilizador.

De seguida são abordados um conjunto de questões relacionadas com o papel dos agentes como assistentes pessoais.

3.5.1 Agentes Baseados em Conhecimento

Em Inteligência Artificial um agente inteligente é aquele que tira partido de uma representação interna de conhecimento para tomar decisões e desempenhar tarefas.

Este agente terá que ser capaz de desempenhar tarefas, baseando as suas decisões em processos de discernimento que operam sobre a representação de conhecimento que que este possui. Este tipo de agente aprende adicionando novos dados sobre o ambiente onde opera à sua base de conhecimento, à medida que vai recolhendo informação sobre o resultado das suas ações. Também é possível que os agentes aprendam através das instruções que lhe são fornecidas pelos seres humanos (Huffman and Laird 1995). É esperado que através da aprendizagem este se torne cada vez mais autónomo.

3.5.2 Agentes e a Informação

A quantidade massiva de informação disponibilizada diariamente na Internet encerra em si não só um imenso mar de oportunidades mas também alguns problemas. Um dos problemas prende-se com a redundância de conteúdos, visto que, por exemplo, uma mesma notícia pode ser disponibilizada por um conjunto enorme de diferentes fontes. Por outro lado, muita da informação disponibilizada apresenta um baixo grau

de estruturação, tornando a sua interpretação mais difícil. Neste sentido torna-se necessário criar mecanismos de seleção de informação que possam reduzir o trabalho do utilizador na tarefa de seleção.

De seguida são apresentadas algumas questões sobre como resolver os problemas acima mencionados.

O Problema da Sobrecarga de Informação

Como já foi referido anteriormente, são produzidos diariamente um número elevadíssimo de conteúdos, disponibilizados pelas mais diversas fontes, a um ritmo elevadíssimo. Esta situação é cada vez mais evidente no contexto da Internet atual, onde os utilizadores são "bombardeados" com uma quantidade de informação não filtrada sempre que utilizam, por exemplo, um motor de pesquisa de notícias como o Google News[16]. Do ponto de vista do utilizador, a incapacidade de conseguir lidar com a quantidade de informação que lhe é entregue, distinguindo a informação relevante da não relevante, constitui o problema da "sobrecarga de informação".

Neste sentido, alguns autores (ver Maes (1994)) defendem que os agentes são uma excelente ferramenta para ultrapassar os problemas mencionados, auxiliando o utilizador a lidar com a tarefa de selecionar informação do seu interesse.

De seguida apresentam-se alguns aspetos importantes dos agentes como assistentes pessoais.

3.5.3 Agentes como Assistentes Pessoais

Um Assistente Pessoal é um tipo de programa de computador que assiste o utilizador na realização de tarefas em que terá que interagir com um determinado dispositivo.

Nos seus trabalhos em Interação Humano-Computador Negroponte (1970) propõe pela primeira vez a ideia de utilizar agentes em "interfaces de utilizador" para mediar a delegação de tarefas destinadas ao computador. O objetivo seria tornar o computador "mais humano". O seu trabalho pioneiro inspirou (Maes 1994) a desenvolver alguns protótipos de agentes capazes de auxiliar o utilizador em algumas das tarefas mais comuns quando utiliza um computador.

Esta autora é provavelmente uma das maiores defensoras da utilização de agentes como assistentes pessoais e neste contexto destacam-se dois trabalhos (ver Maes (1994)):

- *O **MAXIMS***, um assistente de email que aprende as ações que o utilizador tomou ao processar as mensagens que recebeu. Quando uma nova situação acontece, este agente consulta o conjunto de regras aprendidas, prevendo a ação do utilizador.

[16] http://news.google.pt

- O ***NewT***, um filtro de notícias Usenet[17] que prevê se uma determinada notícia será ou não escolhida para ser lida pelo utilizador. O agente era treinado através de um conjunto de exemplos pré-selecionados.

A utilização de agentes no contexto da recomendação, filtragem e extração de informação tem vindo a ganhar cada vez mais popularidade, especialmente em aplicações para a Internet. Segundo Wooldridge (2002) porque esta representa o ambiente ideal para o uso de agentes, pelas seguintes características que possui:

- permite o acesso uma diversidade de fontes de informação;
- constitui uma interface uniforme de acesso a diversas formas de recursos multimédia (texto, imagens, vídeo, etc.);
- é baseada em hipertexto, tornando possível extrair relações e associações de interesse entre os diversos documento.

[17] Usenet é uma rede de comunicação de notícias desenvolvida pela Universidade da Carolina do Norte. As mensagens são colocadas pelos utilizadores em fóruns temáticos denominados "Newsgroups".

4 ABORDAGEM

Os sistemas de recomendação tem sido utilizados com bastante sucesso na resolução de problemas de seleção de informação nos mais diversos domínios. Mais concretamente, a integração de mecanismos de recomendação em assistentes pessoais tem-se revelado extremamente proveitosa do ponto de vista do auxilio ao utilizador.

Diverso trabalho foi já realizado na implementação de sistemas capazes aliviar a sobrecarga de informação a que os utilizadores de sistemas de informação estão sujeitos (ver 2.1). Este tipo de sistema tem sido também utilizado com o objetivo de tornar a experiencia de interação com sistemas informáticos o mais natural possível, auxiliando, guiando e sugerindo possibilidades na realização das tarefas do utilizador.

Neste contexto esta tese propõe a criação de um sistema de agente pessoal baseado em conhecimento, capaz de auxiliar o utilizador a selecionar notícias de acordo com as suas preferências. O sistema extrai informação (palavras-chave) dos textos contidos nas informação recolhida (notícias) por forma a produzir uma estrutura de conhecimento a partir da frequência de coocorrência (ver 3.3.3) das palavras-chave nas notícias. A informação contida na estrutura de conhecimento é de seguida alvo de um processo de deteção de comunidades (ver 3.4.3) por forma a identificar *clusters* contendo tópicos de interesse presentes nos conteúdos recolhidos pelo sistema. Os tópicos de interesse são de seguida alvo de um processo de etiquetação por forma a identificar o tema por estes representado (ver 0).

A informação contida na estrutura de conhecimento é depois combinada com *feedback* adquirido a partir dos agentes pessoais por forma a recomendar notícias de acordo com perfil de cada utilizador. O Agentes pessoais recolhem informação correspondente às leituras e classificação das notícias por parte de cada utilizador, fornecendo então a um agente central (Agente Principal) esta informação, que a utiliza para construir o perfil de cada utilizador. A combinação do conhecimento adquirido pelo processo de categorização de conteúdos com o Modelo de cada utilizador é então utilizada para treinar classificadores (ver 3.3.2) que selecionam as notícias a entregar a cada utilizador.

O processo de identificação do perfil de utilizador é efetuado em dois momentos: numa primeira instância (aquando do arranque da aplicação móvel) são apresentados ao utilizador os tópicos de interesse mais relevantes identificados pelo processo de categorização e presentes na estrutura de conhecimento. O utilizador seleciona então o/os tópicos de interesse de sua preferência, sendo esta informação fornecida ao Agente Principal como base para o perfil de utilizador. Este processo permite implementar um mecanismo de arranque a frio (ver 2.5) que será abordado em 4.5, capaz de fornecer notícias mais próximas das suas preferências ainda antes da aprendizagem dinâmica do modelo de utilizador. Num segundo momento, o sistema prossegue então à aprendizagem dinâmica das preferências de utilizador (Modelo de

Utilizador) a partir do *feedback* fornecido na leitura das notícias, utilizando aprendizagem computacional por forma a gerar recomendações adequadas ao perfil de cada utilizador. Na aplicação implementada é solicitado explicitamente aos utilizadores que classifiquem as notícias de sua preferência, num processo experimental que será descrito em 4.6.

4.1 Arquitetura do Sistema

Dada a relativa complexidade do sistema a implementar optou-se por criar uma arquitetura modular que conferisse à aplicação a flexibilidade necessária à resolução do problema proposto. De seguida é apresentada uma descrição detalhada dos componentes que constituem o sistema proposto nesta tese, bem como uma análise de requisitos funcionais de cada um.

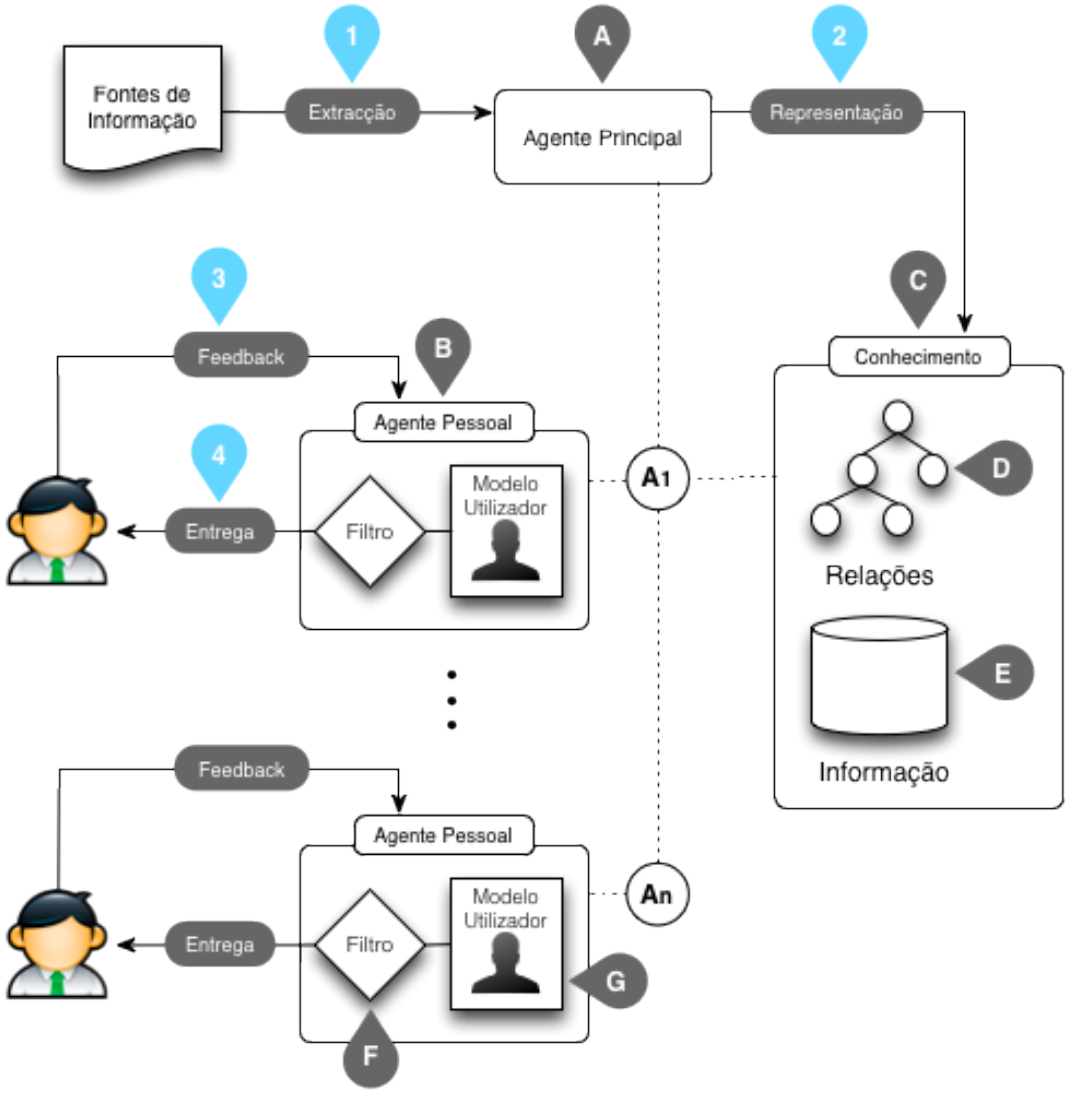


Imagem 3 – Diagrama da Arquitetura do Sistema proposto.

De uma perspectiva de alto nível, podemos identificar no sistema três módulos essenciais: o Agente Principal (A), o Agente Pessoal (B) e a Representação do Conhecimento (C). Na verdade quando falamos do Agente Pessoal, estamos de fato a referir-nos a um número inteiro n de agentes pessoais, um por cada utilizador considerado no sistema. A representação da arquitetura do sistema apresentada acima serve de referência à descrição da implementação utilizada nesta tese, que será alvo de especificação pormenorizada nas secções seguintes. No seguimento da especificação da implementação dos módulos desenvolvidos será também descrito o processo experimental de avaliação do sistema bem como dos resultados obtidos.

No seguimento da especificação da arquitetura dos componentes desenvolvidos foram identificados um conjunto de requisitos funcionais necessários ao eficaz funcionamento do sistema, que se descrevem de seguida.

4.1.1 Agente Principal

Como foi já anteriormente referido, a atividade do sistema começa pela recolha de informação a partir das diferentes fontes. Esta função é desempenhada pelo Agente Principal, que de seguida extrai palavras-chave (1) que representem os tópicos mais importantes contidos na informação. As palavras chave extraídas, são depois associadas entre si, seguindo um critério de coocorrência e inseridas (2) na estrutura de dados que representa a base de conhecimento (D). Do mesmo modo o Agente Principal armazena toda a informação recolhida na base de dados (E), etiquetando cada um dos elementos utilizando as palavras-chave recolhidas.

Requisitos do Agente Principal

- Capacidade de recolher informação a partir de diversas fontes.
- Capacidade de identificar etiquetas associadas a cada peça de informação (caso existam).
- Capacidade de extrair os tópicos principais (palavras-chave) de cada uma das peças de informação (caso não existam etiquetas associadas).
- Capacidade de incluir cada uma das etiquetas extraídas das peças de informação na representação de conhecimento.
- Capacidade de inferir corelações entre as etiquetas na forma de métricas de semelhança.
- Capacidade de armazenar as peças de informação recolhida para mais tarde disponibilizar aos Agentes Pessoais para possível entrega.
- Capacidade de receber "feedback" dos Agentes Pessoais integrando os dados recebidos na representação de conhecimento.

4.1.2 Representação de Conhecimento

O módulo de Representação do Conhecimento na estrutura de dados apresentada em (D) que representa o conhecimento sob a forma de relações entre as palavras-chave recolhidas pelo Agente Principal no processo de extração descrito na secção anterior. A estrutura de conhecimento tem uma relação próxima com a base de dados

relacional onde toda a informação anexa às notícias recolhidas é armazenada. A implementação da estrutura de conhecimento no Agente Principal, constitui o Protótipo 2 do sistema de recomendação.

Requisitos da Representação de Conhecimento

- Capacidade de construir um modelo de dados capaz incorporar na sua estrutura o conhecimento adquirido pelo Agente Principal.
- Capacidade de incorporar os objetos de conhecimento na sua estrutura: palavras-chave, relações entre palavras-chave e respetivos pesos associados.
- Capacidade de disponibilizar interfaces que permitam ao Agente Principal realizar operações de inserção, atualização, remoção e pesquisa de elementos.
- Mecanismo de identificação de *clusters* (comunidades) nos tópicos armazenados.
- Mecanismo de identificação das categorias (*labelling*) presentes nos *clusters* identificados.

4.1.3 Agentes Pessoais

Os Agentes Pessoais constituem a componente de interface com o utilizador. Estes têm a função de filtrar (F) a informação a entregar (4) a cada um dos diferentes utilizadores. Cada um dos Agentes Pessoais recolhe do utilizador informação contextual e de utilização (3) fornecendo ao sistema dados de retorno essenciais à refinação e atualização do modelo de utilizador (G). Os Agentes Pessoais têm uma relação estrita com o Agente Principal, consultando-o para tomar decisões acerca da informação a entregar a cada utilizador.

Requisitos dos Agente Pessoais

- Permitir ao utilizador a seleção prévia de tópicos de interesse por forma a construir uma base do modelo de utilizador.
- Capacidade de representar e alterar as preferências do utilizador.
- Capacidade de entregar informação seletivamente ao utilizador.
- Capacidade de recolha de feedback do utilizador (classificação de notícias).
- Capacidade de interagir com o Agente Principal para envio de "feedback".

De seguida são descritas as experiências realizadas durante o desenvolvimento de cada um dos módulos que constituem a aplicação. Em cada experiência são descritos os componentes necessários envolvidos, o seu desenvolvimento e implementação. As experiências efetuadas tiveram como objetivo o teste e avaliação do sistema à medida que foram sendo desenvolvidos os protótipos de cada componente, tendo sido efetuadas 4 experiências no total utilizando cada um dos módulos, nomeadamente o Agente Principal (Extração de Palavras-Chave e Representação de Conhecimento) e o Agente Pessoal (Aprendizagem e Modelo de Utilizador).

4.2 Experiência 1 (Extração de Palavras-Chave)

A implementação do primeiro protótipo do sistema, coincide na primeira fase de desenvolvimento do Agente Principal, nomeadamente no desenvolvimento dos módulos de agregação de notícias e extração de palavras chave. Antes de partir para a implementação foi selecionado o contexto alvo sobre o qual o sistema iria incidir. Assim, optou-se por utilizar notícias (provenientes de fontes RSS) como a principal fonte de informação, pretendendo explorar o nível considerável de estruturação da informação por estas apresentada, considerando que este tipo de informação apresenta um nível superior de cuidado na linguagem e conteúdo relativamente a outros conteúdos criados espontaneamente por utilizadores não treinados.

Relativamente à escolha das fontes noticiosas optou-se por utilizar fontes *RSS* provenientes do *Yahoo! News*[18] dado que disponibilizam um leque alargado de notícias em Inglês provenientes de várias agências noticiosas, apresentando um conjunto alargado temáticas, representando por este motivo um objeto de estudo abrangente.

4.2.1 Agregação de Notícias e extração de Palavras-Chave

O módulo de agregação de notícias implementado consiste num cliente *RSS* escrito em *Java* que monitoriza um conjunto de *feeds RSS* acerca da existência de atualizações. Aquando da publicação de novas notícias este recolhe e processa os seus conteúdos, recolhendo os elementos constituintes de cada notícia. O *parsing* dos conteúdos *RSS* foi efetuado recorrendo à biblioteca *Rome*, uma biblioteca escrita em *Java* que permite a manipulação de documentos *XML* sob a forma de objetos. Os elementos recolhidos de cada notícia são o seu título, a sua descrição e a sua data/hora de publicação. Cada uma das notícias é então guardada numa base de dados relacional (*MySQL*), para que seja efectuada posteriormente a recolha de palavras-chave a partir das notícias recolhidas.

Extração de Palavras-Chave

A implementação do módulo de extração de palavras-chave começou pela escolha da ferramentas a utilizar, tendo sido utilizadas inicialmente duas soluções distintas que passaram por uma experiência de análise das sua performance que será mais à frente descrita. Das possibilidades analisadas (ver 3.3.4) foram selecionadas duas ferramentas que se revelaram adequadas à realização deste trabalho, disponibilizando as funcionalidades necessárias à extração de palavras-chave a partir de notícias. É importante referir que na versão final deste primeiro protótipo do Agente Principal apenas o mecanismo de extração de palavras-chave que apresentou melhor performance ficou definitivamente implementado.

No módulo de extração de palavras-chave foram implementados dois métodos de extração, utilizando a *framework KEA* e a *API* do *Yahoo! Content Analysis Web*

18 http://news.yahoo.com/sitemap/

Service (YCAWS). Estas duas soluções representam dois tipos de abordagem distintas dado que o *KEA* é uma solução *standalone* e personalizável, contrariamente ao YCAWS que consiste num *Web Service* pronto a utilizar e pouco personalizável.

Na imagem seguinte é possível observar com detalhe técnico os dois módulos implementados e que constituem o primeiro protótipo do Agente Principal.

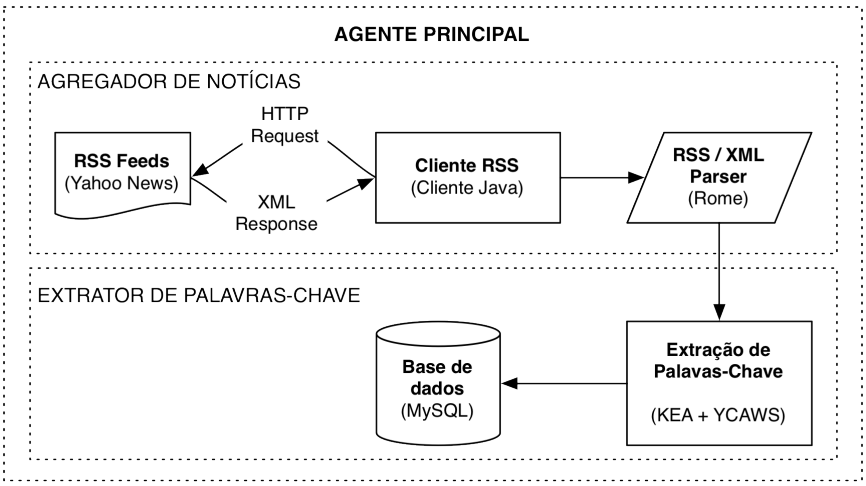


Imagem 4 – Especificação do Módulo de Agregação e Extração de Palavras-Chave.

Efetuada a implementação dos módulos acima representados procedeu-se à realização de uma experiência manual de avaliação da performance dos métodos implementados, utilizando para tal um conjunto de utilizadores que procederam aos testes de avaliação que será de seguida descrito.

Os objetivos da experiencia realizada são os seguintes:

- Avaliar a precisão, abrangência e F_1 (*f-measure*) de cada um dos métodos implementados.
- Quantificar o número máximo de termos que constituem uma palavra-chave adequada à etiquetação de notícias.
- Quantificar o número médio de palavras-chave utilizados para caracterizar uma notícia.

Para levar a cabo esta experiência foi criada uma interface de testes (ver Imagem 5) que permitiu aos utilizadores proceder à avaliação do processo de extração. A interface de testes consiste numa aplicação *Web*, desenvolvida utilizando as linguagens *Php*, *Javascript* e *SQL* que comunicava com o Agente Pessoal por forma a obter dados de teste a partir do módulo de extração de palavras-chave.

4.2.2 Configuração da Experiência

Na Imagem 5 pode ser observada a informação disponibilizada aos utilizadores durante a avaliação. Ao utilizador é apresentado um número identificador da notícia (caso de teste), o título da notícia e o seu resumo descritivo. Logo abaixo são apresentadas nas quatro primeiras colunas (a partir da esquerda) o resultado do

processo de extração, correspondendo cada uma das colunas ao conjunto de palavras extraídas por cada uma das configurações dos mecanismos de extração utilizados. A quinta e última coluna apresentada na interface da aplicação de avaliação permite a inserção de palavras-chave por parte dos utilizadores até um máximo de dez.

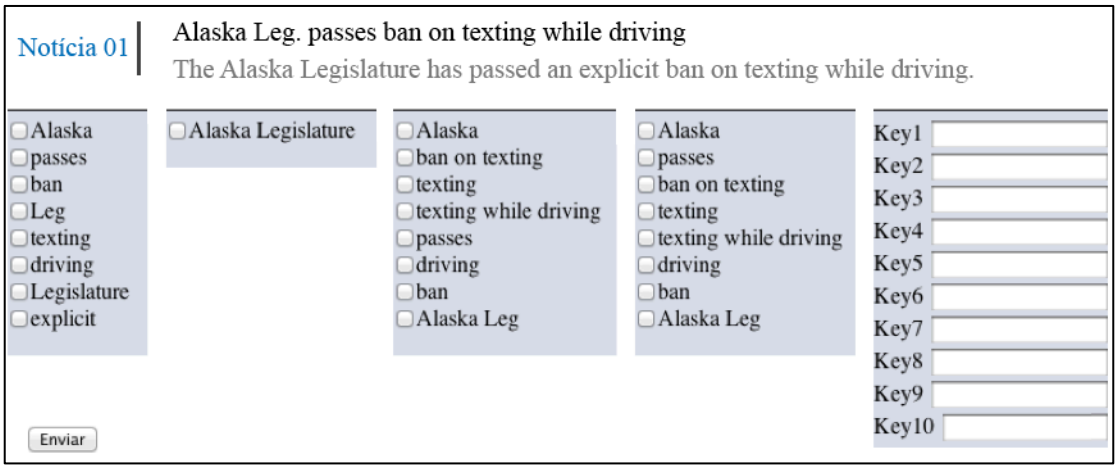


Imagem 5 - Interface de Avaliação da Extração de Palavras-chave.

Como podemos observar na tabela Tabela 5, foram implementadas três variações do KEA e uma configuração do YCAWS. No caso do KEA foi feito variar o número máximo de termos por palavra-chave, limitando o número de palavras-chave extraídas ao máximo de dez. No caso do YCAWS e dada a impossibilidade de fazer variar o número máximo de termos por palavra-chave apenas foi limitado o número de palavras-chave para dez resultados, em coerência com as configurações do KEA. Estas configurações vão de encontro aos objetivos já identificados anteriormente para esta experiência.

Na tabela seguinte podemos observar as configurações utilizadas na experiência de avaliação da extração:

	KEA 2	**KEA 3**	**KEA 4**	**YCAWS**
Nº de Termos por Palavras-Chave	≤ 2	≤ 3	≤ 4	Variável
Nº de Palavras-Chave	≤ 10	≤ 10	≤ 10	≤ 10

Tabela 5 - Configurações do Mecanismo de Extração de Palavras-Chave.

4.2.3 Treino do Algoritmo de Extração do KEA

Antes de proceder aos testes experimentais, foi necessário treinar a implementação do *KEA* dado que este algoritmo utiliza aprendizagem computacional, aprendendo a partir de exemplos. O treino necessário ao funcionamento do *KEA* consiste num conjunto de textos manualmente classificados, às quais são associados as palavras-chave correspondentes. O algoritmo utiliza então este recurso para criar um modelo estatístico que pode depois ser guardado e reutilizado no contexto de uma aplicação

real. Este fator implicou a criação de três conjuntos de treino, um por cada variação implementada.

No treino do *KEA* foram utilizados três conjuntos de notícias manualmente criados, seguindo o critério do número máximo de termos por palavras-chave. No total foram utilizados 480 notícias escritas em Inglês (160 por configuração) provenientes de diversas fontes e categorias temáticas por forma a generalizar o treino do algoritmo de extração. Na configuração do *KEA* optou-se por utilizar indexação livre de palavras-chave, isto é, sem a utilização de uma gramática (dicionário) controlada. Deste modo o algoritmo decide livremente que palavras utilizar, a partir do texto das notícias.

4.2.4 Procedimento do Processo de Avaliação

A avaliação contou com a participação de nove utilizadores voluntários, divididos em três grupos de três pessoas por grupo. A cada grupo foi atribuído um conjunto distinto de 45 notícias de modo que cada notícia foi assim classificada por três utilizadores distintos. Aos utilizadores foi solicitado que utilizando a aplicação de avaliação, selecionassem em cada coluna (lista para cada configuração) as palavras-chave que consideravam mais relevantes/corretas para categorizar cada notícia. As escolhas eram efetuadas de forma independente para cada coluna/configuração por forma a avaliar independentemente cada conjunto de palavras-chave extraídas por cada configuração. Foi ainda solicitado aos utilizadores que caso considerassem existir palavras-chave que deveriam fazer parte do conjunto de palavras-chave recolhidas e não constassem em nenhuma lista, as referissem na avaliação, utilizando para tal os dez campos disponíveis na última coluna da interface da aplicação de avaliação.

No total foram avaliadas 135 notícias, sendo realizadas três avaliações distintas por notícia o que corresponde a um total de 405 avaliações.

4.2.5 Resultados e Conclusões

Por cada avaliação foi gerado um registo contendo as palavras assinaladas em cada coluna, bem como as palavras introduzidas manualmente pelos utilizadores. A partir deste registo foi efetuada uma análise estatística de todo o conjunto das avaliações por forma a verificar a performance de cada configuração. Mais concretamente foram calculadas a precisão, a abrangência e F1 de cada uma das implementações.

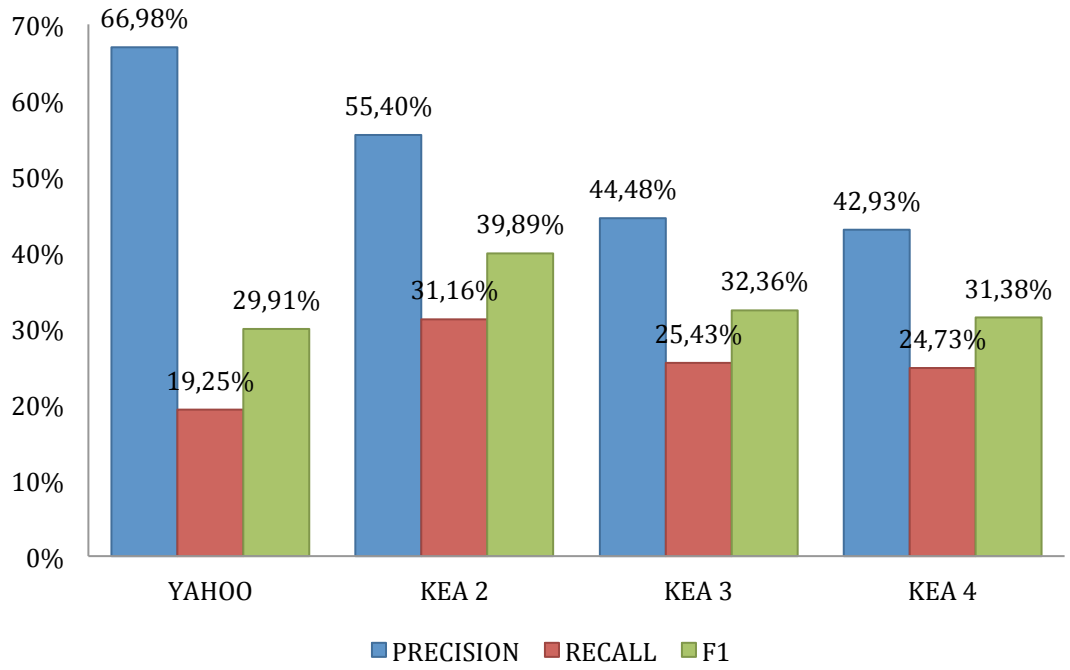
Na tabela seguinte podemos observar os resultados da análise estatística efetuada aos resultados da avaliação manual de cada um dos métodos/configurações de extração:

	PRECISION	RECALL	F1
YAHOO	66,98%	19,25%	29,91%
KEA 2	55,40%	31,16%	39,89%
KEA 3	44,48%	25,43%	32,36%
KEA 4	42,93%	24,73%	31,38%

Tabela 6 - Resultados da avaliação manual da Extração de Palavras-Chave.

Da análise estatística dos resultados da avaliação foi possível concluir que a melhor abordagem consiste em utilizar o *KEA* na sua configuração *KEA 2*, ou seja, treinado e configurado para extrair palavras-chave com o máximo de dois termos. Esta configuração revelou maior equilíbrio relativamente aos parâmetros de precisão e abrangência, como pode ser observado pelo resultado obtido para o F1 de cerca de 40%. Este resultado permite cumprir o primeiro objetivo desta experiência, que consistia em selecionar a implementação que apresentasse melhor performance na extração de palavras-chave.

No gráfico seguinte podemos observar uma comparação dos resultados obtidos para cada uma das implementações:



Relativamente ao segundo objetivo desta experiência, avaliar o número de palavras-chave necessárias para classificar cada notícia, procedeu-se ao calculo do número médio de palavras-chave distintas selecionadas por cada utilizador. Este cálculo foi efetuado para a implementação que obteve melhor performance, o KEA 2.

Na tabela seguinte podemos observar a distribuição do número médio de palavras-chave para a configuração KEA 2.

KEA 2	Número Palavras	Número de Artigos	Média por Artigo
User1	275	45	6,11
User2	265	45	5,89
User3	162	45	3,60
User4	169	45	3,76
User5	219	45	4,87
User6	180	45	4,00
User7	147	45	3,27
User8	168	45	3,73

User9	263	45	5,84
TOTAL	1848	405	4,56

Tabela 7 – Dados estatísticos das palavras-chave selecionas por utilizador.

O valor obtido para o número médio de palavras-chave selecionadas foi de 4,56 ou seja, aproximadamente cinco palavras-chave. Deste modo, 5 foi o valor utilizado na implementação definitiva do extrator de palavras-chave, recolhendo assim cinco palavras-chave por cada notícia, com um tamanho máximo de dois termos por palavra-chave.

4.3 Experiência 2 (Representação de Conhecimento)

A implementação da Representação de Conhecimento compreende três componentes essenciais: a estrutura de conhecimento na forma de grafo, o algoritmo de identificação de comunidades e o mecanismo de identificação de categorias. De seguida são descritos os processos de implementação, integração e análise de cada um dos componentes.

4.3.1 Implementação da Estrutura de Conhecimento

Relativamente à estrutura que suporta a representação do conhecimento optou-se pela sua implementação através da utilização de uma base de dados de grafos, por se enquadrar tecnologicamente na estrutura idealizada para representar o conhecimento (grafo).

De entre as diversas possibilidades analisadas o *Neo4j* (ver 3.2.5) foi a opção escolhida pelas razões de seguida apresentadas:

- Permite a sua utilização não só na forma de base de dados embebida mas também a sua implementação como servidor autónomo que pode ser acedido remotamente através de uma *API* disponível para várias linguagens.
- Possui uma implementação da sua *API* em *Java*, linguagem preferida na implementação deste trabalho por questões de integração.
- A sua *API* apresenta um conjunto de funcionalidades que permitem com eficiência implementar todo o tipo de operações sobre grafos, nomeadamente procura, inserção, atualização e remoção de nós e arestas.
- É uma base de dados com boa maturidade, altamente escalável e que apresenta uma boa documentação.
- Possui uma versão livre e em código aberto (*Community Edition*).

4.3.2 Configuração do Neo4J

No desenvolvimento do módulo de conhecimento utilizou-se o *Neo4J Community Edition* (versão 1.6) na sua vertente embebida. O módulo de representação de conhecimento foi incorporando na estrutura do Agente Principal, já desenvolvida para o protótipo anterior da aplicação (Protótipo 1).

As funcionalidades de base de dados embebida do *Neo4J* foram implementadas no Agente Principal através da sua integração numa componente de servidor construída por forma a servir os seguintes propósitos:

- Integração do módulo de extração de palavras-chave com a estrutura de conhecimento.
- Disponibilização de uma interface que permite a criação e atualização dos nós do grafo (que contêm as palavras-chave e os seus atributos) e dos pesos das arestas (frequência de coocorrência entre palavras-chave).
- Disponibilização de um *Web Service* que permitirá à aplicação móvel desenvolvida no protótipo final (Agente Pessoal), comunicar com a estrutura de conhecimento e a base de dados de notícias.

Na Imagem 6 podemos observar a estrutura do grafo implementado no *Neo4J*:

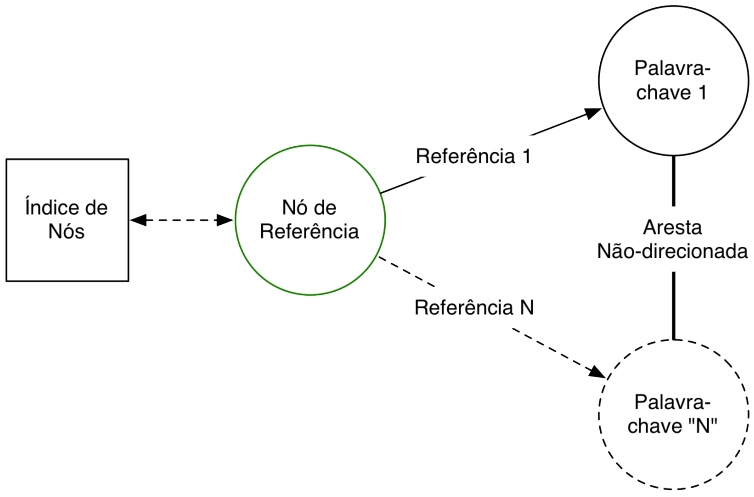


Imagem 6 – Estrutura Interna do Grafo Implementado no Neo4J.

A estrutura criada no *Neo4J* implementa um grafo não-direcionado, representando a coocorrência de cada um dos pares de palavras-chave de forma independente.

Internamente o Neo4J implementa um nó adicional ao grafo que funciona como "nó de referência". Utilizando o conjunto dos nós referenciados pelo "nó de referência" foi também implementado um indicie de nós que permite a procura rápida de nós no grafo, utilizando os atributos associados, nomeadamente a palavra-chave que cada nó representa.

O atributos criados em cada um dos nós bem como nas arestas do grafo implementado podem ser observados na figura seguinte:

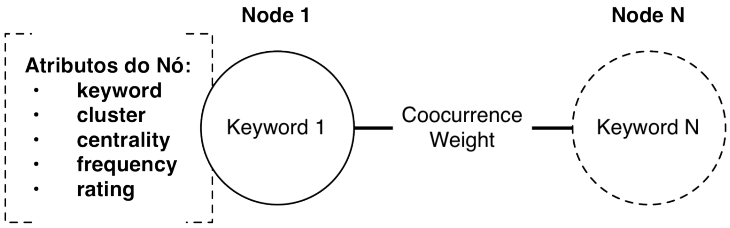


Imagem 7 – Representação de Nós, Arestas e Atributos no Neo4J.

- ***Keyword*:** palavra-chave contida no nó;
- ***Cluster*:** identificação do *cluster* atribuído ao nó;
- ***Frequency*:** frequência ocorrência da palavra-chave;
- ***Rating*:** medida de popularidade da palavra-chave (baseada no *feedback* dos utilizadores);
- ***Authority*:** medida de autoridade da palavra-chave calculada no processo de *clustering* (ver 0);
- ***Co-occurrence Weight*:** peso da relação entre palavras-chave, baseado na sua coocorrência no texto das notícias.

O peso de coocorrência foi normalizado para uma melhor integração no processo de *clustering* a realizar.

4.3.3 Identificação de Tópicos de Interesse

Um dos passos críticos neste trabalho passou por criar um mecanismo que determinasse os tópicos presentes no conjunto de notícias recolhido, identificando conjuntos de palavras-chave que representassem cada uma das temáticas. Ao processo de agregação de um conjunto de palavras-chave que possuem algum tipo de afinidade designamos por "*tag clustering*".

Existem um conjunto de técnicas para levar a cabo este processo que foram já identificadas anteriormente (ver 3.4) e que foram objeto de discussão por forma a determinar qual dos métodos melhor se adequaria no contexto e objetivos deste trabalho.

Particionamento do Grafo vs. Detecção de Comunidades

Uma das premissas deste trabalho tem que ver com a capacidade de lidar com um número indeterminado de fontes de informação, isto porque numa situação real de implementação cada utilizador pode selecionar as fontes informativas que são do seu agrado. Este fator remete para uma situação em que o sistema terá que ser capaz de identificar temáticas presentes nos conteúdos selecionados, independentemente das fontes selecionadas. Deste modo torna-se necessário que o método utilizado seja capaz de descobrir os diferentes temas presentes desconhecendo à partida a quantidade de temas esperados. Devido à condição referida anteriormente, torna-se evidente que os métodos tradicionais de *clustering*, nomeadamente os que dependem puramente de abordagens baseadas em *k-means,* nos quais é necessário definir à partida um número desejado de partições não se apresentam como uma opção válida.

Abordagens mais recentes sugerem a utilização de uma função de qualidade por forma a identificar *clusters* que otimizam ou cumprem os requisitos impostos pela função utilizada. Neste campo o algoritmo de *Girvan-Newman* (Girvan and M. E. J. Newman 2002) é uma das abordagens mais revolucionárias na identificação de comunidades em grafos. Apesar de ter sido criado com base no estudo de redes de natureza social, desde a sua criação que diversas variantes têm surgido com aplicações nas mais diversas áreas como a biologia (Luo and Scheuermann 2006) ou até mesmo a neurociência (Zhuo et al. 2011).

Efetuada esta análise de hipóteses foi selecionada uma implementação recente do algoritmo de *Girvan-Newman* que será descrita de seguida.

Algoritmo de Deteção de Comunidades

Uma das implementações do algoritmo de *Girvan-Newman* mais interessantes analisadas durante a pesquisa para este trabalho foi o algoritmo *"fast unfolding of communities in large networks"* (Blondel et al. 2008) que se apresentou como uma boa solução dada a sua elevada performance e escalabilidade em comparação com os métodos de *clustering* tradicionais. Este algoritmo, para além de revelar as estruturas comunitárias presentes nas redes, também expõe a estrutura hierárquica dos elementos que as compõe pois de certa forma o algoritmo propõe uma aglomeração iterativa e hierárquica durante a formação dos clusters (ver 3.4.3).

De entre as ferramentas estudadas para a realização desta tarefa e que podem ser encontradas em detalhe em 3.3.4, foram testadas duas que implementam o algoritmo *"fast unfolding of communities in large networks"*: são estas o *NetworkX* e o *Gephi*. Os testes preliminares de deteção de comunidades foram inicialmente implementados utilizando o *NetworkX*, servindo esta primeira abordagem ao problema para aferir a adequabilidade do algoritmo à aplicação. Contudo surgiu a necessidade de estudar mais profundamente os resultados obtidos, estudo este que passava pela visualização das partições encontradas por forma a aferir a qualidade das mesmas. Apesar de o *NetworkX* implementar funções básicas de visualização de grafos, este carece de alguma flexibilidade de manipulação e análise estatística. Verificou-se então que o *Gephi* neste campo se apresentou mais adequado pelas funções de análise disponibilizadas. Outro fator relevante é o fato de o *Gephi* permitir a sua implementação integrada em qualquer sistema utilizando a sua *API Java*, mas também a sua utilização *standalone* como ferramenta de análise de grafos

Implementação do Processo de Deteção de Comunidades

Antes de iniciar o processo de detecção de comunidades foi necessário reunir um conjunto de notícias e respectivas palavras-chave associadas que constituísse um objeto de estudo interessante. Deste modo, fez-se correr o protótipo do Agente Principal implementado anteriormente e descrito na secção 4.2.1 durante cerca de duas semanas, tendo sido recolhidas aproximadamente 6000 notícias. Se considerarmos que o módulo de extração recolhe até 5 palavras-chave de cada notícia temos um universo potencial de 30000 palavras-chave. No entanto verifica-se que

existe grande sobreposição nas palavras-chave extraídas, ou seja, uma grande maioria das notícias recolhidas partilha palavras-chave. O número total de palavras-chave extraídas e inseridas no grafo implementado situou-se perto das 5000 (≈17% do potencial valor), sendo que este foi o universo de palavras chave sujeito ao processo de *clustering* descrito de seguida.

O processo de identificação de comunidades passou inicialmente pela transladação para o *Gephi* do grafo de palavras-chave implementado no *Neo4J*. O grafo foi então exporto na forma de uma lista de nós e arestas com os respectivos pesos de coocorrência associados e importados utilizando a *API* do *Gephi* por forma a construir uma representação do grafo a analisar.

De seguida foi utilizada a função *Modularity* disponibilizada pela *API* do *Gephi* e que implementa o algoritmo de detecção de comunidades, algoritmo que devolve uma estrutura contento todos os conjuntos de nós pertencentes a uma mesma classe de Modularidade, ou seja, o conjunto das comunidades identificadas no grafo. Por último foi utilizada a função *Partitioning* também disponibilizada pela *API* do *Gephi* por forma a particionar o grafo nos seus subconjuntos. Como parâmetros de particionamento esta função recebe os nós devidamente classificados por classes de Modularidade, devolvendo conjuntos de nós contendo cada comunidade.

Na Imagem 8 podemos observar uma representação do grafo após a aplicação do algoritmo de *clustering* sobre a rede de palavras-chave:

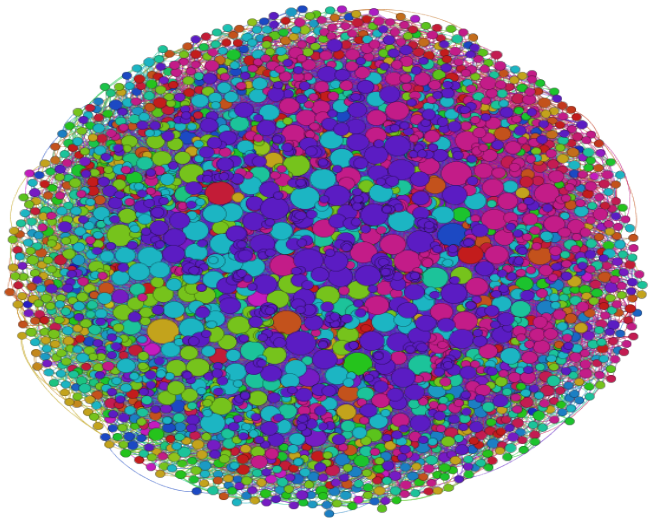


Imagem 8 - Visualização do grafo após identificação de comunidades.

Para proceder à criação da visualização apresentada na Imagem 8, o grafo foi colorido de acordo com as classes definidas pelo particionando, atribuindo uma cor distinta a cada classe.

A Imagem 9 apresenta a distribuição do número de nós por cada comunidade identificada:

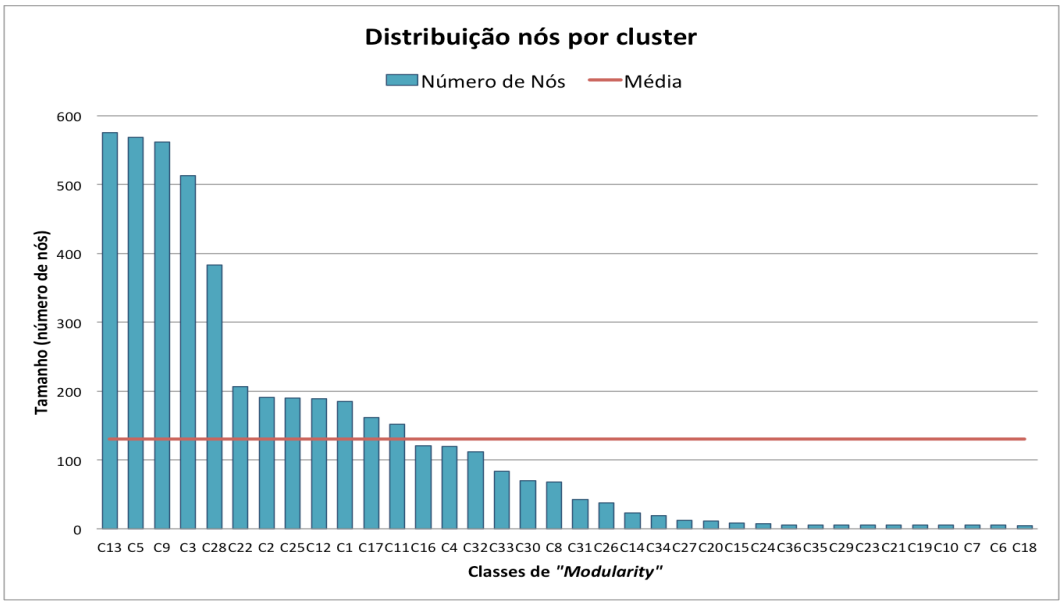


Imagem 9 – Gráfico da distribuição de nós por comunidade (cluster).

Como podemos observar no gráfico apresentado na figura anterior foram identificadas 35 comunidades (*clusters)*, sendo que os tamanhos dos *clusters* possuem uma distribuição relativamente dispersa. Os 12 clusters mais significativos, os que contêm um número de nós acima da média, representam cerca de 83% do número de nós do grafo, sendo os restantes 17% correspondentes a comunidades de significado reduzido.

De seguida apresentam-se alguns valores estatísticos relativos à distribuição de nós:

- **Número de Nós** = 4665 nós;
- **Maior cluster** = 575 nós;
- **Menor cluster** = 2 nós;
- **Média por cluster** ≈ 130 nós.

Numa análise manual a cada uma das comunidades detetadas no grafo, verificou-se que as comunidades que apresentavam um número de nós abaixo da média não constituíam informação relevante suficiente para eventualmente identificar tópicos de interesse, ou apresentavam informação relativamente ambígua. Nesse sentido, como será observado mais adiante, nas experiências a realizar com o protótipo final da estrutura de conhecimento e do sistema de recomendação, serão consideradas comunidades que apresentam um número de nós um acima da média por forma a garantir uma maior abrangência no número de palavras-chave associadas a cada tópico de interesse.

Categorização de Tópicos de Interesse

Como foi referido na secção anterior uma das características do algoritmo *"fast unfolding of communities in large networks"* (Blondel et al. 2008) e que foi utilizado neste trabalho, prende-se com o fato de este revelar a estrutura hierárquica dos elementos contidos nos *clusters*. Deste modo espera-se que em cada um dos *clusters* (comunidades) exista uma hierarquia de valor e autoridade para cada elemento que

permite identificar os mais relevantes, que podem servir como identificadores das temáticas presentes em cada comunidade (categorias).

Relativamente à identificação das palavras-chave que servem de identificador da categoria a opção tomada neste trabalho recaiu sobre o cálculo da *Betweenness Centrality* de cada palavra-chave dentro de cada um dos *clusters*, selecionando a palavra-chave com maior valor para esta métrica como identificador da categoria. O cálculo da *Betweenness Centrality* foi efetuado recorrendo à *package Statistics* disponibilizada pela *API do Gephi*.

Na Imagem 10 podemos observar o *cluster* correspondente à categoria "Desporto" no qual foi identificada a palavra-chave "GAME" como seu identificador:

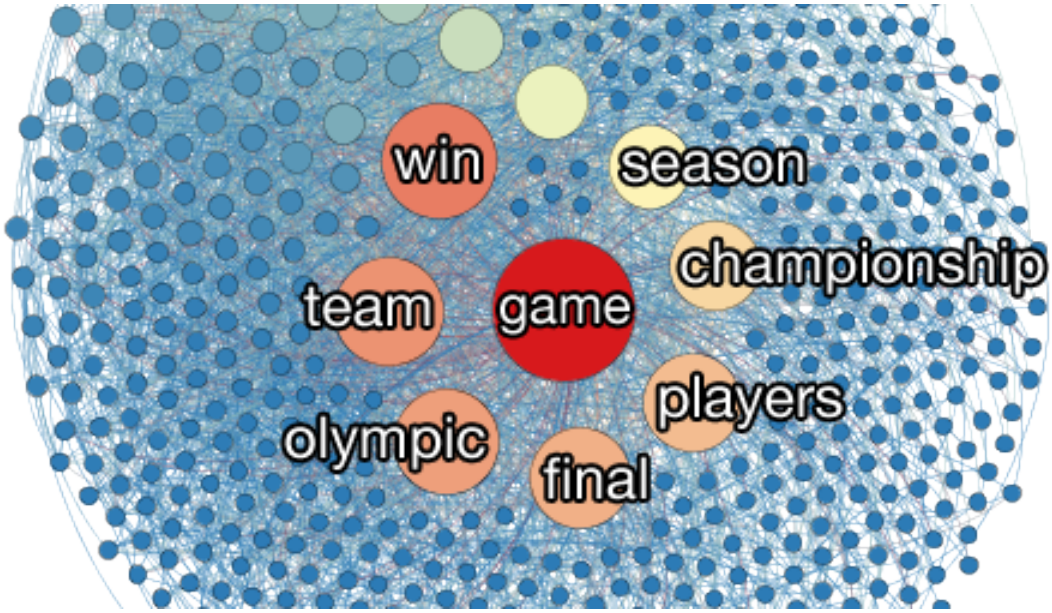


Imagem 10 - Visualização da Comunidade "Game".

O mesmo processo de identificação das palavra-chave de maior centralidade (*Betweenness Centrality*) foi efetuado para cada um dos *clusters* de maior relevância, por questões de adequação à aplicação final a produzir e por estes apresentarem maior abrangência temática. Como foi já oportunamente referido foram apenas considerados *clusters* com um número de nós acima da média (ver gráfico na Imagem 9), o que corresponde a um total de 12 *clusters*.

Na Tabela 8 são apresentadas as palavras-chave, identificadoras das categorias referentes a cada um dos 12 *clusters* mais relevantes, bem como a avaliação manual de cada um:

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
Sistema	scientists	broadway	political	market
Manual	Science	Entertainement	Politics	Economy

	Cluster 5	**Cluster 6**	**Cluster 7**	**Cluster 8**
Sistema	health	study	disease	patients
Manual	Health	Research	Research	Health

	Cluster 9	**Cluster 10**	**Cluster 11**	**Cluster 12**
Sistema	internet	technology	world	game
Manual	Internet	Technology	World	Sports

Tabela 8 – Identificação de Tópicos de Interesse Associados a cada um dos clusters mais significativos.

Efetuando uma correspondência entre os clusters identificados e as palavras-chave que os representam (categorias) foi possível criar uma visualização do grafo, que corresponde ao resultado final do processo de identificação de tópicos de interesse que pode ser observado na Imagem 11.

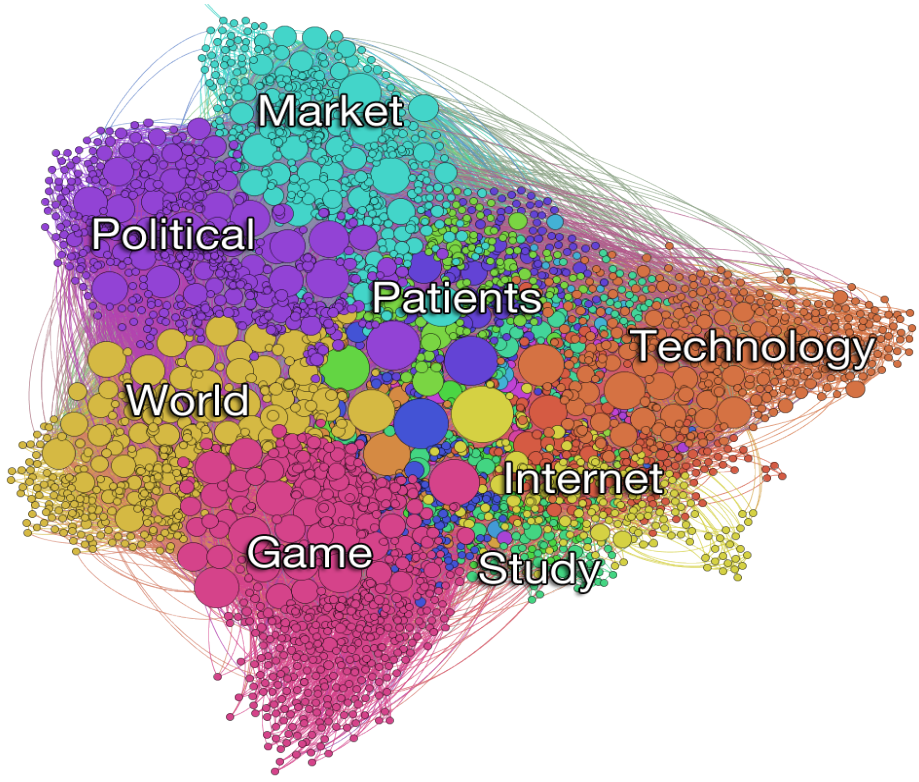


Imagem 11 - Visualização de tópicos de interesse no grafo de comunidades.

Por uma questão de apresentação e espaço neste documento o grafo representado na Imagem 11 foi manipulado aplicando um algoritmo de reordenamento topológico de forma a realçar alguns dos tópicos mais relevantes.

4.3.4 Resultados e Conclusões

Terminado o processo de *clustering* de palavras-chave e analisados os resultados foi possível constatar que o algoritmo *"fast unfolding of communities in large networks"* (Blondel et al. 2008) se revelou bastante eficaz. Foi possível identificar um conjunto razoável de tópicos de interesse, sendo que as comunidades mais significativas apresentam um leque de palavras-chave bastante representativas dos tópicos que representam. Numa análise efetuada aos 12 *clusters* mais significativos foi possível estimar o grau de relação das palavras-chave em cada *cluster* com o tópico que representam. A análise das palavras chave foi realizada de forma manual para as 50 palavras-chave com maior valor de *Betweenness Centrality* em cada *cluster*, tendo sido atribuída uma classificação binária de enquadramento ou não no tópico do *cluster* a que pertencem. Os resultados desta análise podem ser observados no gráfico presente na Imagem 12.

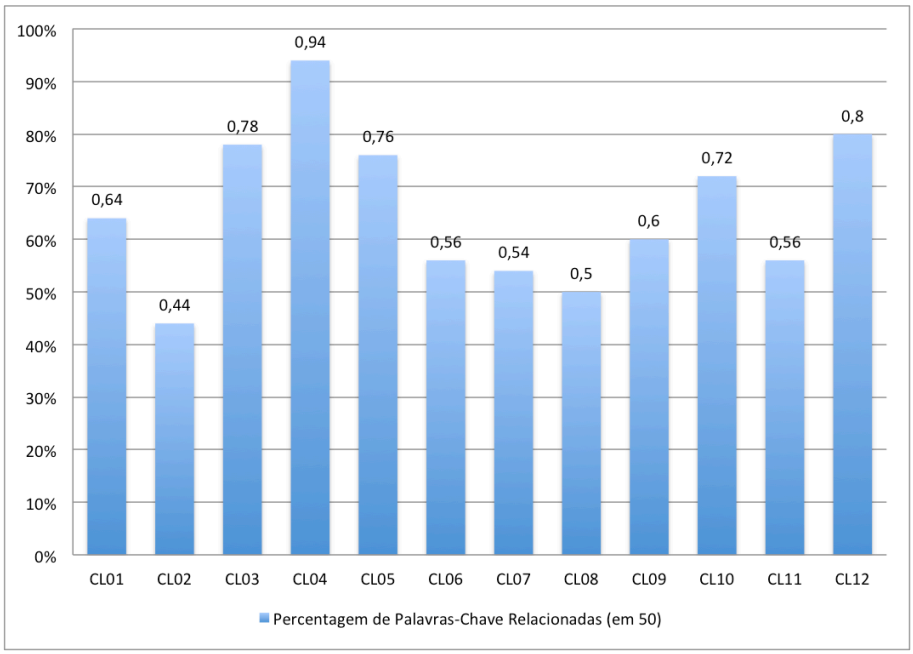


Imagem 12 - Gráfico da relação de palavras-chave com o tópico de cada cluster.

A partir da análise dos resultados presentes no gráfico anterior podemos estimar que em média cerca de 65,33% das palavras-chave se relacionam com o tópico identificado para o *cluster* a que pertencem. Outro importante a reter, é o fato de o número de nós pertencentes aos 12 clusters mais significativos representarem 83% dos nós do grafo. Pode-se assim concluir que os resultados obtidos pelo processo de *clustering* apresentam uma abrangência bastante significativa.

Concluído o processo de identificação e análise dos *clusters* procedeu-se à integração dos resultados obtidos na estrutura previamente implementada no Neo4J. Os nós

contendo cada palavra-chave foram etiquetados com o identificador do cluster a que pertencem, bem como com o seu valor de centralidade no cluster. Na estrutura de conhecimento foram apenas incorporados os 12 *clusters* mais significativos, tendo sido atribuído o identificador *"Cluster 0"* a todas as palavras-chave que pertencem a *clusters* pouco significativos (com número de nós inferior à média). Numa implementação dinâmica do sistema de *clustering* os nós contendo palavras-chave pertencentes ao *"Cluster 0"* seriam gradualmente integrados num dos *clusters* representativos do tema a que pertencem.

Deste modo o grafo de palavras-chave fica povoado com os dados necessários ao seu funcionamento como estrutura de conhecimento, integrada na estrutura do Agente Principal e disponível para consulta pela rede de agentes.

Com a integração do conhecimento no Agente Principal fica completo o Protótipo 2 do sistema de recomendação. Na Imagem 13 podemos observar a integração do conhecimento na aplicação do Agente Principal.

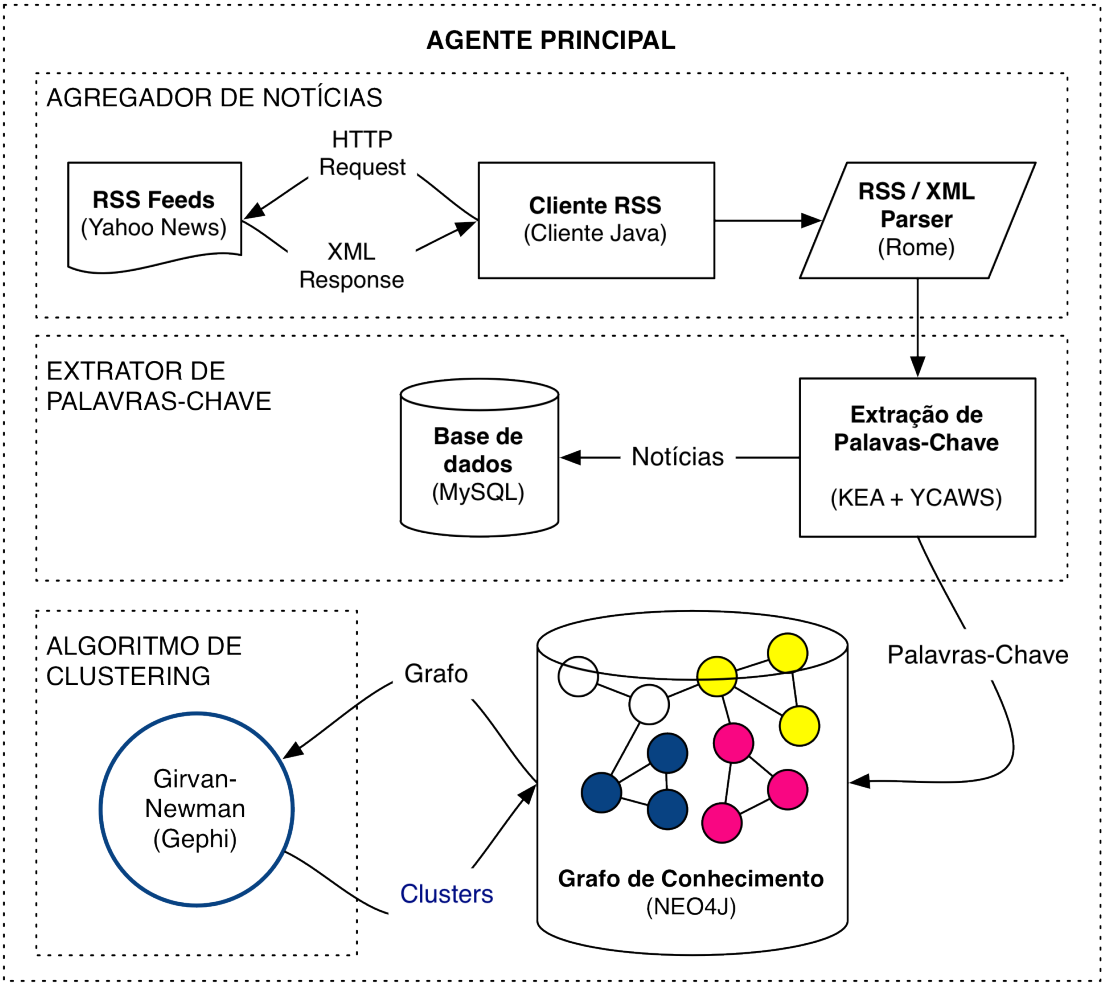


Imagem 13 - Arquitetura do Agente Principal após integração do Conhecimento.

4.4 Aprendizagem e Modelo de Utilizador

O desenvolvimento dos Agentes Pessoais e a sua ligação com o Agente Principal constitui a última fase de implementação do sistema de recomendação. O Agente Pessoal tem a sua face visível para o utilizador na forma de uma aplicação desenvolvida na plataforma *Google Android*. Através desta aplicação os utilizadores efetuam a leitura de notícias de acordo com as suas preferências, tendo a possibilidade de as classificar consoante o seu interesse. Este mecanismo de recolha de *feedback* é essencial à identificação do Modelo de Utilizador, pois esta informação é utilizada e combinada com o conhecimento, no treino dos algoritmos de recomendação.

Nas secções seguintes encontra-se descrita a implementação dos seguintes componentes:

- A implementação da Rede de Agentes.
- A representação do Modelo de Utilizador como conjunto de instâncias de treino dos classificadores.
- O módulo de classificação onde foram implementados os algoritmos de aprendizagem utilizados pelo Agente Pessoal.

O treino dos classificadores e a avaliação do seu desempenho na aprendizagem do Modelo de Utilizador são descritos nas experiências 3 e 4 (ver 4.5 e 4.6 respetivamente).

4.4.1 Implementação da Rede de Agentes

Como foi referido anteriormente a aplicação desenvolvida implementa um sistema de Agente Pessoal. Este Agente Pessoal é composto por uma interface móvel, a representação do Modelo de Utilizador, bem como o módulo de classificação onde foi implementada a aprendizagem. O Agente Pessoal recolhe dados relativos às preferências e classificação de notícias por parte dos utilizadores, combinando esta informação com a informação contida na estrutura de conhecimento para gerar recomendações. Esta combinação do conhecimento com o *feedback* dos utilizadores é utilizada no treino de classificadores que possam aprender o modelo de utilizador e classificar notícias autonomamente.

Um aspeto muito importante a realçar prende-se com o fato de parte da arquitetura do Agente Pessoal ter sido implementada ao nível aplicacional, junto do Agente Principal. Ou seja, foi implementada uma abordagem híbrida do Agente Pessoal, fugindo um pouco às abordagens tradicionais, nas quais o Agente Pessoal possui do lado do servidor ou do lado do cliente, todos os mecanismo necessários ao seu funcionamento autónomo. Foi tomada esta opção por se concluir que o dispositivos móveis não apresentam ainda a capacidade de processamento necessário à implementação do módulo de classificação, i.e., ainda não é possível tirar partido dos vários algoritmos disponibilizados pelo *Weka* nos dispositivos móveis.

Na Imagem 14 está apresentada a arquitetura dos componentes desenvolvidos nesta última fase.

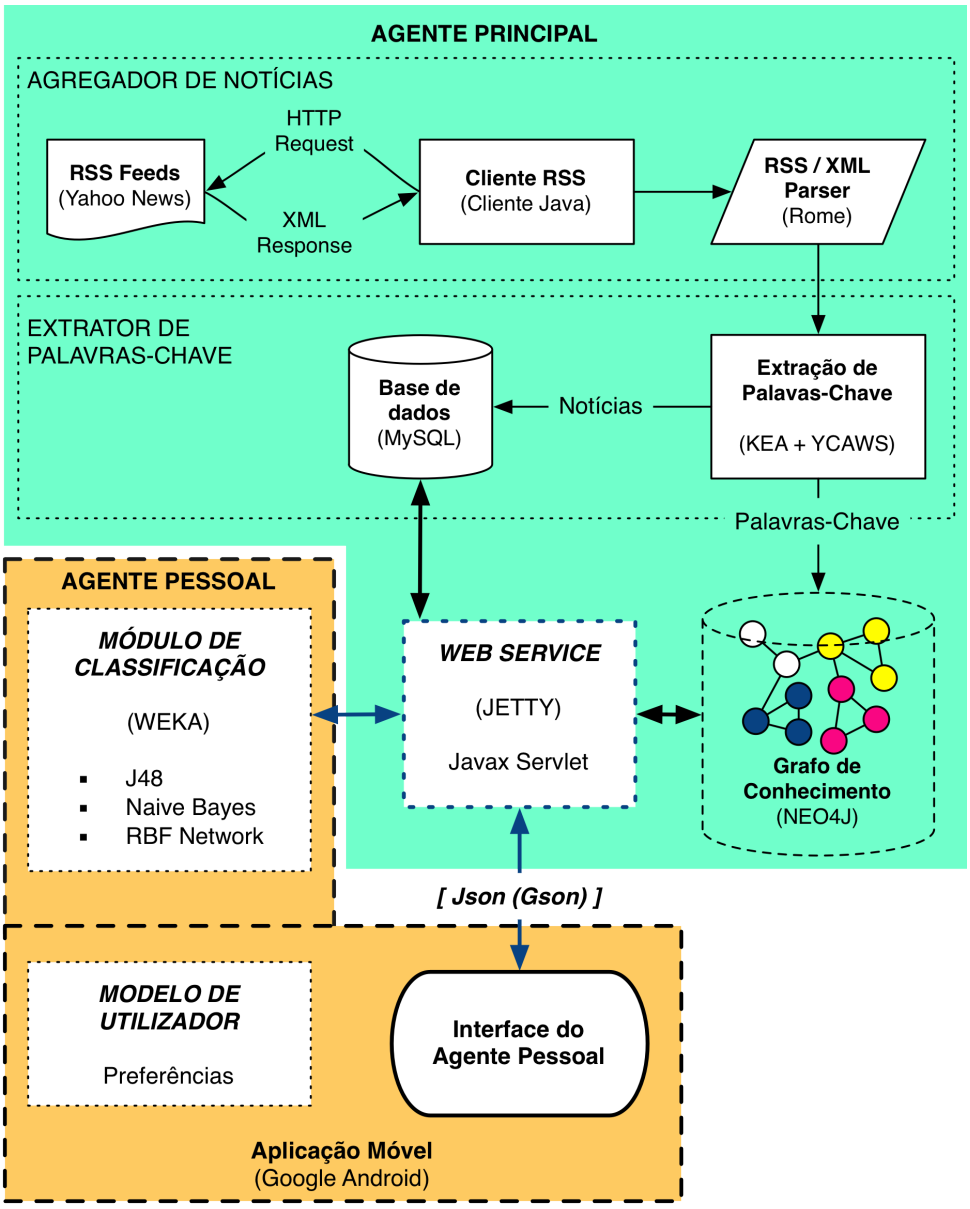


Imagem 14 – Arquitetura da rede de Agentes.

As questões identificadas anteriormente relativas às capacidade de processamento dos dispositivos móveis, levaram a que parte do processamento fosse efetuado do lado do Agente Principal. Deste modo, foi implementado um módulo servidor do lado do Agente Principal, que serve de camada de integração dos componentes do Agente Pessoal. Em suma, o Agente Pessoal funciona de uma forma híbrida, acedendo aos algoritmos de aprendizagem através do servidor, tornando também transparente a consulta da estrutura de conhecimento, implementada no Agente Principal.

Implementação do Web Service

O servidor foi implementado utilizando tecnologia Java, mais concretamente recorrendo ao software de servidor *Jetty*. O *Jetty* disponibiliza um servidor *HTTP*, bem como uma implementação do *javax.servlet* que permitem implementar *Web Services* na forma de *Servlets*.

Como se pode observar na Imagem 14 o servidor integra ainda a implementação do módulo de classificação, implementados utilizando a API do Weka e a estrutura de conhecimento implementada no Neo4J. O servidor comunica ainda com a base de dados *MySQL* que armazena as notícias recolhidas e a informação sobre os utilizadores registados no sistema.

Utilizando as tecnologias anteriormente mencionadas foram implementadas as seguintes funcionalidades na interface do *Web Service*:

- Função *doLogin*: disponibiliza o mecanismo de autenticação à aplicação cliente.
- Função *setProfile*: permite à aplicação cliente enviar o perfil de utilizador ao módulo de classificação.
- Função *getTestSet*: permite à aplicação cliente requisitar conjuntos de notícias que neste caso correspondem aos *Test Sets* utilizados nas experiências 3 e 4.
- Função *setFeedback*: permitem ao cliente transmitir o *feedback* do utilizador ao módulo de classificação.
- Função *doEvaluation*: esta função trabalha em conjunto com a anterior, solicitando ao módulo de classificação a avaliação da aprendizagem aquando da recolha de *feedback* por parte do Agente Pessoal. Esta função é utilizada essencialmente no contexto das experiências 3 e 4 descritas de seguida, nas Secções 4.5 e 4.6.

Implementação da Interface do Agente Pessoal

Relativamente à interface do Agente Pessoal este consiste numa aplicação desenvolvida utilizando a plataforma *Google Android*, permitindo a cada utilizador receber notícias para avaliação e respetiva devolução de *feedback* relativo à sua leitura e classificação.

Esta aplicação disponibiliza as seguintes três interfaces essenciais que podem ser observadas na Imagem 15:

- A **interface (a)** de autenticação do utilizador (canto superior esquerdo da Imagem 15);
- A **interface (b)** de escolha de preferências do utilizador(canto inferior esquerdo da Imagem 15);
- A **interface (c)** de leitura de notícias e envio de *feedback* (lado direito da Imagem 15);

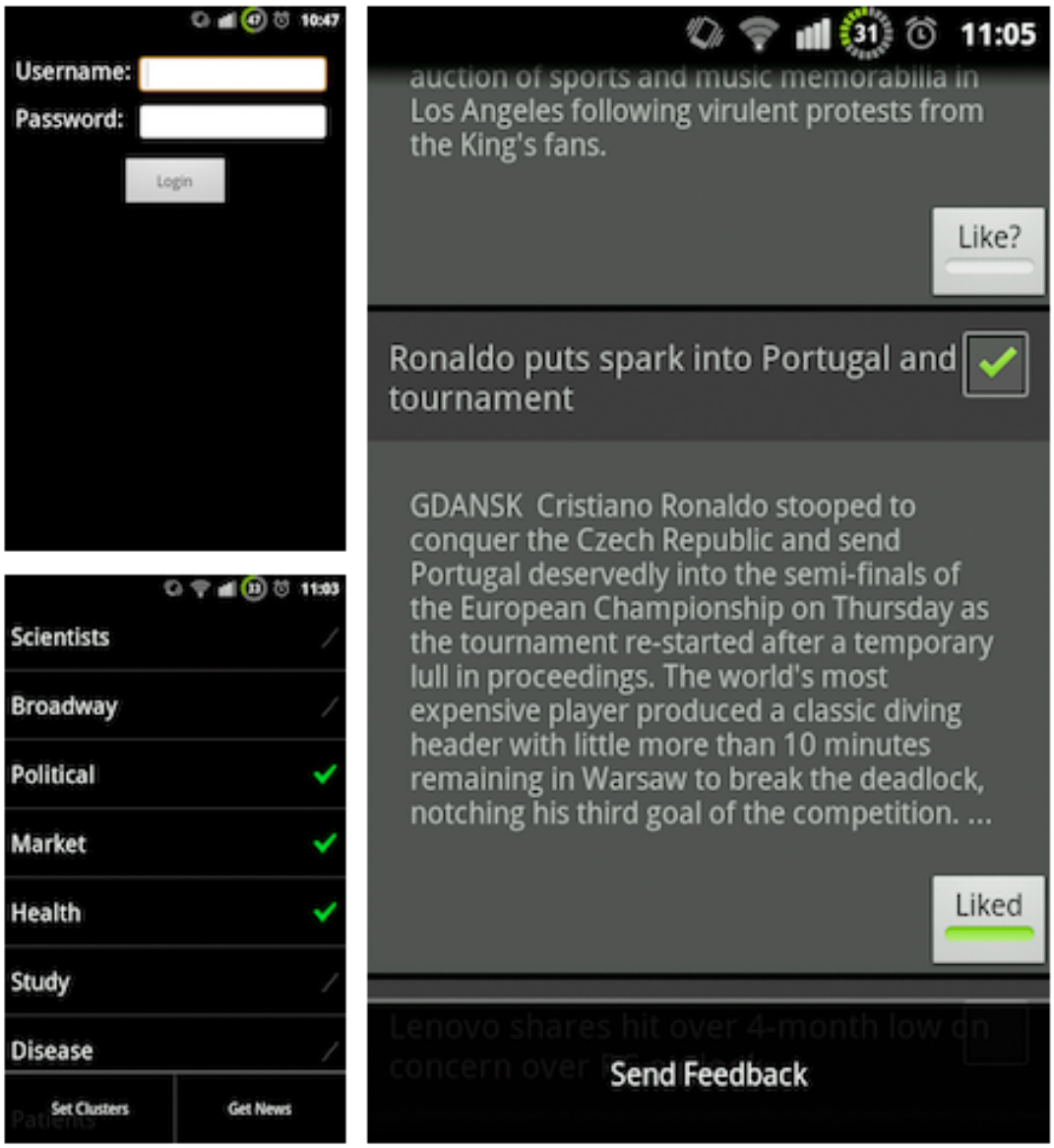


Imagem 15 – Interfaces da aplicação *Android*.

4.4.2 Modelo de Utilizador

Um passo essencial para o desenvolvimento de um sistema personalizado, como é o caso do sistema de recomendação aqui proposto, consiste na identificação não só dos conteúdos e das suas características, mas também na identificação dos utilizadores alvo. Nas secções anteriores foram especificados os modelos de dados relativos aos conteúdos (notícias) bem como o seu processo de estruturação e classificação (ver 0 e 4.3). Nesta secção será abordada a representação do modelo de utilizador e a forma como este foi codificado internamente no Sistema de Agente Pessoal.

Dependendo dos objetivos e especificidades de cada sistema de recomendação é natural que a forma como o modelo de utilizador é identificado e o modelo utilizado para o representar variem e se apresentem nas mais diversas formas. Ainda assim, de uma forma genérica o Modelo de Utilizador (MU) representa as suas preferências

relativamente a um determinado domínio, ou seja, o MU consiste num mapeamento entre o modelo de dados que representa os conteúdos e o modelo de dados que representa as preferências dos utilizadores relativamente aos conteúdos.

No contexto deste trabalho foi necessário que a representação escolhida para representar os conteúdos e o modelo de utilizador se enquadrassem na representação de conhecimento abordada na Secção 4.3. Por outro lado, de modo a utilizar aprendizagem computacional para dinamicamente identificar as preferências dos utilizadores e realizar as recomendações, tornou-se necessário representar o *feedback* dos utilizadores de uma forma que pudesse ser utilizada no treino dos algoritmos de aprendizagem. O processo de aprendizagem consiste no treino de classificadores que com base no *feedback* dos utilizadores classificam automaticamente novas instancias, ou seja, novas notícias.

De seguida é descrita a forma de representação das notícias classificadas pelos utilizadores que constituem o MU e consequentemente, a base do treino de aprendizagem.

4.4.3 Representação das Instâncias de Treino

Diversas abordagens de sistemas de recomendação que utilizam *content-based filtering* sugerem o uso de palavras-chave (extraídas automaticamente dos conteúdos) ou *social-tags* (adicionadas manualmente pelos utilizadores) para representar os conteúdos a classificar. Este tipo de abordagem possui um problema que se prende com o número elevado de palavras-chave que podem surgir no sistema, tornando impraticável o seu uso para treino de classificadores. Para contornar este problema é frequente a utilização de um vocabulário controlado, limitado e manualmente definido, por forma a reduzir o número de palavras-chave a considerar.

Na abordagem proposta por esta tese decidiu-se não utilizar a opção de construção de um vocabulário controlado, procurando assim uma abordagem genérica e flexível, tornando o sistema adaptável a novos conteúdos que forem sendo subscritos pelos utilizadores. Esta opção tornou necessária a criação de um método de redução do número de características a utilizar no treino dos classificadores.

Assim, dadas as condicionantes anteriormente referidas foi concebida a seguinte representação para codificar as instancias (notícias) a classificar, tendo como objetivo reduzir o vetor de características que define cada notícia. Como foi referido na Secção 0, a cada notícia foi associado um conjunto de cinco palavras-chave extraídas automaticamente pelo módulo de extração. Estas palavras-chave constituem a informação que caracteriza cada notícia, sendo que é a partir delas que são criadas a instâncias utilizadas para treinar os classificadores.

A figura presente na Imagem 16 representa o processo de correspondência e codificação das notícias como instâncias de treino:

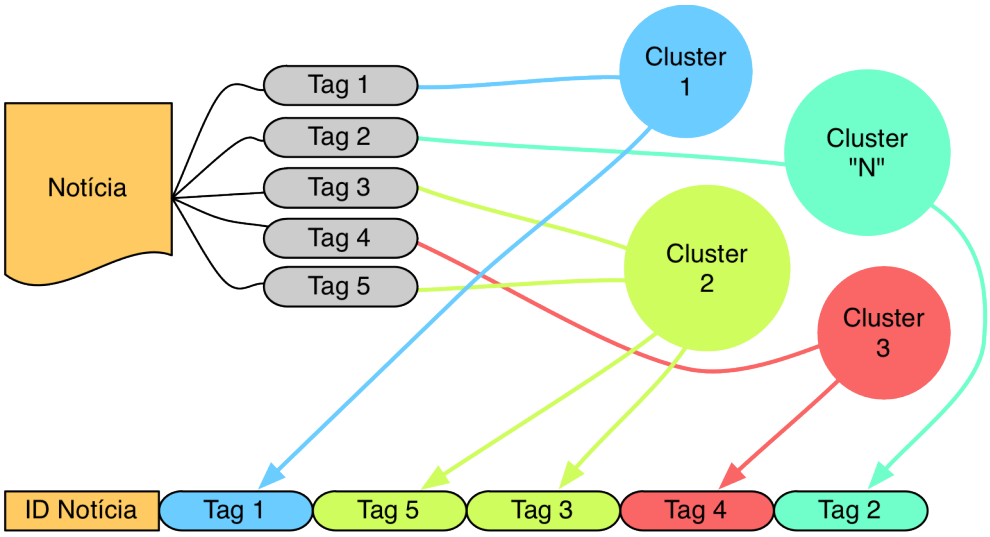
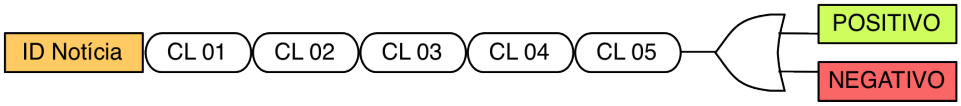


Imagem 16 - Codificação das instâncias de treino.

Para cada palavra-chave foi efetuada uma correspondência ao *cluster* a que pertencem, sendo deste modo possível enquadrar de forma simplificada e com um numero controlado de características cada uma das notícias, no universo das temáticas (tópicos de interesse) identificadas e presentes na estrutura de conhecimento. Podemos interpretar esta representação como um "código genético temático" simplificado que representa o enquadramento de cada notícia, de acordo com a estrutura de conhecimento do sistema e dos tópicos de interesse identificados.

A codificação anteriormente descrita e definida permite então definir o modelo de utilizador como o conjunto de instâncias classificadas, positivamente ou negativamente, por um utilizador do sistema, na seguinte forma:



Esta codificação é utilizada no sistema para representar não só instâncias dos conjuntos de treino, mas também as instâncias dos conjuntos de teste. Deste modo obtém-se uma representação universal de instâncias (notícias classificadas) que será utilizada pelos Agentes para representar as escolhas dos utilizadores.

4.4.4 Criação dos Conjuntos de Treino

Para levar a cabo os testes realizados ao sistema foi necessário recolher um conjunto significativo de notícias, para tal, o sistema de agregação descrito em 0 foi colocado em funcionamento durante duas semanas, tendo recolhido cerca de 6000 notícias que foram armazenadas cronologicamente em base de dados.

Para realizar as experiências e testes dos protótipo implementados a base de dados foi dividida em três partes distintas, contínuas mas cronologicamente separadas, contendo cerca de 2000 notícias cada. Esta divisão serviu para a criação de três fontes distintas de instâncias de teste e treino para as experiências realizadas.

Para esta primeira experiência com os Agente Pessoais - Experiência 3 - (ver Secção 4.5) foram utilizados os dois primeiros terços da base de dados de notícias.

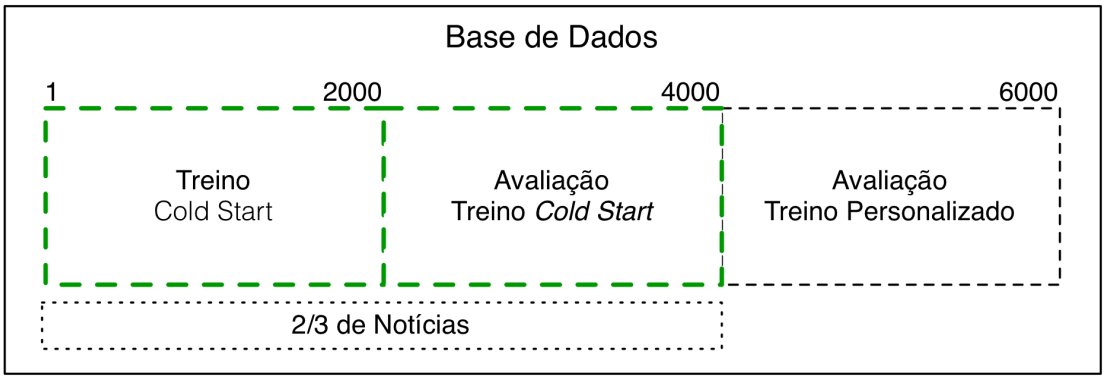


Imagem 17 - Divisão da base de dados de notícias para os testes experimentais.

A partir do primeiro terço da base de dados de conteúdos, foram recolhidas as notícias a partir das quais foi criado o conjunto de instâncias treino para a Experiência 3 (Treino *Cold Start).* Para realizar a avaliação do Treino *Cold Start* foram recolhidas notícias provenientes do segundo terço da base de dados, notícias que foram classificadas pelos utilizadores na realização da avaliação desta experiência.

A última e terceira parte da base de dados ficou reservada para a recolha das notícias utilizadas como conjunto de teste para a avaliação da experiência de Treino Personalizado, que foram também classificadas pelos utilizadores na realização da Experiência 4 (ver Secção 4.6). Relativamente ao conjunto de treino utilizado na experiência de Treino Personalizado este será oportunamente explicado na Secção 4.6.

4.4.5 Implementação do Módulo de Classificação

O módulo de classificação (Aprendizagem) foi implementado utilizando a *API* do *Weka* (ver descrição em 3.3.4) e integrado no Agente Pessoal na forma descrita em 4.4.1.

Neste módulo foram implementados três classificadores, nomeadamente *Naive Bayes*, *J48* (*C4.5*) e *RBF Networks,* utilizando as suas configuração "por defeito" na *framework* do *Weka.* A escolha específica destes algoritmos de aprendizagem teve como objetivo avaliar três métodos de natureza distintas, respetivamente um classificador Bayesiano, uma Árvore de Decisão e uma Rede Neuronal. Uma descrição mais detalhada destes algoritmos pode ser encontrada na Secção 3.3.2.

Deste modo, foi possível efetuar uma avaliação comprativa da performance dos algoritmos de aprendizagem à medida que os utilizadores liam notícias e devolviam *feedback* ao módulo de classificação.

A experiências realizadas abordaram o problema da aprendizagem do modelo de utilizador e respectiva recomendação baseada em preferências (ver 2.2.2), assim como o problema de *Cold Start* associado aos sistemas de recomendação (ver 2.5).

De seguida são descritas as experiências realizadas relativamente à aprendizagem utilizando o terceiro protótipo do sistema de recomendação, utilizando a aplicação móvel anteriormente especificada (ver 4.4.1) como interface de testes.

4.5 Experiência 3 (Treino Cold Start)

A primeira experiência realizada com o protótipo de Agente Pessoal, teve como objetivo avaliar a implementação de um mecanismo que pudesse suprimir o problema de *Cold Start*, identificado em 2.5 e que afeta a maioria dos sistemas de recomendação.

Ao utilizador, são disponibilizados no arranque da aplicação móvel, o conjunto dos tópicos de interesse mais relevantes identificados na estrutura de conhecimento aquando da elaboração do processo de *clustering* descrito na Secção 4.3.3. Utilizando a interface disponibilizada pela aplicação móvel (ver Imagem 15) o utilizador escolhe os tópicos de sua preferência, os quais são enviado para o Módulo de Classificação que os utiliza para criar os conjuntos de treino dos classificadores, de acordo com aos tópicos selecionados.

Para implementar o processo de aprendizagem, treinando os classificadores para recomendar notícias correspondentes às categorias selecionadas, foi necessário recorrer a uma heurística capaz de fornecer elementos de treino ao classificador, tendo como referência o tópico de interesse selecionado pelo utilizador. A heurística seguida será explicitada na secção seguinte.

4.5.1 Metodologia de Treino *Cold Start*

Após uma análise manual e extensiva das instâncias criadas para representar as notícias, instâncias estas na forma descrita em 4.4.3, foi possível perceber que instâncias com "mais de duas palavra-chave" pertencentes ao mesmo *cluster* constituíam um bom exemplo de um notícia pertencente ao tópico de interesse representado por esse *cluster*. A experiência seguinte pretende colocar à prova esta hipótese, testando a sua eficácia no treinos dos classificadores para cada tópico de interesse selecionado por cada utilizador.

Na imagem seguinte podemos observar uma representação do tipo de instâncias que segue a heurística referida anteriormente, e que constituem então exemplos do treino utilizado:

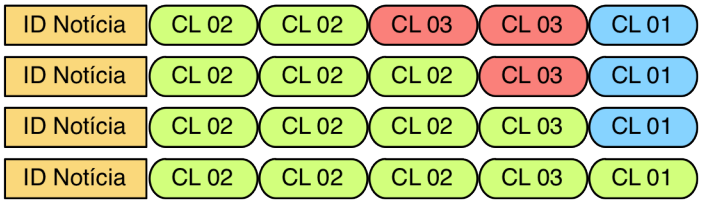


Imagem 18 - Exemplo de treino Cold Start.

Considerando os exemplos apresentados na Imagem 18 e assumindo por hipótese que um determinado "*Cluster 02*" correspondente ao tópico de interesse "Desporto" podemos então utilizar instâncias com esta configuração para treinar classificadores capazes de entregar aos utilizadores notícias da categoria Desporto. O mesmo pode ser efetuado para qualquer tópico identificado na estrutura de conhecimento.

Quando um utilizador seleciona este tópico de interesse no arranque da aplicação móvel, o seu Agente Pessoal "comunica" ao Módulo de Classificação a escolha do utilizador, que utilizando a heurística representada de seguida na Imagem 19 recolhe exemplos de treino para o tópico selecionado.

Na implementação do Treino *Cold Start* utilizada neste trabalho o Módulo de Classificação recolhe um número igual de elementos de treino para cada uma das classes positiva e negativa por forma a equilibrar o conjunto de treino. As classes positiva e negativa correspondem respetivamente à representação do interesse e desinteresse do utilizador em cada notícia/instância. Na figura seguinte podemos observar um esquema representativo da seleção dos casos de treino, segundo a heurística que permite ao Agente Pessoal (Módulo de Classificação) selecionar o treino consoante o tópico selecionado pelo utilizador.

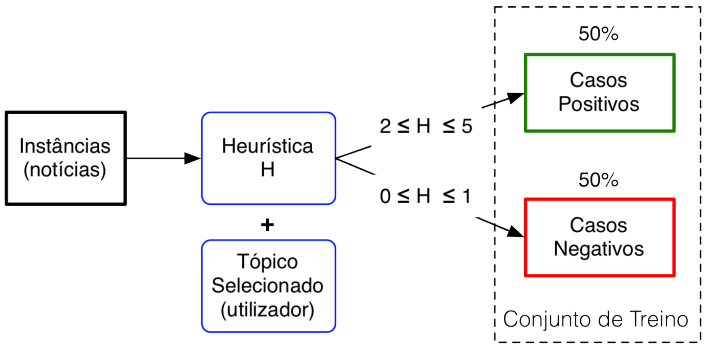


Imagem 19 - Heurística para o treino Cold Start.

H = número de palavras-chave pertencentes ao *cluster* representativo do tópico de interesse selecionado pelo utilizador.

Nos testes efetuados fez-se variar o tamanho dos conjuntos de treino entre os 25 e os 150 casos de treino, com incrementos de 25 instâncias entre teste, por forma a verificar a evolução do processo de aprendizagem (performance dos classificadores) com o aumento do número dos exemplos fornecidos aos classificadores. Foram utilizados conjuntos de instâncias com casos positivos e negativos para cada tópico de interesse específico sendo que, os casos positivos e negativos foram recolhidos aleatoriamente a partir do primeiro terço da base de dados de notícias como foi especificado na secção 4.4.4.

4.5.2 Procedimento Experimental

Como já foi anteriormente referido, o objetivo desta experiência consiste em avaliar a capacidade do sistema implementado treinar de forma automática, classificadores

capazes de selecionar notícias de uma determinada categoria (tópico de interesse) da preferência dos utilizadores, no arranque da aplicação.

Para tal foi solicitado a um conjunto de nove utilizadores que participassem nesta experiência, utilizando o protótipo da aplicação móvel desenvolvida para ler notícias.

Os nove utilizadores foram divididos em três grupos de três utilizadores consoante as suas preferências específicas. Nesta experiência foram criados três grupos estereotipados de utilizadores, sendo que foram selecionados de forma a fazer corresponder as suas preferências com um tema noticioso bem definido.

Os temas selecionados para teste foram as categorias Economia, Política e Desporto, correspondentes aos *Clusters* identificados pelos tópicos "Market", "Political" e "Game" respectivamente. Estas três categorias foram selecionadas por corresponderem às temáticas de três dos *clusters* mais significativos, relativamente ao tamanho e abrangência presentes no sistema. Por outro lado, houve a intenção de colocar em teste uma categoria com uma temática bem delimitada, o caso de "Desporto", bem como duas outras categorias que apresentam alguma probabilidade de sobreposição temática, os casos das temáticas "Política" e "Economia". Estas duas últimas categorias apresentam à partida maior subjetividade na avaliação efetuada pelos utilizadores por apresentarem conteúdos eventualmente comuns. Dado que na estrutura de conhecimento construída não existem palavras-chave sobrepostas entre tópicos de interesse, torna-se interessante verificar o efeito da separabilidade das mesmas. A utilização de três tópicos de interesse (categorias) bem definidos e utilizadores estereotipados com preferências bem definidas torna-se essencial para conferir um nível de controlo considerável durante os testes efetuados.

A cada um dos três grupos foram entregues durante uma semana um total de 150 notícias aleatoriamente distribuídas pelas diversas temáticas presentes no conjunto de notícias presentes no segundo terço da base de dados (ver 4.4.4). Inicialmente foi solicitado aos utilizadores que assinalassem um tópico de interesse de sua preferência (ver Imagem 15 interface b) e de seguida que efetuassem a leitura das notícias, classificando como interessantes as notícias que iam de encontro à sua preferência. Para identificar o interesse numa notícia, os utilizadores colocavam um "Like" na notícia correspondente (ver Imagem 15 interface c).

O resultado do processo de leitura das 150 notícias e o respetivo *feedback* dos utilizadores constitui o primeiro conjunto de teste, ou seja, um conjunto de notícias classificadas pelos seus leitores como interessantes (atribuição de classe positiva) ou não interessantes (atribuição de classe negativa) de acordo com a sua preferência. Este primeiro conjunto de notícias classificadas pelos utilizadores durante a Experiência 3 foi designado por "Conjunto de Teste 1".

Na avaliação desta experiência foi testada a capacidade do sistema reconhecer notícias de um determinado tópico de interesse, confrontando os resultados obtidos pelos classificadores pré-treinados, contra o feedback proveniente das escolhas dos utilizadores (Conjunto de Test1). O resultados obtidos foram recolhidos a partir do

módulo *"Evaluation"* que o *Weka* implementa associado a cada um dos algoritmos de aprendizagem utilizados, fornecendo os resultados estatísticos da avaliação da Precisão, Abrangência e F_1 de cada um dos classificadores.

Na secção seguinte apresentam-se os gráficos dos resultados obtidos para a Experiência 3.

4.5.3 Resultados Experimentais

De seguida são apresentados os resultados obtidos na avaliação desta primeira experiência referente ao "Treino *Cold Start*", avaliações realizadas como já foi referido anteriormente para os tópicos de interesse "Desporto", "Política" e "Economia". Cada categoria foi avaliada por três utilizadores distintos. Deste modo, os gráficos apresentados de seguida, apresentam os valores médios dos resultados obtidos pelos três utilizadores que avaliaram cada uma das categorias (tópicos de interesse).

Resultados médios (3 avaliações) para a categoria Game (Desporto)

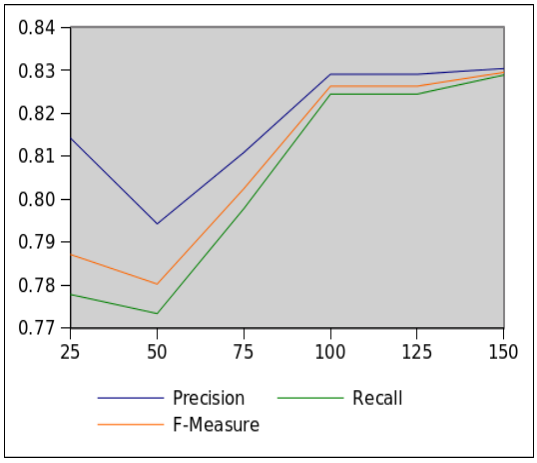


Imagem 20 - Média *Naive Bayes*

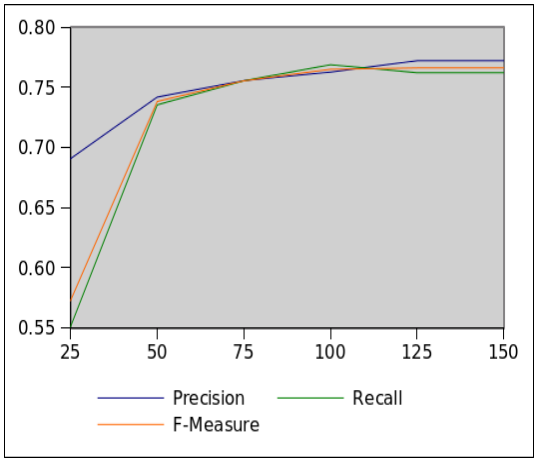


Imagem 21 - Média *J48*

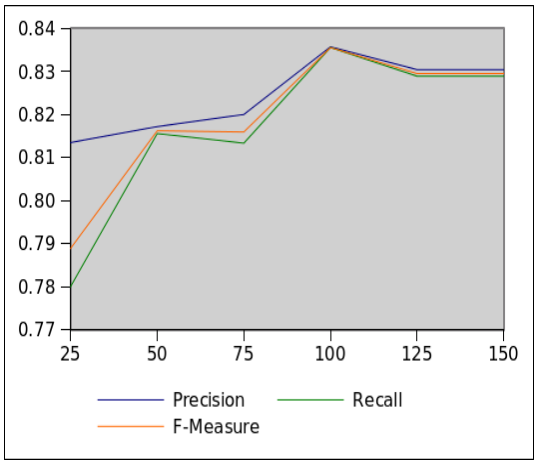


Imagem 22 - Média *RBF Network*

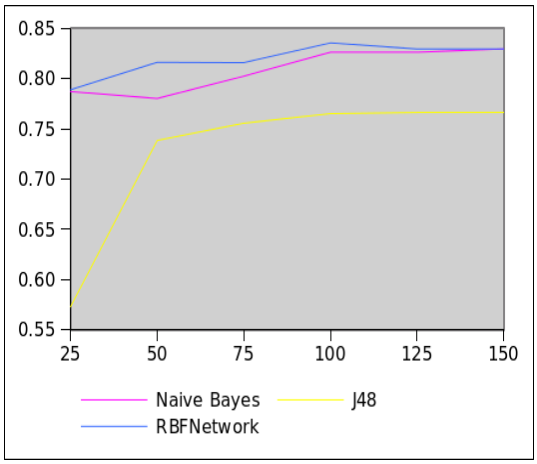


Imagem 23 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador.

Os gráficos presentes na, Imagem 21 e Imagem 22 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 23 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios (3 avaliações) para a categoria Market (Economia)

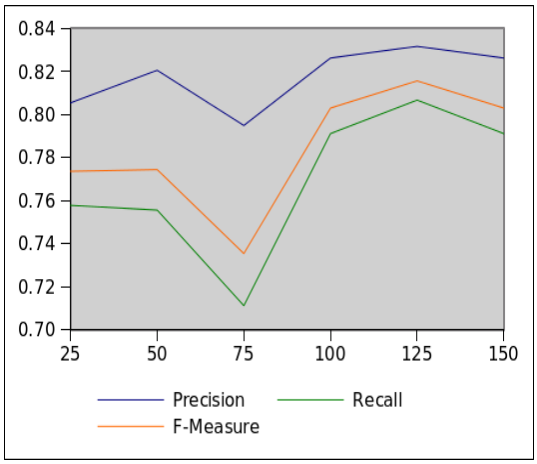


Imagem 24 - Média *Naive Bayes*

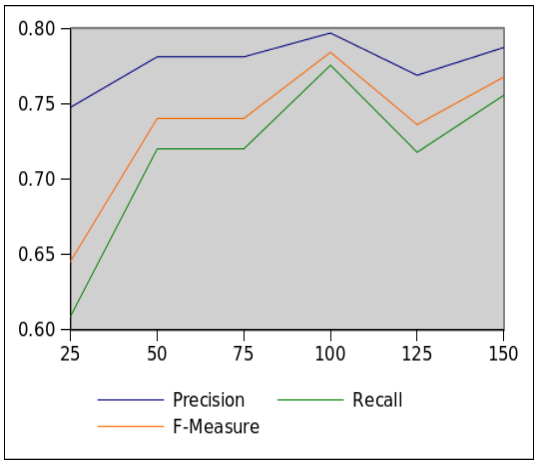


Imagem 25 - Média *J48*

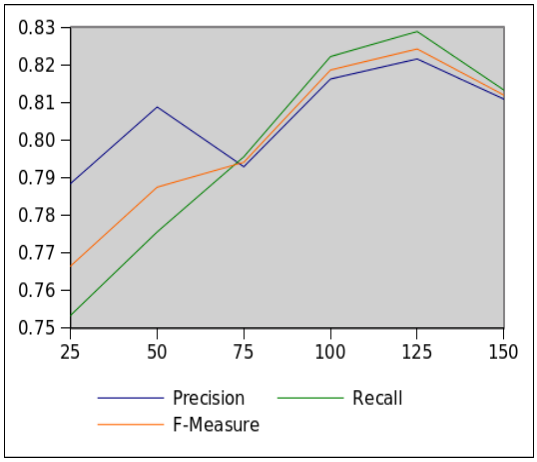


Imagem 26 - Média *RBF Network*

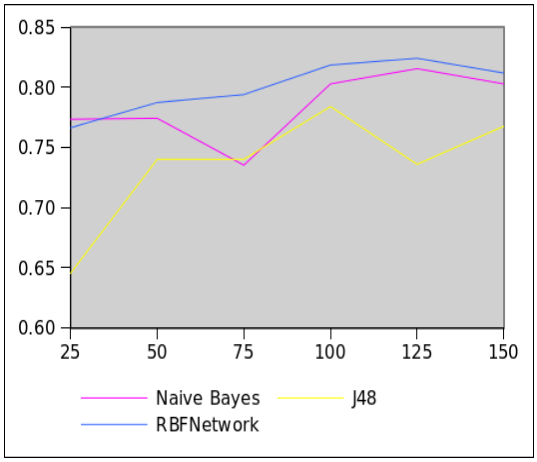


Imagem 27 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 24, Imagem 25 e Imagem 26 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 27 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios (3 avaliações) para a categoria Political (Política)

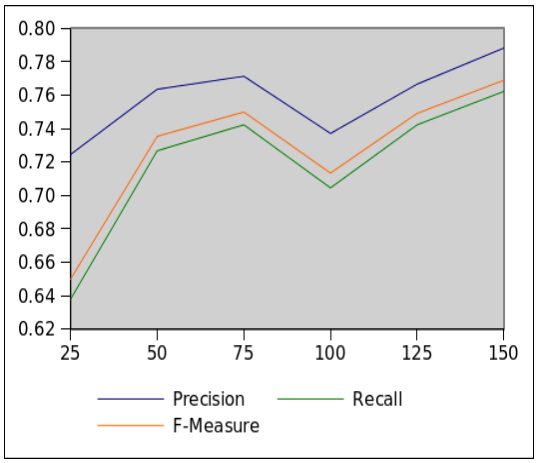


Imagem 28 - Média *Naive Bayes*

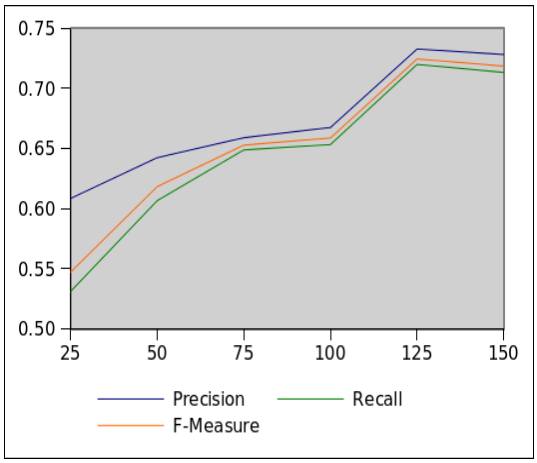


Imagem 29 - Média *J48*

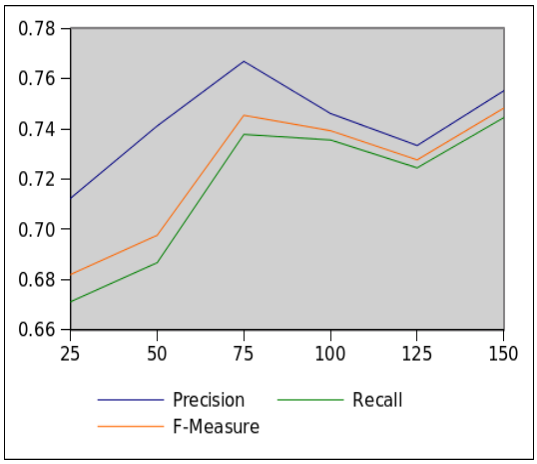


Imagem 30 - Média *RBF Network*

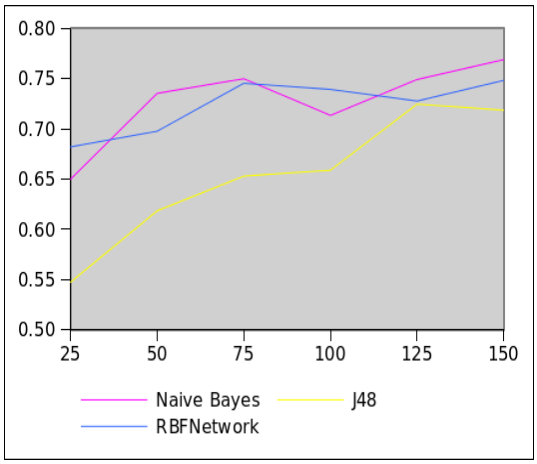


Imagem 31 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes nas Imagem 28, Imagem 29 e Imagem 30 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 31 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios para as 3 Categorias (Game, Market, Political)

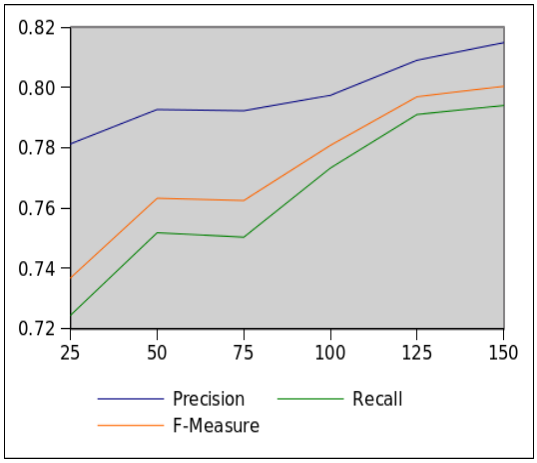


Imagem 32 - Média *Naive Bayes*

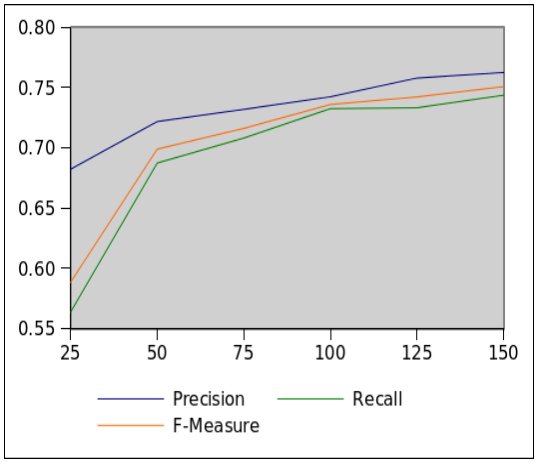


Imagem 33 - Média *J48*

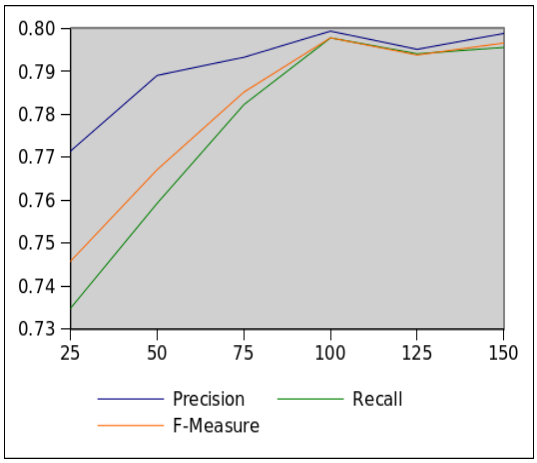


Imagem 34 - Média *RBF Network*

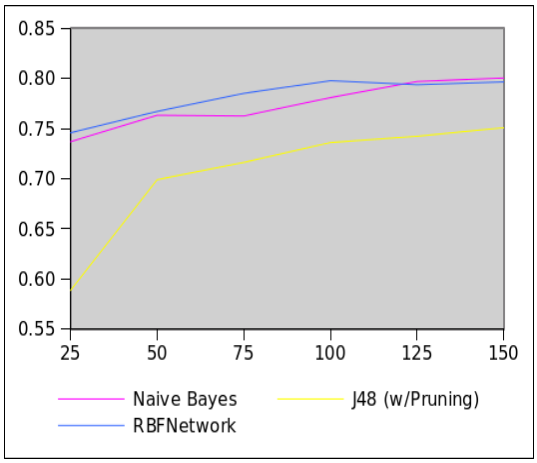


Imagem 35 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical os valores médios para as três categorias (em percentagem) dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 32, Imagem 33 e Imagem 34 apresentam os resultados médios para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 35 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores para as três categorias.

4.5.4 Conclusões para a Experiência 3

Nesta primeira experiência utilizando o protótipo do Agente Pessoal foi possível verificar em primeiro lugar o correto funcionamento e interligação entre os diversos componentes que compunham o módulo de classificação. Foi também avaliada a performance dos três classificadores implementados confrontando os seus resultados com a avaliação manual efetuada pelos utilizadores. Os resultados obtidos, anteriormente apresentados, são de seguida analisados.

Análise para a categoria Game (Desporto)

Para o tópico "Desporto" os resultados dos três classificadores apresentam um desfasamento considerável entre a precisão e a abrangência (*recall*) para o primeiro conjunto de 25 instâncias de treino. Com 50 instâncias de treino verifica-se uma aproximação dos valores da precisão e abrangência, sendo que a partir deste momento o comportamento dos classificadores começam a estabilizar e a apresentar um aumento da sua performance. Ainda que com linhas de evolução diferentes, o aumento de performance torna-se visível no aumento da F_1 para os três classificadores. A partir das 100 instâncias de treino todos os classificadores apresentam uma estabilização da curva de aprendizagem.

O pico de performance verificado tanto para *Naive Bayes* como para o *J48* verificou-se às 150 instâncias de treino como seria esperado num comportamento natural dos classificadores. Ainda assim a *RBF Network* apresentou o seu máximo de performance às 100 instâncias de treino, um pouco mais cedo que os anteriores, sendo que foi o classificador que atingiu maior performance com um valor de pico para a F_1 de 84%, estabilizando de seguida nos 83%.

Tanto os classificadores *Naive Bayes* como *RBF Network* estabilizaram a F_1 aos 83% atingidas as 150 instâncias de treino. Apesar de apresentarem evoluções diferentes ao longo do aumento das instâncias de treino, os classificadores *Naive Bayes* e *RBF Network* apresentaram resultados máximos semelhantes, ainda assim a *RBF Network* superou o classificador *Naive Bayes* por uma pequena margem para conjuntos de treino mais pequenos. Já no caso do *J48* a sua evolução seguiu sempre no sentido da melhoria de performance apesar da sua F_1 ter estabilizado nos 77%, um pouco abaixo dos demais classificadores.

Análise para a categoria Market (Economia)

Para o tópico "Economia" os resultados dos três classificadores apresentam maior desfasamento entre a precisão e a abrangência (*recall*) que no caso anterior "Desporto", não só para o menor conjunto de 25 instâncias de treino mas ao longo de todo o processo de treino. Este resultado poderá deve-se a um fator já referido na preparação desta experiência e que se prende com o fato de existir maior sobreposição temática das notícias sobre "Economia" com outros tópicos de interesse, tornando as escolhas dos utilizadores potencialmente mais ambíguas. Neste caso o conjunto de teste, contendo as instâncias classificadas pelos utilizadores, deverá conter também

maior divergência nos casos possíveis para este tópico de interesse. Quer estejamos a falar de desporto, política ou saúde, o assunto "dinheiro" é transversal a praticamente todos os tópicos de interesse, daí poder existir esta ambiguidade nas escolhas dos utilizadores. Nesta situação este fator parece revelar-se na performance dos classificadores.

Ainda que com uma tendência de evolução positiva, tanto o classificador *Naive Bayes* como *J48* revelam maior instabilidade na aprendizagem. É possível observar a existência de inversões descendentes acentuadas na variação do valor de F_1 para ambos os classificadores. O classificador *Naive Bayes* atingiu valores para a F_1 acima dos 80% ao passo que o *J48* ficou um pouco abaixo desta marca. Para este tópico o classificador *RBF Network* revelou-se como o mais estável à medida que o tamanho do conjunto de treino foi aumentando, estabilizando a F_1 a rondar os 82%.

Análise para a categoria Political (Política)

O tópico "Política" apresentou-se como o grande desafio para os classificadores dado que este é o tema com maior representatividade e dispersão de conteúdos no universo das notícias recolhidas.

Neste caso a evolução da aprendizagem voltou a ser pouco uniforme, sendo que a performance dos classificadores ficou abaixo dos valores obtidos para os tópicos anteriores "Desporto" e "Economia". Os três classificadores ficaram abaixo dos 80% para a F_1. A justificação para esta diminuição na performance está provavelmente relacionada com o maior número de casos para classificar, para o mesmo número de elementos de treino utilizados. A divergência da precisão e abrangência é aqui bastante significativa para conjuntos de treino pequenos (menos de 75 instâncias), justificando a hipótese colocada anteriormente.

Para este tópico o classificador *Naive Bayes* revelou-se superior à *RBF Network* tendo sido o único classificador a ultrapassar os 75% para a F_1. Outro resultado interessante foi obtido pelo classificador J48 que apresentou uma evolução na aprendizagem bastante regular, tendo atingido valores bastante próximos da *RBF Network* às 125 instâncias de treino. O processo de *"prunning"* utilizado no J48, parece nesta circunstância ter dado os seus frutos, na medida em que terá evitado algum efeito do ruído.

Considerações finais para a Experiência 3

Tal como esperado e a partir da análise dos gráficos dos valores médios para os três tópicos podemos verificar que os classificadores *Naive Bayes* e *RBF Network* apresentam os melhores resultados. Estes dois classificadores não só apresentam performance uma performance como um comportamento evolutivo da aprendizagem semelhante, ou seja, apresentam resultados semelhantes para os mesmos conjuntos de treino. O valor da F_1 para estes dois classificadores atinge cerca de 80% acima das 100 instâncias de treino. Apesar de resultados semelhantes para a F_1 existe maior divergência entre a precisão e abrangência para o classificador *Naive Bayes* que para *RBF Network* acima das 100 instâncias de treino. A *RBF Network* apresenta um

tendência convergente entre estes dois valores, sendo que se apresenta como o classificador com os melhores resultados médios para os três tópicos analisados. No caso do *J48* a performance fica cerca de 5% abaixo dos anteriores, alcançando 75% para a F_1 às 150 de instâncias de treino.

Relativamente ao treino e aprendizagem na experiência *Cold Start* pode-se constatar que apesar de ter sido utilizada uma heurística bastante simples para escolha das instâncias de treino os resultados obtidos são bastante satisfatórios. Tanto o classificador *Naive Bayes* como o classificador *RBF Network* se revelaram adequados à função que desempenhavam, tendo atingido cerca de 80% para a F_1 num contexto em que não existia muita informação acerca das preferências específicas dos utilizadores. O classificador *J48* que implementa uma árvore de decisão *C4.5* apesar de revelar uma performance inferior conseguiu atingir resultados perto dos 75% para a F_1. Em algumas situações o classificador *Naive Bayes* apresentou um comportamento desviante, como é o caso da descida de performance na passagem de 25 para 50 casos de treino no tópico "Desporto". Para os três algoritmos de classificação a curva da F_1 estabiliza a partir das 100 instâncias de treino, demonstrando ainda assim uma tendência crescente do seu valor.

Como já foi referido anteriormente, a tentativa de equilibrar o número de instâncias de treino entre casos positivos e negativos para cada classe proporcionou uma evolução positiva da aprendizagem, acompanhando o aumento dos conjuntos de treino.

4.6 Experiência 4 (Treino Personalizado)

A segunda experiência realizada com o protótipo de Agente Pessoal tem como objetivo avaliar a capacidade do sistema de recomendação na aprendizagem do modelo de utilizador de forma dinâmica, classificando novas notícias de acordo com o seu histórico de notícias lidas e classificadas anteriormente. De seguida descrevem-se os procedimentos de implementação e avaliação desta experiência.

4.6.1 Metodologia de Treino Personalizado

Tal como na Experiência 3 descrita na secção anterior, nesta abordagem foi utilizado o mesmo método de classificação utilizando aprendizagem computacional, onde são testados novamente os classificadores *Naive Bayes*, *RBF Network* e *J48* implementados no módulo de classificação do sistema.

Contrariamente ao método de treino utilizado na experiência anterior, em que as instâncias de treino foram selecionadas heuristicamente, neste caso os classificadores são treinados utilizando o *feedback* do utilizador de modo aprender as suas preferências dinamicamente.

O conjunto de treino utilizado provém de notícias já lidas e classificadas pelo utilizador. Na sequencia da Experiência 3 (ver 4.5) houve a oportunidade de criar um conjunto de notícias classificadas por nove utilizadores que participaram na experiência. Recorde-se que nessa experiência os nove utilizadores foram divididos em subgrupos de três utilizadores consoante as suas preferências relativamente às categorias "Desporto", "Economia" e "Política" (ver 4.5.1). As notícias avaliadas manualmente pelos utilizadores na Experiência 3 ("Conjunto de Teste 1") funcionaram nesta segunda experiência como conjuntos de treino para os classificadores personalizados.

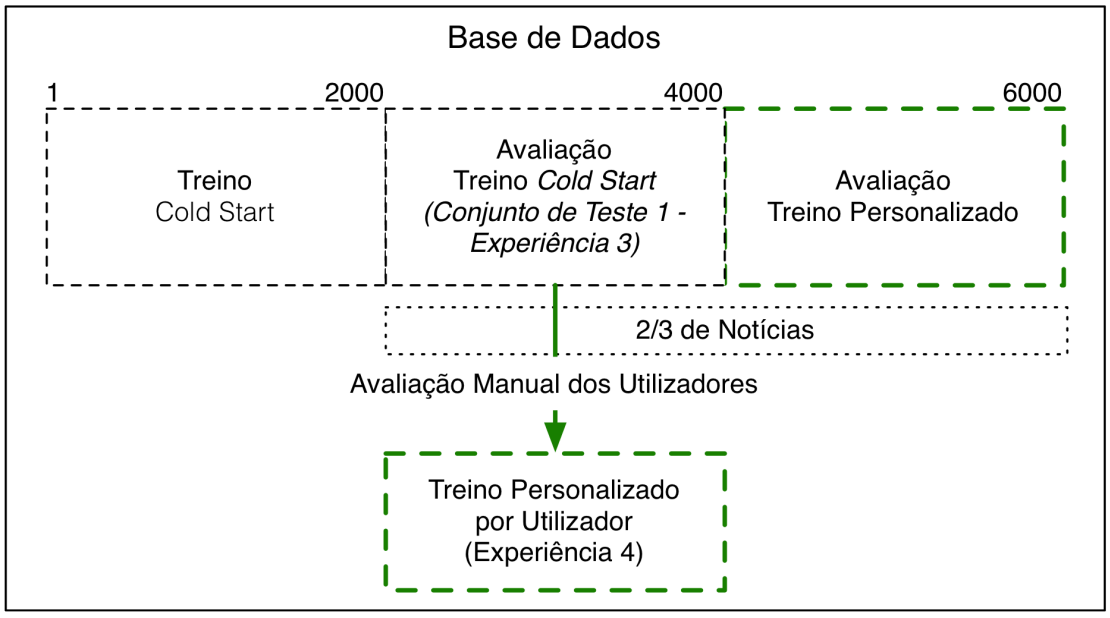


Imagem 36 - Avaliação do Treino Personalizado.

Nesta experiência o conjunto de teste consiste num segundo grupo de notícias avaliadas manualmente pelos utilizadores (utilizando a aplicação móvel), recolhidas do último terço da base de dados de notícias como podemos observar na Imagem 36.

4.6.2 Procedimento Experimental

Nesta experiência os classificadores *Naive Bayes*, *J48* e *RBF Network* são testados na aprendizagem da recomendação personalizada, de acordo com o histórico de notícias anteriormente classificadas por cada um dos utilizadores.

Para o treino dos classificadores, foram utilizadas as 150 instâncias classificadas manualmente por cada um dos utilizadores na avaliação manual realizada na Experiência 3. Os classificadores foram novamente treinados utilizando conjuntos de treino com 25, 50, 75, 100, 125 e 150 instâncias. Esta utilização incremental de instâncias de treino pretende simular a leitura e classificação gradual das 150 notícias por parte dos utilizadores ao longo do tempo.

O procedimento experimental da Experiência 4 seguiu então exatamente os mesmos processos implementados na Experiência 3, utilizando os mesmos nove avaliadores, com a respetiva alteração no método de treino e no conjunto de notícias de avaliação.

Novamente a cada um dos três grupos de utilizadores foram entregues durante uma semana um total de 150 notícias aleatoriamente distribuídas pelas diversas temáticas presentes no último terço da base de dados (ver Imagem 36). Foi-lhes solicitado que classificassem este novo conjunto de notícias de acordo com as suas preferências. O resultado do processo de leitura das 150 novas notícias e o respetivo *feedback* dos utilizadores constitui o segundo conjunto de teste, que foi aqui designado por "Conjunto de Teste 2".

Esta experiência contrariamente à Experiência 3 não utiliza conjuntos de treino equilibrados relativamente ao número de casos positivos e negativos (interesse/desinteresse do utilizador) pois utiliza todo o conjunto de 150 instâncias classificadas por cada um dos utilizadores. Neste cenário também não existe garantia que a distribuição de casos interessantes para o utilizador seja homogénea ao longo do tempo. Esta condicionante é um fator difícil de ultrapassar pois cada utilizador é livre de ler as notícias que entender. Por outro lado, torna-se difícil discriminar que notícias não lidas/não classificadas representam casos que não interessam ao utilizador. Apesar de um utilizador não ler uma notícia, não se pode afirmar que este comportamento significa desinteresse no tema, neste caso "assume-se" que não interessa.

Do confronto entre as escolhas dos classificadores treinados com as escolhas dos utilizadores na Experiência 3 (Conjunto de Teste 1), com as escolhas dos utilizadores nesta Experiência 4 (Conjunto de Teste 2) foram obtidos os resultados estatísticos desta experiência. Os resultados obtidos foram recolhidos a partir do módulo *"Evaluation"* que o *Weka* implementa associado a cada um dos algoritmos de aprendizagem utilizados, fornecendo os resultados estatísticos da avaliação relativamente aos conjuntos de teste.

4.6.3 Resultados Experimentais

De seguida são apresentados os resultados obtidos na avaliação desta primeira experiência referente ao "Treino Personalizado", avaliações realizadas como já foi referido anteriormente para os tópicos de interesse "Desporto", "Política" e "Economia". Cada categoria foi avaliada por três utilizadores distintos. Deste modo, os gráficos apresentados de seguida, apresentam os valores médios dos resultados obtidos pelos três utilizadores que avaliaram cada uma das categorias (tópicos de interesse).

Resultados médios (3 avaliações) para a categoria Game (Desporto)

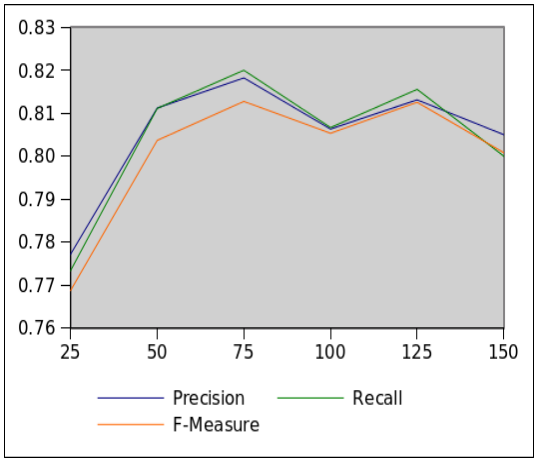


Imagem 37 - Média *Naive Bayes*

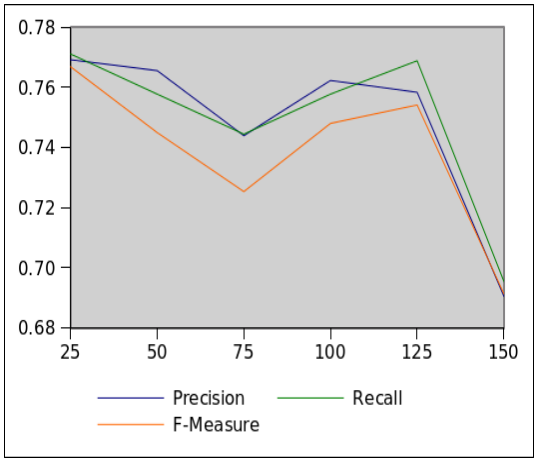


Imagem 38 - Média *J48*

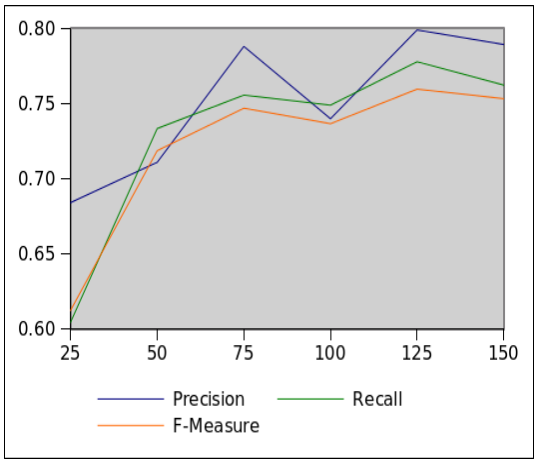


Imagem 39 - Média *RBF Network*

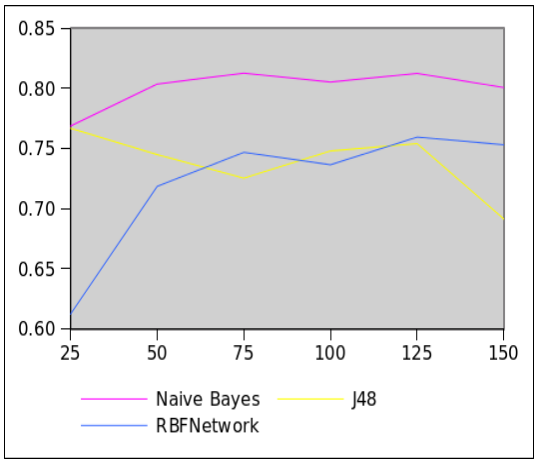


Imagem 40 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 37, Imagem 38 e Imagem 39 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 40 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios (3 avaliações) para a categoria Market (Economia)

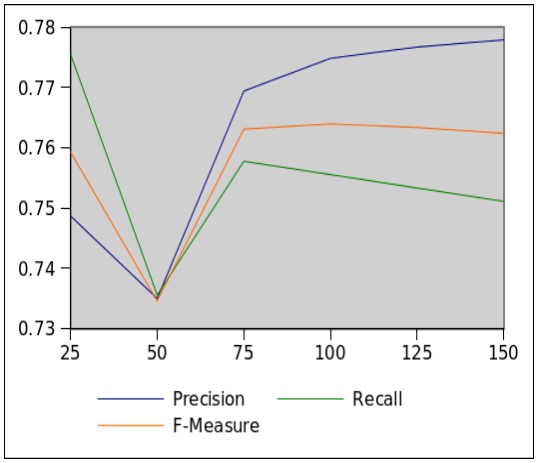


Imagem 41 - Média *Naive Bayes*

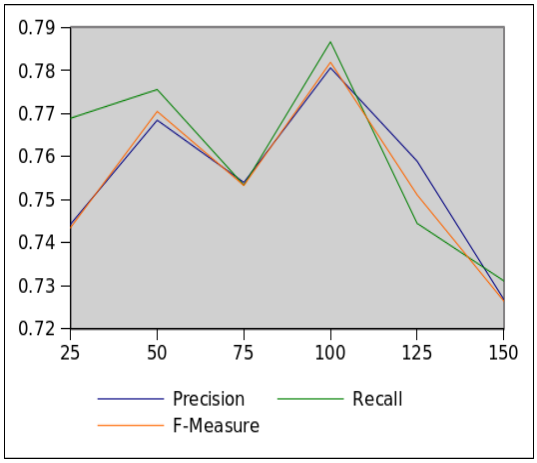


Imagem 42 - Média *J48*

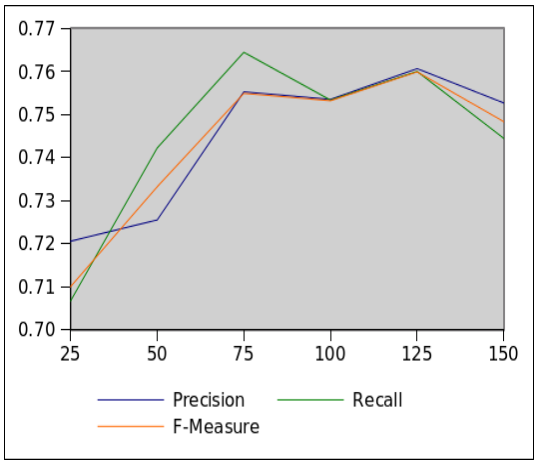


Imagem 43 - Média *RBF Network*

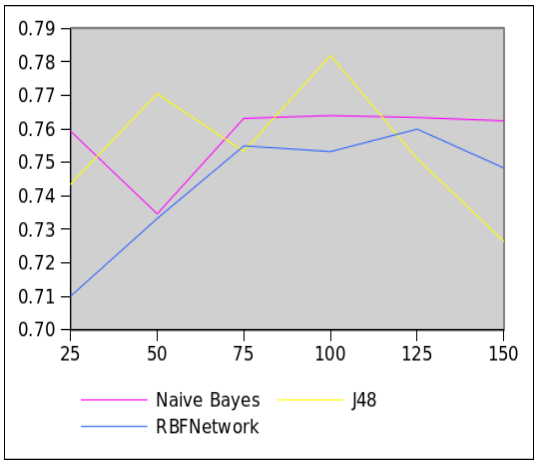


Imagem 44 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 41, Imagem 42 e Imagem 43 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Imagem 44 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios (3 avaliações) para a categoria Political (Política)

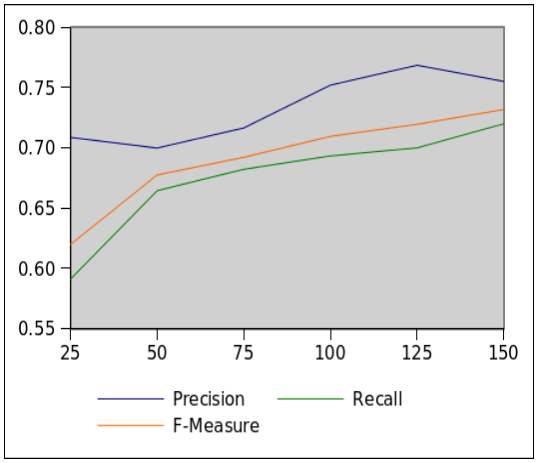


Imagem 45 - Média *Naive Bayes*

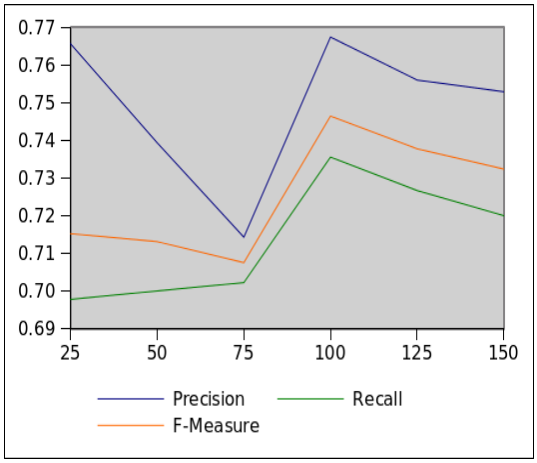


Imagem 46 - Média *J48*

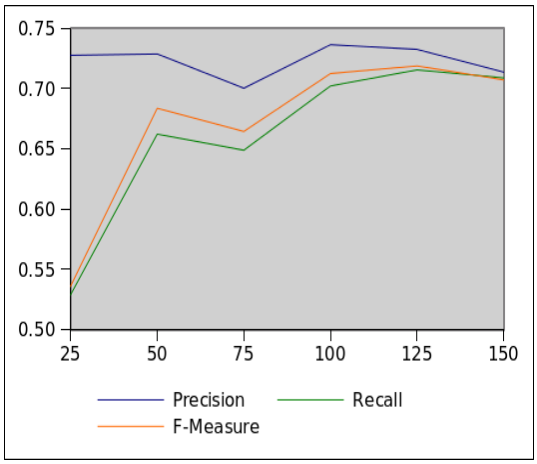


Imagem 47 - Média *RBF Network*

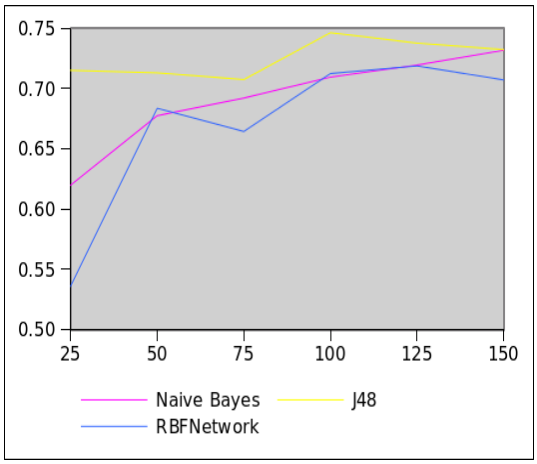


Imagem 48 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical o valor em percentagem dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 45, Imagem 46 e Imagem 47 apresentam os resultados para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Imagem 48 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores.

Resultados médios para as 3 Categorias (Game, Market, Political)

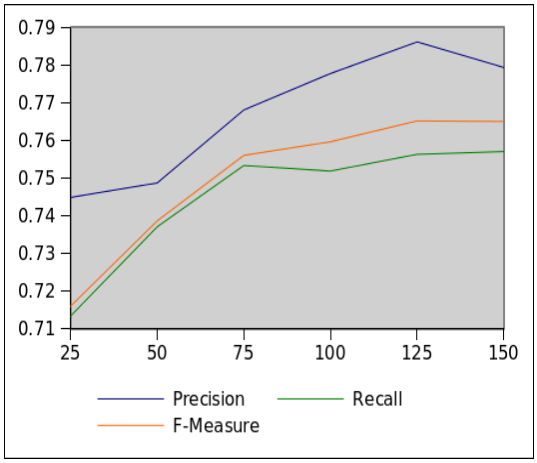


Imagem 49 - Média *Naive Bayes*

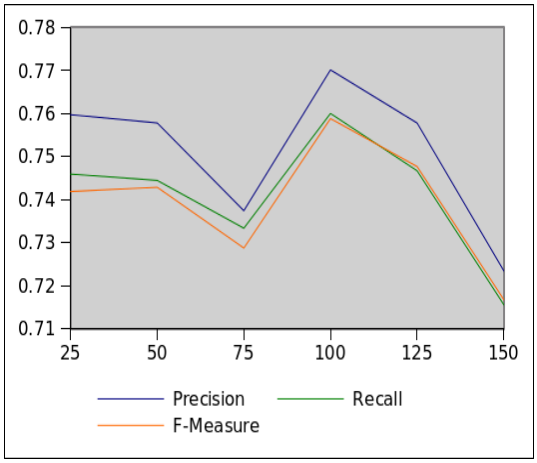


Imagem 50 - Média *J48*

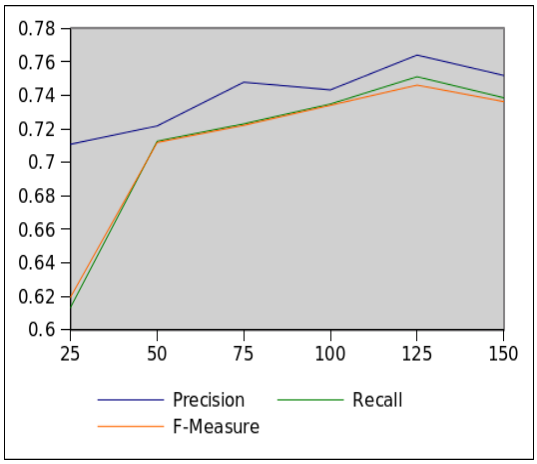


Imagem 51 - Média *RBF Network*

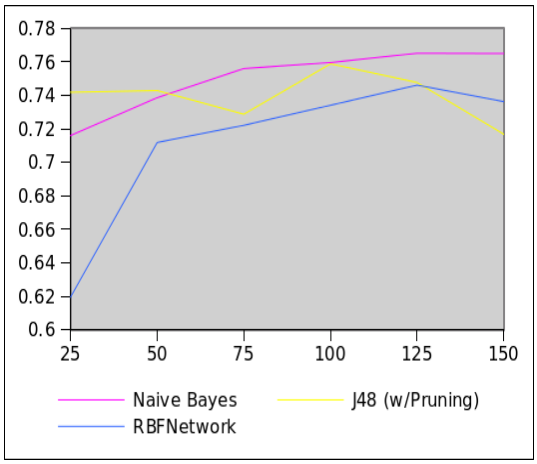


Imagem 52 - Comparação F_1 (*f-measure*)

No eixo horizontal de cada gráfico está representado o número de instâncias de treino utilizadas e no eixo vertical os valores médios para as três categorias (em percentagem) dos resultados obtidos para a Precisão (*Precision*), Abrangência (*Recall*) e F_1 (*f-measure*), para cada classificador. Os gráficos presentes na Imagem 49, Imagem 50 e Imagem 51 apresentam os resultados médios para as implementações dos classificadores *Naive Bayes*, *J48* e *RBF Network* respectivamente. Na Imagem 52 está representada a comparação dos resultados médios da F_1 para cada um dos três classificadores para as três categorias.

4.6.4 Conclusões para a Experiência 4

Nesta segunda experiência utilizando o protótipo do Agente Pessoal foi possível verificar em primeiro lugar o correto funcionamento e interligação entre os diversos componentes que compunham o módulo de classificação. Foi também avaliada a performance dos três classificadores implementados confrontando os seus resultados com a avaliação manual efetuada pelos utilizadores. Os resultados obtidos, anteriormente apresentados, são de seguida analisados.

Análise para a categoria Game (Desporto)

Da análise dos gráficos de resultados para o tópico de interesse "Desporto" podemos observar que os classificadores *Naive Bayes* e *RBF Network* continuam a oferecer desempenhos bastante bons, evoluindo positivamente com o aumento do número de instâncias de treino. Ainda assim, o classificador *RBF Network* foi neste caso suplantado pelo classificador *Naive Bayes*, sendo que este último apresentou valores para a F_1 acima dos 80% ao passo que *RBF Network* obteve cerca de 75%. Apesar de a evolução destes dois classificadores ao longo do tempo e com o aumento de casos de treino ter apresentado um tendência crescente da sua performance, o mesmo não aconteceu para o classificador *J48* tendo a sua performance sofrido degradação com o aumento dos conjuntos de treino, com maior evidência acima das 125 instâncias de treino.

Contrariamente ao método utilizado na experiência *Cold Start* nesta experiência não foi garantido equilíbrio entre casos positivos e negativos, bem como a distribuição homogénea dos casos para as duas classes. Neste caso o número de casos positivos (notícias do interesse do utilizador) suplanta claramente o número de casos negativos (notícias não classificadas como interessantes). Este fator pode explicar a degradação de performance do *J48* na medida em pode ser induzido um desequilíbrio na construção da árvore de decisão apesar da utilização do processo de *prunning* na sua implementação. Outra explicação possível será algum fenómeno de *overfitting* do classificador, dado que inicialmente este até apresenta resultados bastante bons para conjuntos de treino pequenos.

Para o tópico de interesse "Desporto" o classificador *Naive Bayes* revelou claramente melhor performance que o *RBF Network* e o *J48*, tendo obtido bons resultados mesmo com poucas instâncias de treino, com uma F_1 acima dos 80%. Os resultados para a F_1 rondaram os 75% nos casos do *RBF Network* e o *J48.*

Análise para a categoria Market (Economia)

Relativamente ao tópico de interesse "Economia" voltou a ser notória a degradação de performance do classificador *J48* acima das 100 instâncias de treino, apesar de a sua F_1 ter melhorado relativamente ao tópico anterior "Desporto". Este classificador atingiu ironicamente o maior valor registado para os três classificadores, para a F_1 com cerca de 78% às 100 instâncias de treino. Tanto o classificador *RBF Network* como o classificador *Naive Bayes* apresentam novamente resultados semelhantes para

a F_1, a rondar os 76% acima das 75 instâncias, ainda que com comportamentos distintos na evolução da aprendizagem.

No caso do classificador *Naive Bayes* observamos uma queda de performance na passagem de 25 para 50 instâncias de apesar de a F_1 evoluir positivamente a partir desse momento. A sua precisão e abrangência apresentam também um comportamento altamente divergente até às 150 instâncias de treino. Já no caso da *RBF Network* este aumenta a sua performance drasticamente na passagem de 50 para 75 instâncias de treino, revelando maior convergência entre a precisão e abrangência na classificação. O classificador *RBF Network* revelou assim um comportamento estável na aprendizagem, tendo a sua performance aumentado gradualmente com o número de instâncias de treino.

Mais uma vez e de forma semelhante à experiência anterior, o tópico "Economia" foi o que revelou grande desequilíbrio no comportamento dos classificadores e na evolução da aprendizagem ao longo do treino.

Análise para a categoria Political (Política)

Tal como verificado na Experiência 3 para o tópico de interesse "Política" os classificadores revelam maior divergência entre a precisão e abrangência. Mais uma vez este comportamento parece estar relacionado não só com a existência de mais notícias do tópico política nos conjuntos de teste, mas também com escolhas mais dispares entre os utilizadores para este tópico. Ainda assim, para este tópico o classificador *J48* já apresenta uma evolução relativamente positiva na aprendizagem (contrariamente ao sucedido para os outros dois tópicos) ainda que apresentando elevada divergência entre a precisão e abrangência. Apesar da divergência, o *J48* apresenta em todos os momentos do treino melhores resultados que os demais classificadores, atingindo um pico de 75% para a F_1 às 100 instâncias de treino.

Nos casos do classificador *RBF Network* e do classificador Naive Bayes observa-se um comportamento convergente entre a precisão e a abrangência à medida que o tamanho do treino aumenta, sendo que esse comportamento se reflete na evolução positiva do valor da F_1 para os dois classificadores. Estes dois classificadores apresentam performance semelhante ao longo do treino, apesar de o classificador Naive Bayes ter superado ligeiramente o classificador *RBF Network* atingidas as 150 instâncias de treino, com 73% e 71% respetivamente.

Considerações finais para a Experiência 4

O resultados médios para esta experiência revelam que o melhor classificador foi novamente e à semelhança da Experiência 3 o classificador *Naive Bayes* que atinge o seu pico de performance às 150 instâncias de treino, com 76% para a F_1. Logo a seguir apresenta-se o *J48* com uma evolução da F_1 aproximada do classificador *Naive Bayes* mas que apresenta uma queda a partir das 100 instâncias de treino. Apesar de ter revelado um comportamento positivo na aprendizagem (dentro do que seria esperado) o classificador *RBF Network* apresentou resultados para a F_1 um pouco abaixo tanto do classificador *Naive Bayes* como do *J48* para conjuntos de treino com

poucas instâncias. A sua performance obteve melhores resultados acima das 50 instâncias de treino, sendo que abaixo desse número este classificador revelou elevada divergência entre a precisão e abrangência.

Ainda relativamente a esta experiência podemos observar que o comportamento dos três classificadores sofreu algumas alterações relativamente à Experiência 3. A primeira conclusão a retirar prende-se com a descida da performance média dos classificadores, tendo os valores para a F_1 média nunca ultrapassado os 76%. Este resultado pode ser explicado se tivermos em conta que nesta experiência não existe garantia de equilíbrio entre casos positivos e casos negativos (exemplos das classes negativas e positivas), pois os conjuntos de treino dependem das escolhas dos utilizadores. Os conjuntos de treino variam não só com o perfil de utilizador mas também com o seu comportamento na seleção de notícias. O número de escolhas/leituras varia de utilizador para utilizador tal como os seus interesses específicos, mesmo quando estamos a considerar utilizadores com um perfil estereotipado.

Apesar de no geral o classificador *Naive Bayes* se ter revelado como o melhor dos três classificadores testados, verificou em algumas situações um comportamento anómalo, como é o caso da descida de performance na passagem de 25 para 50 casos de treino na categoria "Economia". Provavelmente este fenómeno dever-se-á a um de dois fatores: o aumento da entropia no conjunto de treino, ou seja, terão surgido muitos casos com elevada quantidade de nova informação ou a um fenômeno de contradição de pares, no qual devido ao processo de *"estimation smoothing"* não se preserva a ordem das estimativas de maior probabilidade.

O problema das distribuições com elevada entropia bem como o problema da contradição de pares foi já estudado em alguns trabalhos de investigação (Rish, Hellerstein, and Thathachar 2001) e (C. Zhang et al. 2009) respectivamente. No seu trabalho, (Rish et al. 2001) verificam experimentalmente que as *Naive Bayes* revelam melhor performance em distribuições com baixa entropia. Provavelmente o desiquilíbrio nas instâncias de treino aumenta a entropia da distribuição de informação.

5 TRABALHO RELACIONADO

Nesta secção é realizada uma análise crítica de um conjunto de trabalhos que têm vindo a ser desenvolvidos nas áreas dos sistemas de recomendação de conteúdos. As abordagens aqui revistas constituem uma base de inspiração e orientação para o trabalho que esta tese propõe.

5.1 Categorização de Informação

- Gemmell et al. (2008) propuseram um método de personalização de utilizadores a partir do *clustering* de *tags* em folksonomias.

  Na sua abordagem os *tag clusters* são utilizados como ligação entre os utilizadores e os recursos. Depois do processo de *clustering* é calculado o interesse de cada utilizador, em cada um dos *clusters*, com base no princípio de que se o utilizador utiliza frequentemente uma *tag* tem maior interesse no *cluster* onde se enquadra a *tag*. Do mesmo modo, é calculado uma medida de relação entre cada *cluste*re cada recurso. Deste modo, é possível estabelecer uma relação de interesse entre recursos e utilizadores, utilizando os *clusters* como elo de ligação. Esta relação, permite ainda derivar interesses anexos de cada utilizador.

  Na abordagem seguida no sistema implementado por este trabalho, também os *clusters* servem de elo entre utilizadores e conteúdos, sendo que no trabalho proposto os *clusters* são inferidos a partir de palavras-chave recolhidas a partir dos conteúdos e não a partir de folksonomias. Também no método de relação entre conteúdos e utilizadores a abordagem aqui proposta não utiliza métricas de proximidade mas optou-se por uma abordagem baseada em aprendizagem computacional.

- Chiang et al. (2004) apresentam um sistema inteligente de categorização de conteúdos para melhorar a pesquisa de notícias no contexto do projeto *INRA*, um sistema de escrita assistida que permite pesquisa de conteúdos notícias.

  O sistema que propuseram apresenta um mecanismo de categorização automática de texto como base para a recomendação de notícias aos utilizadores. Mais especificamente a sua abordagem combina um processo hierárquico de categorização de texto com filtragem baseada em conteúdo. Previamente ao processo de categorização, os textos foram processados utilizando um analisador sintático que identifica os termos gramaticais de uma frase, etiquetando-os. Os elementos etiquetados como "nomes" são depois recolhidos e selecionados para eliminar *stop words* e reduzir as variantes das palavras às suas formas radicais. O mecanismo de categorização utiliza uma *fuzzy network*, um conceito que combina métodos de raciocínio com redes neuronais para atribuir categorias às notícias, aprendendo computacionalmente a partir de exemplos de treino.

Uma particularidade do sistema foi a criação de um classificador específico para cada uma das categorias. De modo reduzir o tamanho do vetor de documentos e torna-lo útil ao treino dos classificadores, foi utilizada um método baseado em *TF-IDF* para calcular o peso de cada palavra para cada uma das categorias. Assim foi possível eliminar palavras menos relevantes.

Esta abordagem de categorização hierárquica demonstrou ser bastante eficiente na categorização de *corpus* de texto, obtendo resultados ao nível de um classificador *K-Nearest Neighbour* que havia obtido a melhor performance para o mesmo *corpus* testado.

A abordagem anteriormente descrita possui a utilização de aprendizagem computacional em comum com o trabalho proposto, ainda assim a grande diferença reside no fato de terem sido criados classificadores específicos para cada categoria, manualmente treinados para o efeito. Esta opção apesar de evidenciar grande performance para o *corpus* de teste utilizado carece de generalidade, na medida em que as categorias são definidas à partida.

5.2 Sistemas de Recomendação

- Ahn et al. (2007) propuseram um sistema de personalização de notícias (*YourNews*) no qual são utilizadas um conjunto de métricas para calcular o grau de semelhança entre os utilizadores e as notícias.

  Na sua abordagem pretendiam uma diferenciação dos sistemas tradicionais, tornando o sistema de recomendação mais transparente. Para tal permitiram ao utilizador a inspeção e modificação do seu perfil. Nos seus testes esta características parece não trazer benefícios ao sistema, causando até evidente perda de performance. A sua análise aos resultados aponta comportamentos desviantes dos utilizadores (remoção inesperada de grande número de categorias do seu perfil) como causa para os fracos resultados.

  Fica patente nos resultados obtidos que a liberdade de modificar a qualquer altura o perfil de utilizador pode induzir no sistema desequilíbrios que levam ao ineficaz funcionamento dos sistemas de recomendação. A cada alteração por parte do utilizador é necessário que o sistema se consiga adaptar em tempo útil por forma a considerar o novo perfil do utilizador.

- Billsus and Pazzani (1999) propuseram um agente de recomendação de notícias que batizaram de *NewsDude*.

  A sua abordagem consiste num sistema de recomendação em duas fases: o sistema começa por recomendar notícias baseando-se nos interesses a curto prazo do utilizador, caso não sejam encontradas notícias que satisfaçam a condição este recomenda notícias baseadas nos seus interesses a longo prazo.

  A recomendação a curto prazo é efetuada a partir de um modelo criado recorrendo a uma combinação de *TF-IDF* com *Nearest Neighbour* (NN),

capaz de lidar com o carácter dinâmico dos interesses de curto prazo. No caso da recomendação a longo prazo o sistema recorre a um classificador baseado em *Naive Bayes*.

A abordagem aqui proposta apresenta uma tendência cada vez mais evidente nos sistemas de classificação modernos, ou seja, a utilização de modelos híbridos de recomendação. Neste caso, o objetivo de utilizar um modelo de duas camadas de recomendação, utilizando métodos distintos para recomendar a curto e a longo prazo tem como objetivo combater o maior dinamismo dos interesses a curto prazo dos utilizadores. Este fator ficou bem patente no trabalho referido anteriormente Ahn et al. (2007) em que a constante alteração do perfil por parte de alguns utilizadores prejudicou a performance do sistema de recomendação.

- Agrawal et al. (2009) propuseram um sistema de recomendação de notícias para a Rede Social *Facebook*. O sistema recolhe notícias online e filtrando-as de acordo com a descrição da comunidade.

  O sistema começa por recolher palavras chave criadas pelos utilizadores para descrever as comunidades. Para cada comunidade o sistema procura no *Yahoo! News*[19] notícias relacionadas. As notícias são depois processadas extraindo palavras chave que servirão para identificar categorias através de um processo de *clustering* aglomerativo.

  Um resumo diário das notícias é entregue aos utilizadores utilizando recomendação baseada no conteúdo. É também entregue aos utilizadores um conjunto de notícias populares construído com base no *feedback* fornecido pelos utilizadores. O *feedback* colaborativo é gerado a partir da medição das seguintes observações: número de cliques nos artigos, classificação dos artigos (*ratings*), número de recomendações do artigo a terceiros.

  O sistema proposto consiste essencialmente num sistema de recolha e categorização de notícias. Neste caso não existe propriamente uma aprendizagem do modelo de utilizador, mas sim uma filtragem por categoria de notícias associadas às temáticas presentes em cada comunidade, sendo a recomendação efetuada pela sua popularidade na rede. Esta abordagem é de certa forma redutora pois provavelmente nem todos os membros de uma comunidade gostarão do mesmo tipo de conteúdos. Por outro lado os artigos mais populares tendem a sobrepor-se aos menos populares, resultando numa possível perda de alguns conteúdos negligenciados pela rede de utilizadores.

- Lee et al. (2007) propuseram o *MONERS*, um sistema de recomendação de notícias direcionado para dispositivos móveis.

  No seu trabalho identificaram dois pontos essenciais na recomendação de notícias no contexto dos dispositivos móveis: o primeiro prende-se com o

[19] http://news.yahoo.com

serviço e os conteúdos, os utilizadores que leem notícias em dispositivos móveis procuram geralmente notícias recentes, o segundo ponto prende-se com os aspetos demográficos e sociais dos utilizadores que consideram ter impacto nos padrões de utilização. A sua abordagem reflete o fator tempo no sentido em que a recomendação tem em conta a "idade" das notícias, dando também maior relevância a notícias associadas a categorias que recentemente atraíram a atenção do utilizador (medindo o número de visualização para cada categoria). As recomendações consideram ainda uma divisão segmentada por estereótipos de utilizador e por importância dos conteúdos (notícias encontradas sem secções de destaque, como as primeiras páginas dos jornais).

Nos testes as notícias foram entregues ao utilizador de duas formas distintas: numa primeira experiência organizando as notícias cronologicamente e numa segunda experiência organizando as notícias por categorias. Os resultados revelaram que o pior rácio de aceitação aconteceu na recomendação baseada na "idade" das notícias, sendo que o rácio de leitura foi maior quando as notícias eram entregues por categorias. Os utilizadores tendem a selecionar várias notícias dentro de cada categoria.

Um aspeto importante a reter, é o fato de os testes efetuados revelarem uma tendência dos utilizadores de dispositivos móveis, para preferirem sistemas que organizam os conteúdos de uma forma que torna a leitura mais confortável. Este comportamento estará certamente associado à maior facilidade em identificar os conteúdos da sua preferência, quando estes lhes são apresentados e filtrados por tópicos de interesse.

6 CONSIDERAÇÕES FINAIS

Neste capítulo é efetuada uma reflexão crítica do trabalho desenvolvido no decorrer das fazes de desenvolvimento do sistema proposto para elaboração desta tese, bem como o levantamento de algumas conclusões relativas ao trabalho efetuado. Serão também apresentados algumas considerações relativamente às opções tomadas e a possíveis caminhos a seguir no desenvolvimento futuro do sistema (secções 6.1 e 6.2 respetivamente).

Na realização deste trabalho foram abordados um conjunto de métodos de extração de informação e construção de estruturas de conhecimento a partir de fontes não estruturadas, nomeadamente a categorização de notícias e a identificação de tópicos de interesse. Foram também abordadas técnicas de aprendizagem computacional aplicadas à aprendizagem de modelos de utilizador com aplicação em sistemas de recomendação (ver secção 3).

O sistema desenvolvido apresentou resultados positivos, tendo os módulos desenvolvidos cumprido os seus objetivos, mostrando-se adequados à realização das experiências levadas a cabo. Foi possível implementar com sucesso o módulo de recolha e extração de palavras-chave a partir de notícias recorrendo à *framework KEA* (Witten et al. 1999) tendo este sido testado na Experiência 1 (secção 4.2).

A palavras-chave recolhidas foram utilizadas para construir a representação de conhecimento com base na coocorrência das palavras-chave nas notícias, a partir da *Distributional Hypothesis* em Linguística (Harris 1954). A estrutura de conhecimento foi implementada eficazmente na forma de grafo de coocorrência utilizando o *Neo4J*. Estes dois módulo foram então integrados no primeiro protótipo da aplicação, constituindo a primeira versão do Agente Principal.

A estrutura de conhecimento foi então sujeita ao processo de identificação de tópicos de interesse, utilizando o algoritmo *"fast unfolding of communities in large networks"* (Blondel et al. 2008), uma adaptação do algoritmo originalmente proposto por Newman-Girvan (Girvan and M. E. J. Newman 2002). Este algoritmo revelou-se bastante eficaz na identificação de um conjunto considerável de tópicos de interesse presentes na estrutura de conhecimento, como revelam os resultados obtidos na Experiência 2 (secção 4.3). Neste processo foram ainda selecionados os identificadores de cada tópico de interesse, através do cálculo da *Betweenness Centrality* (Freeman 1978) para cada um dos nós de cada comunidade. A informação produzida pelos processos anteriores foi adicionada à estrutura de conhecimento, tendo sido assim concluída com sucesso a fase de produção de conhecimento. Deste modo ficou concluída a segunda versão do Agente Principal.

Na última fase de desenvolvimento foi implementada a rede de agentes, composta pelo Agente Principal e a aplicação móvel que corresponde aos Agentes Pessoais. Ao Agente Principal foi adicionada uma componente de servidor, na forma de *Web*

Service que permitiu a comunicação com os Agentes Pessoais. Ainda nesta fase foi desenvolvido o módulo de classificação no Agente Principal. Para o efeito recorreu-se à *framework Weka* (Hall et al. 2009) que disponibilizou as implementações dos três classificadores estudados nesta tese, respetivamente *Naive Bayes*, *RBF Networks* e *J48*. Deste modo deu-se por concluída a versão final do Agente Principal que ficou apto a disponibilizar aos Agentes Pessoais o conhecimento e os algoritmos de classificação.

Quanto aos Agentes Pessoais foram implementados através de uma abordagem híbrida, especificada na secção 4.4.1, com a interface de utilizador desenvolvida utilizando a plataforma *Google Android*. A aplicação móvel disponibilizou não só a interface, como o mecanismo de *feedback* necessários à obtenção do Modelo de Utilizador. Com a implementação dos Agentes Pessoais e comunicação com o Agente Principal, deu-se por concluída o último protótipo do sistema, que foi então testado na recomendação de notícias no "arranque a frio" (Experiência 3, secção 4.5) da aplicação bem como a longo prazo aprendendo com a leitura e classificação de notícias por parte dos utilizadores (Experiência 4, secção 4.6).

Com a realização das experiências de treino dos classificadores deu-se por concluída a parte experimental deste trabalho, tendo sido avaliados os resultados produzidos e utilizados na escrita desta tese.

O desenvolvimento do sistema de recomendação proposto, bem como estudo dos resultados obtidos, permitiram retirar um conjunto de conclusões relativamente à recomendação baseada em conhecimento. As principais contribuições desta tese são as seguintes:

- Foi proposta uma abordagem para a criação de um sistema de recomendação de notícias baseado em conhecimento.
- Foram estudadas técnicas de extração de palavras-chave, aplicadas ao contexto da categorização automática de notícias.
- Foram identificadas as características necessárias à concepção de uma estrutura dinâmica de conhecimento tendo como base a informação extraída a partir dos conteúdos noticiosos. Tendo sido testada a sua implementação através da utilização de uma bases de dados de grafos.
- Estudou-se a aplicação do conceito de coocorrência, tendo como ponto de partida a Hipótese de Distribuição em Linguística (Harris 1954), na produção de conhecimento útil ao contexto dos sistemas de recomendação.
- A aplicação de algoritmos de deteção de comunidades em grafos foi estudada com aplicação na identificação de tópicos de interesse.
- Foi efetuado um estudo da aplicação de métricas de autoridade em redes como processo de categorização de tópicos de interesse.
- Estudou-se também a aplicabilidade de mecanismos de aprendizagem computacional na produção de modelos de utilizador e geração de recomendações.

6.1 Fases de Desenvolvimento

Neste capítulo são descritos em pormenor as fases de desenvolvimento do sistema proposto e respetiva elaboração de tese, bem como a sua planificação temporal.

No diagrama seguinte é possível observar o planeamento seguido no decorrer do primeiro semestre letivo que abrangeu o levantamento do estado da arte e revisão bibliográfica bem como a elaboração da proposta de tese.

Planeamento do Primeiro Semestre

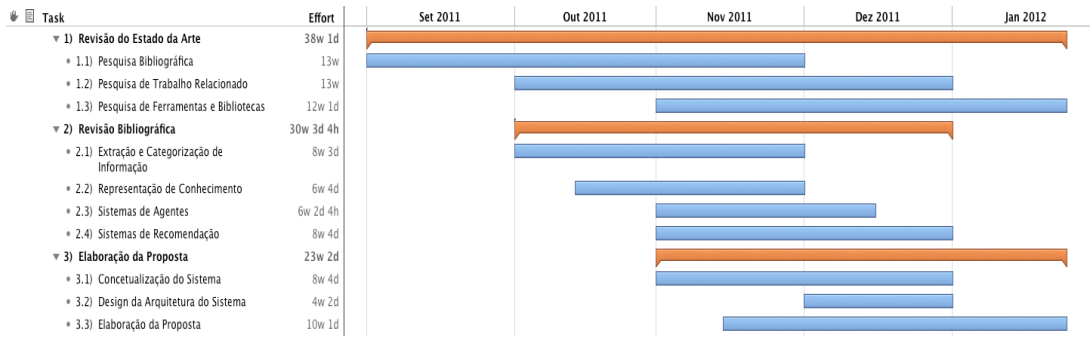


Imagem 53 – Diagrama das tarefas realizadas no primeiro semestre.

Neste segundo diagrama está representado o planeamento do trabalho realizado no segundo semestre letivo, que incluiu a fase de desenvolvimento dos protótipos do sistema e sua validação, os testes e avaliação do seu desempenho, bem como a escrita da versão final da tese que suporta esta abordagem.

Planeamento do Segundo Semestre

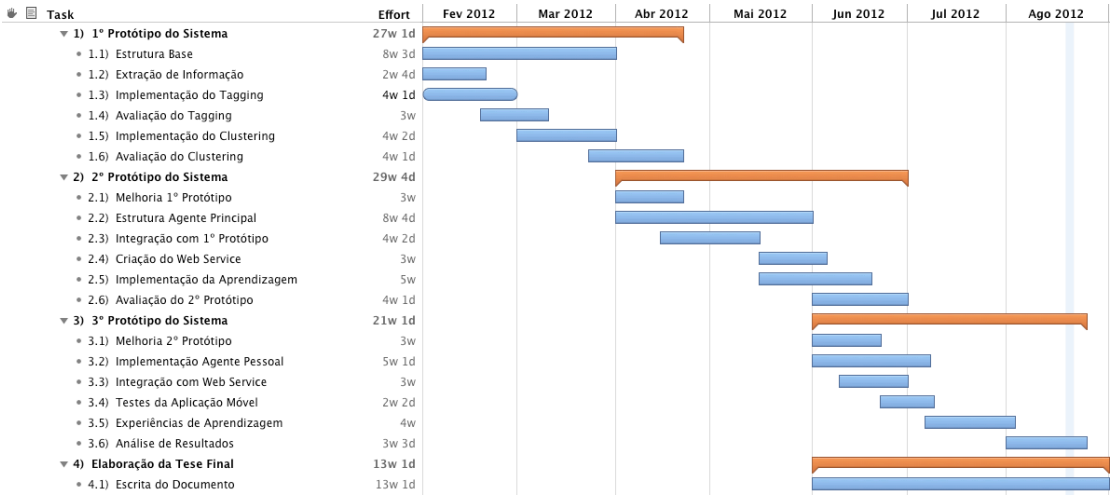


Imagem 54 – Diagrama das tarefas a realizadas no segundo semestre.

Seguindo como referência o diagrama representado na Imagem 54, em seguida são descritas em detalhe as várias fases do trabalho desenvolvido durante o segundo semestre:

1. Implementação do Primeiro Protótipo

Nesta primeira fase foi necessário criar a base de toda a estrutura futura do sistema de modo a albergar no futuro toda a arquitetura proposta (ver Capítulo 4). Inicialmente foi necessário desenvolver os módulos de recolha de notícias e extração de palavras-chave. Deste modo foi possível começar o processo de agregação e categorização de conteúdos que seriam utilizados para testar o sistema. A determinado momento, recolhidos que estavam um conjunto interessante de notícias, o sistema foi "paralisado" por forma a proceder a uma avaliação dos módulos implementados, bem como da informação recolhida. Na implementação do processo de extração de palavras-chave foi desenvolvido adicionalmente uma interface *Web* que permitiu realizar a avaliação deste processo por um conjunto de utilizadores (ver Experiência 1, secção 4.2).

No seguimento da implementação do processo de recolha e classificação de informação, foi criada a estrutura que alberga as relações entre as palavras-chave recolhidas. O sistema foi de novo colocado em funcionamento agora com o processo de inserção de palavras-chave na estrutura de conhecimento já funcional. Neste momento foi efetuada uma avaliação da criação da estrutura de conhecimento (grafo de coocorrência de palavras-chave) para aferir o correto funcionamento de todo o processo de classificação e recolha de informação.

Ainda nesta fase e após uma avaliação das estruturas e dos processos até então criados procedeu-se à implementação do *clustering* de tópicos de interesse bem como à categorização de cada um dos *clusters* identificados (ver Experiência 2, secção 4.3). Este processo seguiu uma abordagem *offline*, no qual o grafo de palavras-chave foi exportado a partir do servidor de grafos e importado utilizando a aplicação de manipulação e análise grafos. Após um processo de avaliação manual do *clustering* e da identificação de tópicos, a informação relativa a este processo foi integrada na estrutura de conhecimento. Neste momento estava concluída a fase de construção da estrutura de conhecimento a partir das palavras-chave extraídas.

2. Implementação do Segundo Protótipo

Na segunda fase de implementação e seguindo uma linha de construção incremental do desenvolvimento do sistema, foram efetuados alguns ajustes e correções ao protótipo anterior de modo a garantir a integração dos módulos a desenvolver de seguida.

Este segundo protótipo consistiu numa modificação da arquitetura do sistema desenvolvido no primeiro protótipo por forma a funcionar agora como servidor. Esta transformação conferiu ao sistema as características necessárias ao funcionamento híbrido do Agente Pessoal, conforme explicado em 4.4.1. A construção do servidor passou pela implementação de um *Web Service* que disponibiliza *API* de comunicação a aplicação móvel a desenvolver. Numa última fase da implementação do segundo protótipo foi implementado no servidor o módulo de aprendizagem das preferências de utilizador (Módulo de Classificação, secção 4.4.5), bem como as estruturas de representação do modelo de utilizador no Agente Pessoal (ver secção 4.4.2).

Concluída a implementação deste segundo protótipo foram realizados alguns testes experimentais do funcionamento dos módulo criados, nomeadamente testes preliminares do funcionamento do *Web Service* e dos classificadores implementados no módulo de aprendizagem.

3. Implementação do Terceiro Protótipo

Na última fase de desenvolvimento do sistema foi desenvolvida a aplicação móvel que funciona como interface para o Assistente Pessoal. A aplicação que constitui o Assistente Pessoal foi implementada como um cliente do servidor (*Web Service*) que disponibiliza aos agentes as

funcionalidades necessárias à aprendizagem do modelo de utilizador e geração de recomendações. Após a produção da aplicação móvel seguiu-se um conjunto de testes para aferir a correta implementação da comunicação na rede de agentes e servidor.

Ainda nesta fase foram efetuadas as experiências de obtenção do modelo de utilizador e geração de recomendações utilizando a implementação final deste protótipo servidor-cliente. Nas experiências realizadas com o terceiro protótipo foram avaliados os classificadores implementados na capacidade de aprender o modelo dos utilizadores e recomendar notícias do seu interesse. Estas experiências geram do lado do servidor um conjunto de estatísticas de performance que permitiram realizar uma análise dos resultados da aprendizagem e classificação (ver secções 4.5.3 e 4.6.3 para as experiências 3 e 4 respetivamente).

Os dados resultantes das experiências foram então processados, gerando os dados necessários à escrita da tese que suporta o sistema proposto.

4. Elaboração da tese final

A última tarefa realizada consistiu na análise dos resultados obtidos, culminando na escrita da versão final deste documento.

6.2 Trabalho Futuro

Relativamente ao trabalho futuro são apresentadas de seguida algumas linhas orientadores de possíveis caminhos a seguir no desenvolvimento futuro deste sistema:

- A utilização de *palavras-chave* provenientes de anotações sociais.
- A adaptação dos mecanismos de extração de palavras-chave para outras Línguas, nomeadamente o português.
- O estudo de outros métodos de *clustering* com sobreposição de palavras-chave em tópicos distintos.
- A utilização do mecanismo de *feedback* de modo a recolher dados relativos à popularidade, tendências e efeito surpresa de novas notícias, numa abordagem *community-based* ou *collaborative-filtering*.
- A implementação de mecanismos de recolha automática de informação do contexto e comportamentos do utilizador na leitura das notícias, evitando assim a classificação manual e explícita dos conteúdos.
- A implementação de mecanismos de *matching* de utilizadores com preferências próximas de forma a utilizar esta informação para gerar recomendações.
- A utilização de outras fontes de informação menos estruturadas, nomeadamente Redes Sociais, conteúdos de Blogues.
- Aprendizagem do modelo de utilizador a partir de fontes externas como os perfis e comentários em Redes Sociais ou artigos criados pelos utilizadores.
- A implementação de uma versão *standalone* o sistema de recomendação, sem recorrer a um Agente Principal ou implementada de forma distribuída ao estilo *peer-to-peer*.

Referências

Agrawal, Manish, M. Karimzadehgan, and C.X. Zhai. 2009. "An online news recommender system for social networks." in *Proceedings of the Workshop on Search in Social Media (SSM '09)*. Boston, MA, USA.

Ahn, Jae-wook, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. 2007. "Open user profiles for adaptive news systems: help or harm?" Pp. 11–20 in *Proceedings of the International Conference on World Wide Web (WWW '07)*. Banff, AB, Canada.

Aliakbary, Sadegh, Hassan Abolhassani, Hossein Rahmani, and Behrooz Nobakht. 2009. "Web Page Classification Using Social Tags." Pp. 588–593 in *Proceedings of the International Conference on Computational Science and Engineering (CSE '09)*. Washington, DC, USA: IEEE Computer Society.

Bain, Alexander. 1873. *Mind and Body: The Theories of Their Relation*. American E. New York, NY, USA: D. Appleton and Company.

Bayes, Thomas. 1763. "An Essay towards solving a Problem in the Doctrine of Chances." *Philosophical Transactions* 53:37–38.

Billsus, Daniel, and Michael J. Pazzani. 1999. "A hybrid user model for news story classification." Pp. 99–108 in *Proceedings of the International Conference on User Modeling (UM '99)*. Banff, AB, Canada: SPRINGER-VERLAG.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):1000–8.

Brafman, Ronen I., and Carmel Domshlak. 2009. "Preference Handling – An Introductory Tutorial." *Artificial Intelligence Magazine*, 58–86.

Burke, Robin. 2002. "Hybrid Recommender Systems: Survey and experiments." *User Modeling and User-Adapted Interaction* 12(4):331–370.

Burke, Robin. 2007. "Hybrid Web Recommender Systems." Pp. 377–408 in *The Adaptive Web*, vol. 4321, edited by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Berlin, Heidelberg: Springer Berlin Heidelberg.

Clark, Peter, and Tim Niblett. 1989. "The CN2 Induction Algorithm." *Machine Learning* 3(4):261–283.

Codd, Edgar Frank. 1970. "A relational model of data for large shared data banks." *Communications of the ACM* 13(6):377–387.

Cristo, Marco, Edleno Silva Moura, Nivio Ziviani, and Berthier Ribeiro-Neto. 2003. "Link Information as a Similarity Measure in Web Classification." Pp. 43–55 in *Proceedings of the 10th Symposium On String Processing and Information Retrieval*, vol. 2857. Mansus, AM, Brazil: Springer Verlag.

Cunningham, Hamish et al. 2011. *Text Processing with GATE (Version 6)*. Sheffield, UK: University of Sheffield, Department of Computer Science.

Dagan, Ido, Lillian Lee, and Fernando C. N. Pereira. 1998. "Similarity-Based Models of Word Cooccurrence Probabilities." *Machine Learning* 34(1):43–69.

Dhillon, Inderjit, Yuqiang Guan, and Brian Kulis. 2005. *A Unified View of Kernel k-means, Spectral Clustering and Graph Cuts*. Austin, Texas, USA.

Fensel, Dieter. 2001. *Ontologies: A silver bullet for knowledge management and electronic commerce*. New York, NY, USA: Springer Verlag.

Fix, Evelyn, and JL. Hodges. 1951. "Discriminatory analysis. Nonparametric discrimination: Consistency properties." 21.

Fox, Mark S., and John McDermott. 1986. "The role of databases in knowledge-based systems." Pp. 407–430 in *On knowledge base management systems: integrating artificial intelligence and d atabase technologies*. New York, NY, USA: Springer Verlag.

Freeman, Linton C. 1978. "Centrality in social networks: Conceptual clarification." *Social Networks* 1(3):215–239.

Gemmell, Jonathan, Andriy Shepitsen, Bamshad Mobasher, and Robin Burke. 2008. "Personalization in folksonomies based on tag clustering." *6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems* 12:37–48.

Gemmis, Marco et al. 2009. "Preference learning in recommender systems." Pp. 41–55 in *Workshop on Preference Learning (ECML/PKDD-09)*. Bled, Slovenia.

Girvan, Michelle, and Mark E. J. Newman. 2002. "Community structure in social and biological networks." *Proceedings of the National Academy of Sciences of the United States of America* 99(12):7821–6.

Golbeck, Jennifer. 2006. "Generating Predictive Movie Recommendations from Trust in Social Networks." Pp. 93–104 in *Proceedings of the 4th international conference on Trust Management (iTrust '06)*, vol. 3986. Pisa, Italy: Springer Verlag.

Goldberg, David, David Nichols, Brian M. Oki, and Douglas Terry. 1992. "Using collaborative filtering to weave an information tapestry." *Communications of the ACM* 35(12):61–70.

Gower, John C., and G. J. S. Ross. 1969. "Minimum spanning trees and single linkage cluster analysis." *Journal of the Royal Statistical Society (Applied Statistics)* 18(1):54–64.

Guttman, Robert H., Alexandros G. Moukas, and Pattie Maes. 1998. "Agent-mediated electronic commerce: a survey." *The Knowledge Engineering Review* 13(02):147–159.

Hall, Mark et al. 2009. "The WEKA data mining software: an update." *ACM SIGKDD Explorations Newsletter* 11(1):10–18.

Hamming, Richard W. 1950. “Error detecting and error correcting codes.” *Bell System Technical Journal* 29(2):147–160.

Harris, Z. 1954. “Distributional structure” edited by Jerry Fodor and Jerrold Katz. *Word Journal Of The International Linguistic Association* 10(23):146–162.

Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. “Evaluating collaborative filtering recommender systems.” *ACM Transactions on Information Systems* 22(1):5–53.

Hockenmaier, Julia, Gann Bierner, and Jason Baldridge. 2000. “Providing Robustness for a CCG System.” Pp. 99–112 in *Proceedings of Workshop on Linguistic Theory and Grammar Implementation (ESSLLI '00)*. Birmingham, UK.

Huang, Zan, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. 2002. “A graph-based recommender system for digital library.” Pp. 65–73 in *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '02)*. Portland, Oregon, USA: ACM Press.

Huffman, Scott B., and John E. Laird. 1995. “Flexibly Instructable Agents.” *Journal Of Artificial Intelligence Research* 3:271–324.

Joachims, Thorsten. 1998. “Text categorization with support vector machines: Learning with many relevant features.” Pp. 137–142 in *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*, vol. 1398. London, UK: Springer Verlag.

Jones, Michael P., and James H. Martin. 1997. “Contextual spelling correction using latent semantic analysis.” Pp. 166–173 in *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLC '97)*. Washington, DC, USA: Association for Computational Linguistics.

Kande, Giri Babu, Satya Savithri, and Venkata Subbaiah. 2007. “Segmentation of Vessels in Fundus Images using Spatially Weighted Fuzzy c-Means Clustering Algorithm.” *Journal of Computer Science* 7(12):102–109.

Kass, Robert, and Tim Finin. 1988. “Modeling the User in Natural Language Systems” edited by Allfred Kobsa and Wolfgang Wahlster. *Computational Linguistics* 14(3):5–22.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data.” Pp. 282–289 in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*. Williamstown, MA, USA: Morgan Kaufmann Publishers Inc.

Levenshtein, V. I. 1966. “Binary codes capable of correcting deletions, insertions, and reversals.” *Soviet Physics Doklady* 10(8):707–710.

Levesque, Hector J., and Gerhard Lakemeyer. 2000. *The logic of knowledge bases*. Cambridge, MA, USA: The MIT Press.

Liu, Jiming, and Jian-Bing Wu. 1999. "Evolutionary group robots for collective world modeling." Pp. 48–55 in *Proceedings of the third annual conference on Autonomous Agents (AGENTS '99)*. Seattle, Washington, USA: ACM Press.

Liu, Ning-han, Szu-wei Lai, Chien-yi Chen, and Shu-ju Hsieh. 2009. "Adaptive Music Recommendation Based on User Behavior in Time Slot." *International Journal of Computer Science and Network Security (IJCSNS)* 9(2):219–227.

Loper, Edward, and Steven Bird. 2002. "NLTK: The natural language toolkit." Pp. 63–70 in *Proceedings of the (ACL-02) Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Philadelphia, USA: ACL.

Luhn, H.P. 1957. "A statistical approach to mechanized encoding and searching of literary information." *IBM Journal of Desearch and Development* 1(4):309–317.

Luo, Feng, and Richard H. Scheuermann. 2006. "Detecting Functional Modules from Protein Interaction Networks." Pp. 123–130 in *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCCS '06)*, vol. 1. Hangzhou, China: IEEE Computer Society.

Maes, Pattie. 1994. "Agents that reduce work and information overload." *Communications of the ACM* 37(7):30–40.

Matsuzaki, Takuya, Yusuke Miyao, and J. Tsujii. 2003. "An efficient clustering algorithm for class-based language models." Pp. 119–126 in *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL (CONLL '03) - Volume 4*. Edmonton, Canada: ACL.

Matuszek, Cynthia, John Cabral, Michael Witbrock, and John Deoliveira. 2006. "An introduction to the syntax and content of Cyc." Pp. 44–49 in *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, vol. 3864. Standford, California, USA: AAAI Press.

McCulloch, Warren S., and Walter Pitts. 1943. "A logical calculus of the ideas immanent in nervous activity." *Bulletin of mathematical biology* 5(4):115–133.

McDonald, Scott, and Michael Ramscar. 2001. "Testing the distributional hypothesis: The influence of context on judgements of semantic similarity." Pp. 611–616 in *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Edinburgh, Scotland: Cognitive Science Society.

Mcnally, Kevin, Michael P. O. Mahony, Barry Smyth, Maurice Coyle, and Peter Briggs. 2010. "Towards a Reputation-based Model of Social Web Search." Pp. 179–188 in *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*. Hong Kong, China: ACM Press.

Meeker, Mary, Scott Devitt, and Liang Wu. 2010. "Internet Trends." P. 87 in *Technology Industry Conference*. Morgan Stanley Research Retrieved

(http://www.morganstanley.com/institutional/techresearch/pdfs/Internet_Trends_041210.pdf).

Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. "Introduction to wordnet: An on-line lexical database." *International Journal of Lexicography* 3(4):235–244.

Mohamed, Samar S., and Magdy MA Salama. 2007. "Spectral clustering for TRUS images." *BioMedical Engineering Online* 6:10.

Neches, Robert et al. 1991. "Enabling technology for knowledge sharing." *AI Magazine* 12(3):36–56.

Negroponte, Nicholas. 1970. *The architecture machine: Toward a more human environment*. London, UK: MIT Press.

Von Neumann, John, and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

Newman, M. 2004. "Analysis of weighted networks." *Physical Review E* 70(5):1–9.

Noll, Michael G., and Christoph Meinel. 2008. "The Metadata Triumvirate: Social Annotations, Anchor Texts and Search Queries." *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 2008 (WI-IAT '08)* 1:640–647.

Orchestr8 LLC. 2012. "AlchemyAPI." Retrieved January 5, 2012 (http://www.alchemyapi.com).

Orr, Mark J.L. 1996. *Introduction to radial basis function networks*. Scotland, UK.

Pham, Manh Cuong, Yiwei Cao, Ralf Klamma, and Matthias Jarke. 2010. "A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis." *Journal of Universal Computer Science* 17(4):1–21.

Quinlan, J R. 1986. "Induction of decision trees." *Machine Learning* 1(1):81–106.

Quinlan, J. Ross. 1993. "C4.5: Programs for Machine Learning." *Machine Learning* 16(3):235–240.

Quinlan, J.R., W.W. Cohen, and H. Hirsh. 1994. "The minimum description length principle and categorical theories." Pp. 233–241 in *Proceedings of the 11th International Conference on Machine Learning (ICML '94)*. New Brunswick, NJ, USA: Morgan Kaufmann.

Ramage, Daniel, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. 2009. "Clustering the tagged web." Pp. 54–63 in *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*. Barcelona, Spain: ACM Press.

Reich, Brian J., and Howard D. Bondell. 2011. "A spatial dirichlet process mixture model for clustering population genetics data." *Biometrics* 67(2):381–90.

Resnick, Paul, and H.R. Varian. 1997. "Recommender systems." *Communications of the ACM* 40(3):56–58.

Ricci, Francesco, Lior Rokach, and Bracha Shapira. 2011. *Introduction to Recommender Systems Handbook*.

Rish, Irina, Joseph Hellerstein, and Jayram Thathachar. 2001. *An analysis of data characteristics that affect Naïve Bayes performance*. Hawthorne, NY, USA.

Russell, Stuart, and Peter Norvig. 2010. *Artificial intelligence: A Modern Approach*. 3rd ed. New Jersey: Prentice Hall Series in Artificial Intelligence.

Sahami, M., Susan Dumais, D. Heckerman, and E. Horvitz. 1998. "A Bayesian approach to filtering junk e-mail." Pp. 98–105 in *AAAI Workshop on Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62. Madison, Wisconsin.

Salton, Gerard, and Christopher Buckley. 1988. "Term-weighting approaches in automatic text retrieval." *International Journal of Information Processing and Management* 24(5):513–523.

Schütze, Hinrich. 1998. "Automatic Word Sense Discrimination." *Computational Linguistics* 24(1):97–123.

Small, Henry. 1973. "Co-citation in the scientific literature: a new measure on the relationship between two documents." *Journal of the American Society for Information Science* 24(4):265–269.

Suchman, Lucy. 1987. *Plans and situated actions: the problem of human-machine communication*. New York, NY, USA: Cambridge University Press.

Terveen, L. 1995. "Overview of human-computer collaboration." *Knowledge-Based Systems* 8(2-3):67–81.

Vapnik, Vladimir. 1999. *The nature of statistical learning theory*. 2nd ed. New York, NY, USA: Springer Verlag.

Wasserman, Stanley, and Faust Katherine. 1994. *Social Network Analysis: Methods and Applications*. 1st ed. Cambridge, MA, USA: Cambridge University Press.

Watanabe, Yotaro, Masayuki Asahara, and Yuji Matsumoto. 2007. "A Graph-based Approach to Named Entity Categorization in Wikipedia Using Conditional Random Fields." Pp. 649–657 in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: ACL.

Whittaker, Mary, and Kathryn Breininger. 2008. "Taxonomy development for knowledge management." Pp. 1–10 in *World library and information congress: 74th IFLA general conference and council*. Québec, Canada: Boeing Library Services.

Witten, I.H., G.W. Paynter, Eibe Frank, Carl Gutwin, and C.G. Nevill-Manning. 1999. "KEA: Practical automatic keyphrase extraction." Pp. 254–255 in *Proceedings of the 4th ACM Conference on Digital libraries*. Berkeley, CA, USA: ACM Press.

Wooldridge, Michael. 2002. *An Introduction to MultiAgent Systems*. 2nd Editio. Chichester, England: John Wiley & Sons Ltd.

World Bank, The. 2012. "Information and Communications for Development 2012: Maximizing Mobile." 244. Retrieved (http://www.worldbank.org/ict/IC4D2012).

Yeung, Au, Ching Man, Nicholas Gibbins, and Nigel Shadbolt. 2008. "A k-Nearest-Neighbour Method for Classifying Web Search Results with Data in Folksonomies." Pp. 70–76 in *Proceedings of The 2008 IEEE/WIC/ACM International Conference on Intelligence Agent Technology*. Sydney, Australia: IEEE Computer Society.

Zhang, Congle, Gui-Rong Xue, Yong Yu, and Hongyuan Zha. 2009. "Web-scale classification with naive bayes." Pp. 1083–1084 in *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. New York, New York, USA: ACM Press.

Zhuo, Zhao, Shi-Min Cai, Zhong-Qian Fu, and Jie Zhang. 2011. "Hierarchical organization of brain functional network during visual task." *Physical Review E - Statistical, nonlinear, and soft matter physics*. 84(3):7.

Zubiaga, Arkaitz, Raquel Martínez, and Víctor Fresno. 2009. "Getting the most out of social annotations for web page classification." Pp. 74–83 in *Proceedings of the 9th ACM symposium on Document engineering (DocEng '09)*. Munich, Germany: ACM Press.